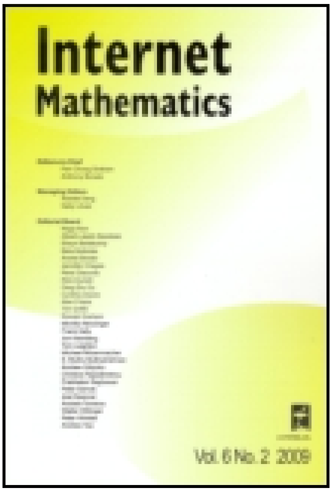


Internet Mathematics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uinm20

A Dynamical System for PageRank with Time-Dependent Teleportation

David F. Gleich[a] & Ryan A. Rossi[a]
[a] Purdue University, Department of Computer Science, 305 N. University St., West Lafayette, IN 47906, USA
Accepted author version posted online: 15 Apr 2014.Published online: 05 May 2014.

To cite this article: David F. Gleich & Ryan A. Rossi (2014) A Dynamical System for PageRank with Time-Dependent Teleportation, Internet Mathematics, 10:1-2, 188-217, DOI: 10.1080/15427951.2013.814092

To link to this article: http://dx.doi.org/10.1080/15427951.2013.814092

Internet Mathematics Vol. 10: 188–217

A Dynamical System for PageRank with Time-Dependent Teleportation

David F. Gleich and Ryan A. Rossi

Abstract. We propose a dynamical system that captures changes to the network centrality of nodes as external interest in those nodes varies. We derive this system by adding time-dependent teleportation to the PageRank score. The result is not a single set of importance scores, but rather a time-dependent set. These can be converted into ranked lists in a variety of ways, for instance, by taking the largest change in the importance score. For an interesting class of dynamic teleportation functions, we derive closed-form solutions for the dynamic PageRank vector. The magnitude of the deviation from a static PageRank vector is given by a PageRank problem with complex-valued teleportation parameters. Moreover, these dynamical systems are easy to evaluate. We demonstrate the utility of dynamic teleportation on both the article graph of Wikipedia, where the external interest information is given by the number of hourly visitors to each page, and the Twitter social network, where external interest is the number of tweets per month. For these problems, we show that using information from the dynamical system helps improve a prediction task and identify trends in the data.

1. Introduction

The PageRank vector of a directed graph is the stationary distribution of a Markovian *random surfer*. At a node, the random surfer either

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/uinm.

ISSN: 1542-7951 print

1. transitions to a new node uniformly chosen from the set of out-edges, or
2. does something else (e.g., leaves the graph and then randomly returns) [Page et al. 99, Langville and Meyer 06].

The probability that the surfer performs the first action is known as the *damping parameter* in PageRank and is denoted by α. The second action is called *teleporting* and is modeled by the surfer picking a node at random according to a distribution called the *teleportation distribution vector* or *personalization vector*. This PageRank Markov chain always has a unique stationary distribution for every $0 \leq \alpha < 1$. In this paper, we focus on the teleportation distribution vector $\mathbf{v}$ and study how changing teleportation behavior manifests itself in a dynamical system formulation of PageRank.

To proceed further, we need to formalize the PageRank model. Let $\boldsymbol{A}$ be the adjacency matrix for a graph, where $A_{i,j}$ denotes an edge from node i to node j. To avoid a proliferation of transposes, we define $\boldsymbol{P}$ as the transposed transition matrix for a random walk on a graph:

$$P_{j,i} = \text{probability of transitioning from node } i \text{ to node } j.$$

Hence, the matrix $\boldsymbol{P}$ is *column-stochastic*, in contrast to the more common row-stochastic matrices found in probability theory. Throughout this manuscript, we employ uniform random walks on a graph, in which case $\boldsymbol{P} = \boldsymbol{A}^T \boldsymbol{D}^{-1}$, where $\boldsymbol{D}$ is a diagonal matrix with the out-degree of each node on the diagonal. However, none of the theory is restricted to this type of random walk, and any column-stochastic matrix will do. If any nodes have no out-links, we assume that they are adjusted in one of the standard ways [Boldi et al. 07] (in our experiments, this applies only to Twitter, in which case we added a uniform transition distribution to nodes with no out-links; this is the weakly personalized case). Let $\mathbf{v}$ be a teleportation distribution vector such that $v_i \geq 0$ and $\sum_i v_i = 1$. This vector models where the surfer will transition when "doing something else." The PageRank Markov chain then has the transition matrix

$$\alpha \boldsymbol{P} + (1-\alpha)\mathbf{v}\mathbf{e}^T.$$

While finding the stationary distribution of a Markov chain usually involves computing an eigenvector or solving a singular linear system, the PageRank chain has a particularly simple form for the stationary distribution vector $\mathbf{x}$:

$$(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = (1-\alpha)\mathbf{v}.$$

The sensitivity of PageRank with respect to $\mathbf{v}$ is fairly well understood. In [Langville and Meyer 06], the authors devote a section to determining the Jacobian of the PageRank vector with respect to $\mathbf{v}$. The choice of $\mathbf{v}$ is often best guided by an application-specific measure. By setting $\mathbf{v} = \mathbf{e}_i$, that is, the ith canonical basis vector

$$\mathbf{e}_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

where the 1 is in the ith row, PageRank computes a highly localized diffusion that is known to produce empirically meaningful clusters and theoretically supported clusters [Andersen et al. 06, Tong et al. 06]. By choosing $\mathbf{v}$ based on a set of known-to-be-interesting nodes, PageRank will compute an expanded set of interesting nodes [Gyöngyi et al. 04, Singh et al. 07]. Yet in all of these cases, $\mathbf{v}$ is chosen once for the graph application or particular problem.

In the original motivation of PageRank [Page et al. 99], the distribution $\mathbf{v}$ should model how users behave on the Web when they do not click a link. When this intuition is applied to a site like Wikipedia, it suggests that the teleportation function should vary as particular topics become interesting. For instance, in our experiments (Section 5), we examine the number of page views for each Wikipedia article during a period during which a major earthquake occurred. Suddenly, page views to "Earthquake" spiked, presumably because that word was being searched. We wish to include this behavior in our PageRank model to understand what is now important in light of radically different behavior. One option would be to recompute a new PageRank vector given the observed teleporting behavior at the current time. Our proposal for a dynamical system is another alternative. That is, we define a new model in which teleportation is the time-dependent function $\mathbf{v}(t)$.

At each time t, $\mathbf{v}(t)$ is a probability distribution of the places to which the random walk teleports. Figure 1 illustrates this model. We return to a comparison between this approach and solving PageRank systems in Section 4.

The dynamical system we propose is a generalization of PageRank in the sense that if $\mathbf{v}(t)$ is a constant function in time, then we converge to the standard PageRank vector (Theorem 2.4). Additionally, we can analyze the dynamical PageRank function for some simple oscillatory teleportation functions $\mathbf{v}(t)$. Bounding the deviation of these oscillatory PageRank values from the static PageRank vector involves solving a *PageRank problem with complex teleportation* [Horn and Serra-Capizzano 07, Constantine and Gleich 10]. This result is,

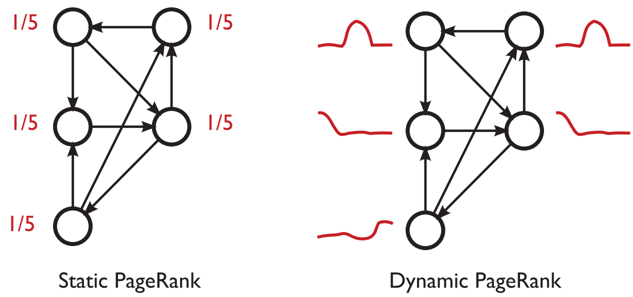


Figure 1. On the left, we have PageRank with static teleportation. At each step, the teleportation is to each node with uniform probability 1/5. On the right, we have the PageRank model with dynamic teleportation. In this case, the teleportation distribution (illustrated to the side of each node) *changes with time*. Thus, the upper nodes are teleported to more frequently during the middle time regime. In both cases, the graph is fixed. We study the effect of such dynamic teleportation on the PageRank scores.

perhaps, the first nonanalytical use of PageRank with a complex teleportation parameter.

In our new dynamical system, we do not compute a single ranking vector as others have done with time-dependent rankings [Grindrod et al. 11]. Rather, we compute a time-dependent ranking function $\mathbf{x}(t)$, the dynamic PageRank vector at time t, from which we can extract different static rankings (Section 2.4). There are two complications that arise from the use of empirically measured data. First, we must choose a time scale for our ODE based on the period of our page-view data (Section 2.5). Put a bit informally, we must choose the time unit for our ODE; it is not dimensionless. We show analytically that some choices of the time scale amount to solving the PageRank system for each change in the teleportation vector. Second, we also investigate smoothing the measured page-view data (Section 2.6). To compute this dependent ranking function $\mathbf{x}(t)$, we discuss ordinary differential equation (ODE) integrators in Section 3.

We discuss the impact of these choices on two problems: page views from Wikipedia and a network from Twitter. We also investigate how the rankings extracted from our methods differ from those extracted by other static ranking measurements. We can use these rankings for a few interesting applications. Adding the dynamic PageRank scores to a prediction task decreases the average error (Section 6.1) for Twitter. Clustering the dynamic PageRank scores yields many of the standard time-series features in social networks (Section 6.2).

$\imath$	the imaginary unit
n	number of nodes in a graph
$\mathbf{e}$	the vector of all ones
$\boldsymbol{P}$	column stochastic matrix
α	damping parameter in PageRank
$\mathbf{v}$	teleportation distribution vector
$\mathbf{x}$	solution to the PageRank computation: $(\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x} = (1-\alpha)\mathbf{v}$
$\mathbf{x}(t)$	solution to the dynamic PageRank computation for time t
$\mathbf{v}(t)$	a teleportation distribution vector at time t
$\mathbf{c}$	the cumulative rank function
$\mathbf{r}$	the variance rank function
$\mathbf{d}$	the difference rank function
$\mathbf{v}_k$	the teleportation distribution for the kth observed page-views vector
θ	decay parameter for time-series smoothing
s	the time scale of the dynamical system
$t_{\max}$	the last time of the dynamical system

Table 1. Summary of notation. Matrices (e.g., $\boldsymbol{P}$) are denoted by boldface uppercase italic letters; vectors (e.g., $\mathbf{e}$) are denoted by boldface lowercase roman letters; scalars (e.g., s and θ) are nonboldface italic or Greek letters. Indexed elements are vectors if they are in boldface (e.g., $\mathbf{v}_k$) and scalars if not (e.g., $t_{\max}$).

Finally, using Granger causality testing on dynamic PageRank scores helps us find a set of interesting links in the graph (Section 6.3).

We have made our code and data available in the spirit of reproducible research.[1]

2. PageRank with Time-Dependent Teleportation

We begin our discussion by summarizing in Table 1 the notation introduced thus far. In order to incorporate changes in the teleportation into a new model for PageRank, we begin by reformulating the standard PageRank algorithm in terms of changes to the PageRank values for each page. This step allows us to state PageRank as a dynamical system, in which case we can easily incorporate changes into the vector.

The standard PageRank algorithm is the power method for the PageRank Markov chain [Langville and Meyer 06]. After simplifying this iteration by

[1] Available at http://www.cs.purdue.edu/homes/dgleich/codes/dynsyspr-im.

assuming that $\mathbf{e}^T\mathbf{x} = 1$, it becomes

$$\mathbf{x}^{(k+1)} = \alpha\boldsymbol{P}\mathbf{x}^{(k)} + (1-\alpha)\mathbf{v}.$$

In fact, this iteration is equivalent to the Richardson iteration for the PageRank linear system $(\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x} = (1-\alpha)\mathbf{v}$. This fact is relevant because the Richardson iteration is usually defined as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(x)} + \omega\left[(1-\alpha)\mathbf{v} - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}^{(k)}\right].$$

For $\omega = 1$, we have

$$\Delta\mathbf{x}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} = \alpha\boldsymbol{P}\mathbf{x}^{(k)} + (1-\alpha)\mathbf{v} - \mathbf{x}^{(k)} = (1-\alpha)\mathbf{v} - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}^{(k)}.$$

Thus, changes in the PageRank values at a node evolve, based on the increment $(1-\alpha)\mathbf{v} - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}^{(k)}$. We reinterpret this update as a continuous-time dynamical system

$$\mathbf{x}'(t) = (1-\alpha)\mathbf{v} - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}(t). \tag{2.1}$$

To define the PageRank problem with time-dependent teleportation, we make $\mathbf{v}(t)$ a function of time.

Definition 2.1. The dynamic PageRank model with time-dependent teleportation is the solution of

$$\mathbf{x}'(t) = (1-\alpha)\mathbf{v}(t) - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}(t), \tag{2.2}$$

where $\mathbf{x}(0)$ is a probability distribution vector and $\mathbf{v}(t)$ is a probability distribution vector for all t.

In the dynamic PageRank model, the PageRank values $\mathbf{x}(t)$ may not "settle" or converge to some fixed vector $\mathbf{x}$. We see this as a feature of the new model, since we plan to utilize information from the evolution and changes in the PageRank values. For instance, in Section 2.4, we discuss various functions of $\mathbf{x}(t)$ that define a rank. Next, we state the solution of the problem.

Lemma 2.2. *The solution of the dynamical system*

$$\mathbf{x}'(t) = (1-\alpha)\mathbf{v}(t) - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}(t)$$

is

$$\mathbf{x}(t) = \exp[-(\boldsymbol{I} - \alpha\boldsymbol{P})t]\mathbf{x}(0) + (1-\alpha)\int_0^t \exp[-(\boldsymbol{I} - \alpha\boldsymbol{P})(t-\tau)]\mathbf{v}(\tau)\,d\tau.$$

This result is found in standard texts on dynamical systems, for example [Berman et al. 89].

Given this solution, let us quickly verify a few properties of this system.

Lemma 2.3. *The solution of a dynamical PageRank system* $\mathbf{x}(t)$ *is a probability distribution* ($\mathbf{x}(t) \geq 0$ *and* $\mathbf{e}^T\mathbf{x}(t) = 1$) *for all* t.

Proof. The model requires that $\mathbf{x}(0)$ be a probability distribution. Thus, $\mathbf{x}(0) \geq 0$ and $\mathbf{e}^T\mathbf{x}(0) = 1$. Assuming that the sum of $\mathbf{x}(t)$ is 1, then the sum of the derivative $\mathbf{x}'(t)$ is 0, as a quick calculation shows. The closed-form solution above is also nonnegative, because the matrix $\exp[-(\boldsymbol{I} - \alpha\boldsymbol{P})] = \exp(\alpha\boldsymbol{P})\exp(-1)$ is nonnegative, and both $\mathbf{x}(0)$ and $\mathbf{v}(t)$ are nonnegative for all t. (This property is known as exponential nonnegativity, and it is another property of M-matrices such as $\boldsymbol{I} - \alpha\boldsymbol{P}$ [Berman et al. 89].) $\square$

2.1. A Generalization of PageRank

This closed-form solution can be used to solve a version of the dynamic problem that reduces to the PageRank problem with static teleportation. If $\mathbf{v}(t) = \mathbf{v}$ is constant with respect to time, then

$$\int_0^t \exp[-(\boldsymbol{I} - \alpha\boldsymbol{P})(t-\tau)]\mathbf{v}(\tau)\,d\tau = (\boldsymbol{I} - \alpha\boldsymbol{P})^{-1}\mathbf{v} \\ - \exp[-(\boldsymbol{I} - \alpha\boldsymbol{P})t](\boldsymbol{I} - \alpha\boldsymbol{P})^{-1}\mathbf{v}.$$

Hence, for constant $\mathbf{v}(t)$,

$$\mathbf{x}(t) = \exp[-(\boldsymbol{I} - \alpha\boldsymbol{P})t](\mathbf{x}(0) - \mathbf{x}) + \mathbf{x},$$

where $\mathbf{x}$ is the solution to static PageRank: $(\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x} = (1-\alpha)\mathbf{v}$. Because all the eigenvalues of $-(\boldsymbol{I} - \alpha\boldsymbol{P})$ are less than 0, the matrix exponential terms disappear over a sufficiently long time horizon. Thus, when $\mathbf{v}(t) = \mathbf{v}$, nothing has changed. We recover the original PageRank vector $\mathbf{x}$ as the steady-state solution

$$\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}, \text{ the PageRank vector.}$$

This derivation shows that dynamic teleportation PageRank is a generalization of the PageRank vector. We summarize this discussion in the following theorem.

Theorem 2.4. *PageRank with time-dependent teleportation is a generalization of PageRank. If* $\mathbf{v}(t) = \mathbf{v}$, *then the solution of the ordinary differential equation*

$$\mathbf{x}'(t) = (1-\alpha)\mathbf{v} - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}(t)$$

converges to the PageRank vector

$$(\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x} = (1-\alpha)\mathbf{v}$$

as $t \to \infty$.

2.2. Choosing the Initial Condition

There are three natural choices for the initial condition $\mathbf{x}(0)$. The first choice is the uniform vector $\mathbf{x}(0) = \frac{1}{n}\mathbf{e}$. The second choice is the initial teleportation vector $\mathbf{x}(0) = \mathbf{v}(0)$. And the third choice is the solution of the PageRank problem for the initial teleportation vector $(\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}(0) = (1-\alpha)\mathbf{v}(0)$. We recommend either of the latter two choices in order to generalize the properties of PageRank. Note that if $\mathbf{x}(0)$ is chosen to solve the PageRank system for $\mathbf{v}(0)$, then $\mathbf{x}(t) = \mathbf{x}$ for all t is the solution of the PageRank dynamical system with constant teleportation (Theorem 2.4).

2.3. PageRank with Fluctuating Interest

One of the advantages of the PageRank dynamical system is that we can study problems analytically. We now do so with the following teleportation function, or forcing function as it would be called in the dynamical systems literature:

$$\mathbf{v}(t) = \frac{1}{k}\sum_{j=1}^{k} \mathbf{v}_j \left(\cos\left(t + (j-1)\frac{2\pi}{k}\right) + 1\right),$$

where $\mathbf{v}_j$ is a teleportation vector. Here, the idea is that $\mathbf{v}_j$ represents the propensity of people to visit certain nodes at different times. To be concrete, we might have $\mathbf{v}_1$ correspond to news websites that are visited more frequently during the morning, $\mathbf{v}_2$ correspond to websites visited at work, and $\mathbf{v}_3$ correspond to websites visited during the evening. This function has all the required properties that we need for it to be a valid teleportation function. At the risk of being overly formal, we shall state these properties as a lemma.

Lemma 2.5. *Let* $k \geq 2$. *Let* $\mathbf{v}_1, \ldots, \mathbf{v}_k$ *be probability distribution vectors. The time-dependent teleportation function*

$$\mathbf{v}(t) = \frac{1}{k}\sum_{j=1}^{k} \mathbf{v}_j \left(\cos\left(t + (j-1)\frac{2\pi}{k}\right) + 1\right)$$

satisfies the two properties

1. $\mathbf{v}(t) \geq 0$ *for all* t,
2. $\sum_{i=1}^{n} v(t)_i = 1$ *for all* t.

Proof. The first property follows directly, since the minimum value of the cosine function is -1, and thus $\mathbf{v}(t)$ is always nonnegative. The second property is also straightforward. Note that

$$\sum_{i=1}^{n} v(t)_i = 1 + \sum_{j=1}^{k} \cos\left(t + (j-1)\frac{2\pi}{k}\right) = 1 + \sum_{j=1}^{k} \Re\left\{\exp\left(\imath t + \imath(j-1)\frac{2\pi}{k}\right)\right\}.$$

Let $r_j(t) = \exp(\imath t + \imath(j-1)2\pi/k)$. For $t = 0$, these terms express the kth roots of unity. For any other t, we simply rotate these roots. Thus we have $\sum_j r_j(t) = 0$ for every t, because the sum of the kth roots of unity is 0 if $k \geq 2$. The second property now follows because the sum of the real component is still zero. $\square$

For this function, we can solve for the steady-state solution analytically.

Lemma 2.6. *Let* $k \geq 2$, $0 \leq \alpha < 1$, $\boldsymbol{P}$ *be column-stochastic,* $\mathbf{v}_1, \ldots, \mathbf{v}_k$ *be probability distribution vectors, and*

$$\mathbf{v}(t) = \frac{1}{k}\sum_{j=1}^{k} \mathbf{v}_j \left(\cos\left(t + (j-1)\frac{2\pi}{k}\right) + 1\right) = \frac{1}{k}\boldsymbol{V}\cos(t + \mathbf{f}) + \frac{1}{k}\boldsymbol{V}\mathbf{e},$$

where $\boldsymbol{V} = \begin{bmatrix}\mathbf{v}_1, \ldots, \mathbf{v}_k\end{bmatrix}$ *and* $f_j = (j-1)2\pi/k$, $j = 1, \ldots, k$. *Then the steady-state solution of*

$$\mathbf{x}'(t) = (1-\alpha)\mathbf{v}(t) - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}(t)$$

is

$$\mathbf{x}(t) = \mathbf{x} + \Re\left\{\mathbf{s}\exp(\imath t)\right\},$$

where $\mathbf{x}$ *is the solution of the static PageRank problem*

$$(\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x} = (1-\alpha)\frac{1}{k}\boldsymbol{V}\mathbf{e},$$

and $\mathbf{s}$ *is the solution of the static PageRank problem with* complex teleportation

$$\left(\boldsymbol{I} - \frac{\alpha}{1+\imath}\boldsymbol{P}\right)\mathbf{s} = (1-\alpha)\frac{1}{k(1+\imath)}\boldsymbol{V}\exp(\imath\mathbf{f}).$$

Proof. This proof is mostly a derivation of the expression for the solution by guessing the form. First note that if

$$\mathbf{x}(t) = \mathbf{x} + \mathbf{y}(t),$$

then

$$\mathbf{y}'(t) = (1-\alpha)\frac{1}{k}\boldsymbol{V}\cos(t+\mathbf{f}) - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{y}(t).$$

That is, we have removed the constant term from the teleportation function by looking at solutions centered on the static PageRank solution. To find the steady-state solution, we look at the complex-phasor problem

$$\mathbf{z}'(t) = (1-\alpha)\frac{1}{k}\boldsymbol{V}\exp(\imath t + \imath\mathbf{f}) - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{z}(t),$$

where $\mathbf{y}(t) = \Re\{\mathbf{z}(t)\}$. Suppose that $\mathbf{z}(t) = \mathbf{s}\exp(\imath t)$. Then

$$\mathbf{z}'(t) = \imath\mathbf{s}\exp(\imath t) = (1-\alpha)\frac{1}{n}\exp(\imath t)\exp(\imath\mathbf{f}) - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{s}\exp(\imath t).$$

The statement of $\mathbf{s}$ in the theorem is exactly the solution after the phasor $\exp(\imath t)$ is canceled. We now have to show that this solution is well defined. PageRank with a complex teleportation parameter γ exists for every column-stochastic $\boldsymbol{P}$ if $|\gamma| < 1$ (see [Horn and Serra-Capizzano 07, Constantine and Gleich 10]). For the problem defining $\mathbf{s}$, we have $\gamma = \alpha/(1+\imath)$ and $|\gamma| = \alpha/\sqrt{2}$. Thus, such a vector $\mathbf{s}$ always exists. □

We conclude with an example of this theorem. Consider a four-node graph with adjacency matrix and transition matrix given by

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{P} = \begin{bmatrix} 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0.5 & 0.5 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0.5 & 0. \end{bmatrix}$$

Let $\mathbf{v}_j = \mathbf{e}_j$ for $j = 1, \ldots, 4$. That is, interest oscillates among all four nodes in the graph in a regular fashion. We show the evolution of the dynamical system for 20 time units in Figure 2. This evolution quickly converges to the oscillators predicted by the lemma. In the interest of simplifying the plot, we do not show the exact curves, since they are visually indistinguishable from those plotted for $t \geq 4$. By solving the complex-valued PageRank to compute $\mathbf{s}$, we can compute the magnitude of the fluctuation:

$$|\mathbf{s}| = [0.0216 \quad 0.0261 \quad 0.0122 \quad 0.0235]^T.$$

This vector accurately captures the magnitude of these fluctuations.

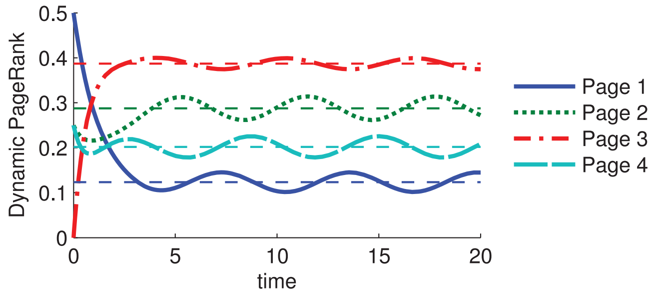


Figure 2. The dashed lines represent the average PageRank vector computed for the teleportation vectors. The curves show the evolution of the PageRank dynamical system for this example of teleportation. We see that the dynamic PageRanks fluctuate about their average PageRank vectors. Lemma 2.6 predicts the magnitude of the fluctuation.

2.4. Ranking from Time Series

The above equations provide a time series of dynamic PageRank vectors for the nodes, denoted formally by $\mathbf{x}(t)$, $0 \leq t \leq t_{\max}$. Applications, however, often want a single score, or small set of scores, to characterize sets of interesting nodes. There are a few ways in which these time series give rise to scores. Many of these methods were explained in [O'Madadhain and Smyth 05] in the context of ranking sequences of vectors. Having a variety of different scores derived from the same data frequently helps in using those scores as features in a prediction or learning task [Becchetti et al. 08, Constantine and Gleich 10].

2.4.1. Transient Rank. We call the instantaneous values of $\mathbf{x}(t)$ a node's *transient rank*. This score gives the importance of a node at a particular time.

2.4.2. Summary, Variance, and Cumulative Rank. Any summary function s of the time series, such as the integral, average, minimum, maximum, or variance, is a single score that encompasses the entire interval $[0, t_{\max}]$. We utilize the *cumulative rank* $\mathbf{c}$ and *variance rank* $\mathbf{r}$ in the forthcoming experiments:

$$\mathbf{c} = \int_0^{t_{\max}} \mathbf{x}(t)\, dt \quad \text{and} \quad \mathbf{r} = \int_0^{t_{\max}} \left(\mathbf{x}(t) - \frac{1}{t_{\max}} \mathbf{c} \right)^2 dt.$$

2.4.3. Difference Rank. A node's *difference rank* is the difference between its maximum and minimum ranks over all time, or a limited time window:

$$\mathbf{d} = \max_t [\mathbf{x}(t)] - \min_t [\mathbf{x}(t)], \quad \mathbf{d}_W = \max_{t \in W} \mathbf{x}(t) - \min_{t \in W} \mathbf{x}(t).$$

Nodes with high difference rank should reflect important events that occurred within the range $[0, t_{\max}]$ or the time window W. We suggest using a window

W that omits the initial convergence region of the evolution. In the context of Figure 2, we would set W to be $[4, 20]$ to approximate the vector $|\mathbf{s}|$ numerically. In Section 6 and Figure 6, we see examples of how current news stories arise as articles with high difference rank.

2.5. Modeling Activity

In the next two sections of our introduction to the dynamic teleportation PageRank model, we discuss how to incorporate empirically measured activity into the model. Let $\mathbf{p}_1, \ldots, \mathbf{p}_k$ be k observed vectors of activity for a website. In the cases we examine below, these activity vectors measure page views per hour on Wikipedia and the number of tweets per month on Twitter. We normalize each of them into teleportation distributions, and conceptually think of the collection of vectors as a matrix

$$\mathbf{v}_1, \ldots, \mathbf{v}_k \to \boldsymbol{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k].$$

Let $\mathbf{e}(i)$ be a functional form representing the vector $\mathbf{e}_i$. The time-dependent teleportation vector we create from these data is

$$\mathbf{v}(t) = \boldsymbol{V}\mathbf{e}(\lfloor t \rfloor + 1) = \mathbf{v}_{\lfloor t \rfloor + 1}.$$

For this choice, the time units of our dynamical system are given by the time unit of the original measurements. Other choices are possible too. Consider

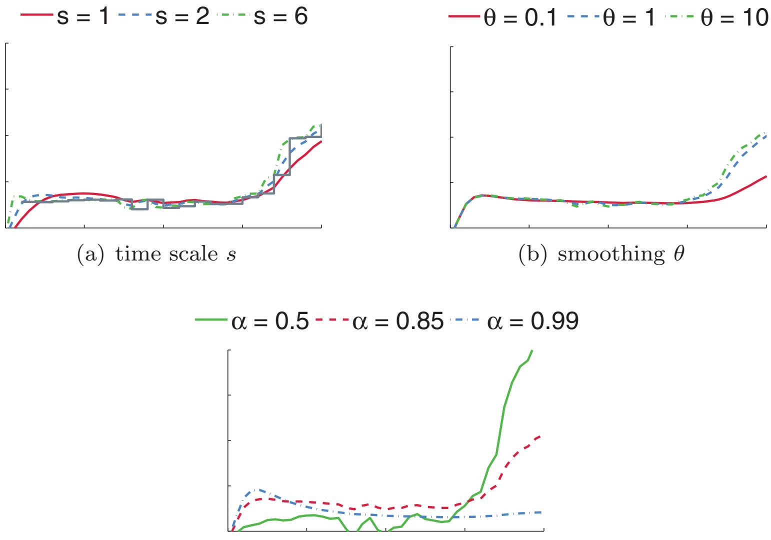
$$\mathbf{v}_s(t) = \boldsymbol{V}\mathbf{e}(\lfloor t/s \rfloor + 1) = \mathbf{v}_{\lfloor t/s \rfloor + 1}.$$

If $s > 1$, then time in the dynamical system slows down. If $s < 1$, then time accelerates. Thus, we call s the time scale of the system. Note that

$$\mathbf{x}(sj), \quad j = 0, 1, \ldots,$$

represents the same effective time point for every time scale. Thus, when we wish to compare different time scales s, we examine the solution at such scaled points.

In the experimental evaluation, the parameter s plays an important role. We illustrate its effect in Figure 3(a) for a small subnetwork extracted from Wikipedia. As we discuss further in Section 3, for large values of s, $\mathbf{v}(t)$ looks constant for long periods of time, and hence $\mathbf{x}(t)$ begins to converge to the PageRank vector for the current, and effectively static, teleportation vector. Thus, we also plot the converged PageRank vectors as a step function. We see that as s increases, the lines converge to these step functions, but for $s = 1$ and $s = 2$, they behave differently.



(a) time scale s

(b) smoothing θ

(c) damping parameter α

Figure 3. The evolution of PageRank values for one node due to dynamical teleportation. The horizontal axis is time $[0, 20]$, and the vertical axis encompasses the interval $[0.01, 0.014]$. In figure (a), $\alpha = 0.85$, and we vary the time-scale parameter (Section 2.5) with no smoothing. The solid dark line corresponds to the step function of solving PageRank exactly at each change in the teleportation vector. All samples are taken from the same effective time points as discussed in the text. In figure (b), we vary the smoothing (Section 2.6) of the teleportation vectors with $s = 2$ and $\alpha = 0.85$. In figure (c), we vary α with $s = 2$ and there is no smoothing. We used the `ode45` function in MATLAB, a Runge–Kutta method, to evolve the system.

2.6. Smoothing Empirical Activity

So far, we have defined a time-dependent $\mathbf{v}(t)$ that changes at fixed intervals based on empirically measured data. A better idea is to smooth out these "jumps" using an exponentially weighted moving average. As a continuous function of time, this yields

$$\bar{\mathbf{v}}'(t;\theta) = \theta\mathbf{v}(t) - \theta\bar{\mathbf{v}}(t;\theta).$$

To understand why this smooths the sequence, consider an implicit Euler approximation

$$\bar{\mathbf{v}}(t) = \frac{1}{1+h\theta}\bar{\mathbf{v}}(t-h;\theta) + \frac{h\theta}{1+h\theta}\mathbf{v}(t).$$

This update can be written more simply as

$$\bar{\mathbf{v}}(t;\theta) = \underbrace{\gamma \mathbf{v}(t)}_{\text{new data}} + \underbrace{(1-\gamma)\bar{\mathbf{v}}(t-h;\theta)}_{\text{old data}},$$

where $\gamma = h\theta/(1+h\theta)$. When $\mathbf{v}(t)$ changes at fixed intervals, then $\bar{\mathbf{v}}(t;\theta)$ will slowly change. If θ is small, then $\bar{\mathbf{v}}(t;\theta)$ changes slowly. We recover the "jump" changes in $\mathbf{v}(t)$ in the limit $\theta \to \infty$.

The effect of θ is shown in Figure 3(b). Note that we quickly recover behavior that is effectively the same as using jumps in $\mathbf{v}(t)$ ($\theta = 1, 10$). So we expect changes with smoothing only for $\theta < 1$.

2.7. Choosing the Teleportation Factor

Choosing α even for static PageRank problems is challenging; see [Gleich et al. 10] and [Constantine and Gleich 10] for some discussion. In this manuscript, we do not perform any systematic study of the effects of α beyond Figure 3(c). This simple experiment shows one surprising feature. Common wisdom for choosing α in the static case suggests that as α approaches 1, the vector becomes more sensitive. For the dynamic teleportation setting, however, the opposite is true. Small values of α produce solutions that more closely reflect the teleportation vector—the quantity that is changing—whereas large values of α reflect the graph structure, which is invariant with time. Hence, with dynamic teleportation, using a small value of α is the sensitive setting. Note that this observation is a straightforward conclusion from the equations of the dynamic vector

$$\mathbf{x}'(t) = (1-\alpha)\mathbf{v}(t) + \alpha \boldsymbol{P}\mathbf{x}(t) - \mathbf{x}(t),$$

so α small implies a larger change due to $\mathbf{v}(t)$. Nevertheless, we found it surprising in light of the existing literature.

3. Methods for Dynamic PageRank

In order to compute the time sequence of PageRank values $\mathbf{x}(t)$, we can evolve the dynamical system (2.2) using any standard method—usually called an integrator. We discuss both the forward Euler method and a Runge–Kutta method next. Both methods, and indeed, the vast majority of dynamical system integrators, require only a means to evaluate the derivative of the system at a time t given $\mathbf{x}(t)$. For PageRank with dynamic teleportation, this corresponds to computing

$$\mathbf{x}'(t) = (1-\alpha)\mathbf{v}(t) - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}(t).$$

The dominant cost in evaluating $\mathbf{x}'(t)$ is the matrix–vector product $\boldsymbol{P}\mathbf{x}$. For the explicit methods we explore, all of the other work is linear in the number of nodes, and hence these methods easily scale to large networks. Both of these methods may also be used in a distributed setting if a distributed matrix–vector product is available.

3.1. Forward Euler

We first discuss the forward Euler method. This method lacks high accuracy, but is fast and straightforward. Forward Euler approximates the derivative with a first-order Taylor approximation,

$$\mathbf{x}'(t) \approx \frac{\mathbf{x}(t+h) - \mathbf{x}(t)}{h},$$

and then uses that approximation to estimate the value at a short time step in the future:

$$\mathbf{x}(t+h) = \mathbf{x}(t) + h\left[(1-\alpha)\mathbf{v}(t) - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}(t)\right].$$

This update is the original Richardson iteration with $h = \omega$. We present the forward Euler method as a formal algorithm as our Algorithm 1 in order to highlight a comparison with the power and Richardson methods. That is, the forward Euler method is simply running a power method but changing the vector $\mathbf{v}$ at every iteration. However, we derived this method based on evolving (2.2). Thus, by studying the relationship between (2.2) and Algorithm 1, we can understand the underlying problem solved by changing the teleportation vector while running the power method.

3.1.1. Long Time Scales. Using the forward Euler method, we can analyze the situation with a large time-scale parameter s. Consider an arbitrary $\mathbf{x}(0)$, $\alpha = 0.85$, $s = 100$, $h = 1$, and no smoothing. In this case, the forward Euler method will run the Richardson iteration 100 times before observing the change in $\mathbf{v}(t)$ at $t = 100$. The difference between $\mathbf{x}(k)$ and the exact PageRank solution for this temporarily static $\mathbf{v}(t)$ is $\|\mathbf{x}(k) - \mathbf{x}\|_1 \leq 2\alpha^k$. For $k > 50$, this difference is small. Thus, a large s and no smoothing corresponds to solving the PageRank problem for each change in $\mathbf{v}$.

3.1.2. Stability. The forward Euler method with time step h is stable if the eigenvalues of the matrix $-h(\boldsymbol{I} - \alpha\boldsymbol{P})$ are within distance 1 of the point -1. The eigenvalues of $\boldsymbol{P}$ are all between -1 and 1 because it is a stochastic matrix, and so this is stable for every $h < 2/(1+\alpha)$.

Algorithm 1. The forward Euler method for evolving the dynamical system: $\mathbf{x}'(t) = (1-\alpha)\mathbf{v}(t) - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}(t)$. The resulting procedure looks remarkably similar to the standard Richardson iteration to compute a PageRank vector. One key difference is that there is no notion of convergence.

Input:
a graph $G = (V, E)$ and a procedure to compute $\boldsymbol{P}\mathbf{x}$ for this graph
a maximum time $t_{\max}$
a function to return $\mathbf{v}(t)$ for any $0 \le t \le t_{\max}$
a damping parameter α
a time step h
Output: $\boldsymbol{X}$, where the kth column of $\boldsymbol{X}$ is $\mathbf{x}(0 + kh)$ for all $1 \le k \le t_{\max}/h$ (or any desired subset of these values)

$t \leftarrow 0$; $k = 1$
$\mathbf{x}(0) \leftarrow \mathbf{v}(0)$ (or any other desired initial condition)
while $t \le t_{\max} - h$ **do**
  $\mathbf{x}(t+h) \leftarrow \mathbf{x}(t) + h\,[(1-\alpha)\mathbf{v}(t) - (\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}(t)]$
  $\boldsymbol{X}(:, k) \leftarrow \mathbf{x}(t+h)$
  $t \leftarrow t + h$; $k \leftarrow k + 1$

3.2. Runge–Kutta

Runge–Kutta [Runge 95, Kutta 01] numerical schemes are some of the most familiar and most widely used. They achieve far greater accuracy than the simple forward Euler method, at the expense of a greater number of evaluations of the function $\mathbf{x}'(t)$ at each step. We use the implementations of Runge–Kutta methods available in the MATLAB ODE suite [Shampine and Reichelt 97]. The step size is adapted automatically based on a local error estimate, and the solution can be evaluated at any desired point in time. The stability region for Runge–Kutta includes the region for forward Euler, so these methods are stable. These methods are also fast. To integrate the system for Wikipedia with over 4 million vertices and 60 million edges, it took between 300 and 600 seconds, depending on the parameters.

3.3. Maintaining Interpretability

Based on the theory of the dynamic teleportation system, we expect that $\mathbf{x}(t) \ge 0$ and $\mathbf{e}^T\mathbf{x}(t) = 1$ for all time. Although this property should be true of the computed solution, we often find that the sum diverges from 1. Consequently, for our experiments, we include a correction term

$$\mathbf{x}'(t) = (1-\alpha)\mathbf{v}(t) - (\gamma\boldsymbol{I} - \alpha\boldsymbol{P})\mathbf{x}(t),$$

where $\gamma = (1-\alpha)\mathbf{e}^T\mathbf{v}(t) + \alpha\mathbf{e}^T\mathbf{x}(t)$. Note that $\gamma = 1$ if $\mathbf{x}(t)$ has sum exactly 1. If $\mathbf{e}^T\mathbf{x}(t)$ is slightly different from 1, then the correction with γ ensures that

$\mathbf{e}^T\mathbf{x}'(t) = 0$ numerically. Similar issues arise in computing static PageRank [Wills and Ipsen 09], although the additional computation in the Runge–Kutta methods exacerbates the problem.

4. Related Work

Note that we previously studied this idea in a conference paper [Rossi and Gleich 12]. These ideas have been significantly refined for this manuscript.

The relationship between dynamical systems and classical iterative methods has been utilized by [Embree and Lehoucq 09] to study eigenvalue solvers. It was also noted in the early paper [Tsaparas 04] that there is a relationship between the PageRank and HITS algorithms and dynamical systems.

In the past, others studied PageRank approximations on graph streams [Das Sarma et al. 08]. More recently, [Bahmani et al. 12] studied how accurately an evolving PageRank method could estimate the true PageRank of an evolving graph that is accessed only via a crawler. The method used here solved each PageRank problem exactly for the current estimate of the underlying graph. A detailed study of how PageRank values evolve during a web crawl was done in [Boldi et al. 05]. In the future, we plan to study dynamic graphs via similar ideas.

As explained in Section 3 and Algorithm 1, our proposed method is related to changing the teleportation vector in the power method as it is being computed. It was noted in [Bianchini et al. 05] that the power method will still converge if either the graph or the vector $\mathbf{v}$ changes a few times during the method, albeit to a new solution given by the new vector or graph. Our method capitalizes on a closely related idea, and we use the intermediate quantities explicitly. Another related idea is the online page importance computation (OPIC) [Abiteboul et al. 03], which integrates a PageRank-like computation along with a crawling process. The method does nothing special if a node has changed when it is crawled again.

While we described PageRank in terms of a random-surfer model, another characterization of PageRank is that it is a sum of damped transitions:

$$\mathbf{x} = (1-\alpha)\sum_{k=0}^{\infty}(\alpha\boldsymbol{P})^k\mathbf{v}.$$

These transitions are a type of probabilistic walk, and [Grindrod et al. 11] introduced the related notion of dynamic walks for dynamic graphs. We can interpret these dynamic walks as a *backward Euler* approximation to the dynamical system:

$$\mathbf{x}'(t) = \alpha \boldsymbol{A}(t)\mathbf{x}(t), \quad \mathbf{x}(0) = \mathbf{e},$$

with time step $h = 1$ and $\boldsymbol{A}$ a time-dependent adjacency matrix. This relationship suggests that there may be a range of interesting models between our dynamic teleportation model and existing evolving graph models.

Outside of the context of web-ranking, [O'Madadhain and Smyth 05] proposes EventRank, a method of ranking nodes in dynamic graphs, which uses the PageRank propagation equations for a sequence of graphs. We use the same idea but place it within the context of a continuous dynamical system. In the context of popularity dynamics [Ratkiewicz et al. 10], our method captures how changes in external interest influence the popularity of nodes and the nodes linked to those nodes in an implicit fashion. Our work is also related to modeling human dynamics, namely, how humans change their behavior when exposed to rapidly changing or unfamiliar conditions [Bagrow et al. 11]. In one instance, our method shows the important topics and ideas relevant to humans before and after one of the largest Australian earthquakes (Figure 6).

In closing, we wish to note that our proposed method does not involve updating the PageRank vector, a related problem that has received considerable attention [Chien et al. 04, Langville and Meyer 04]. Nor is it related to tensor methods for dynamic graph data [Sun et al. 06, Dunlavy et al. 11].

5. Examples of Dynamic Teleportation

We now use dynamic teleportation to investigate page view patterns on Wikipedia and user activity on Twitter. In the following experiments, unless otherwise noted, we set $s = 1$, $\alpha = 0.85$, do not use smoothing ("$\theta = \infty$"), and use the `ode45` method from MATLAB to evolve the system. We study this model on two datasets.

5.1. Datasets

We provide some basic statistics of the Wikipedia and Twitter datasets in Table 2. For Wikipedia, the time unit for $s = 1$ is an hour, and for Twitter, it is one month.

5.1.1. Wikipedia Article Graph and Hourly Page Views. Wikipedia provides access to copies of its database. We downloaded a copy of its database on March 6, 2009, and extracted an article-by-article link graph, where an article is a page in the main

Dataset	Nodes	Edges	$t_{\max}$	Period	Average p_i	Max p_i
WIKIPEDIA	4 143 840	72 718 664	48	hours	1.4243	353 799
TWITTER	465 022	835 424	6	months	0.5569	1056

Table 2. Dataset properties. The page views for each page on Wikipedia or total tweets for each user on twitter is denoted by $\mathbf{p}$, and we show the maximum and average for any page at any time.

Wikipedia namespace, a category page, or a portal page.[2] All other pages and links were removed. See [Gleich et al. 07] for more information.

Wikipedia also provides hourly page views for each page.[3] These are the number of times a page was viewed in a given hour. These are not unique visits. We downloaded the raw page counts and matched the corresponding page counts to the pages in the Wikipedia graph. We used the page counts starting from March 6, 2009, and moving forward in time. Although it would seem as though measuring page views would correspond to measuring $\mathbf{x}(t)$ instead of $\mathbf{v}(t)$, one of our earlier studies showed that users hardly ever follow links on Wikipedia [Gleich et al. 10]. Thus, we can interpret these page views as a reasonable measure of external interest in Wikipedia pages.

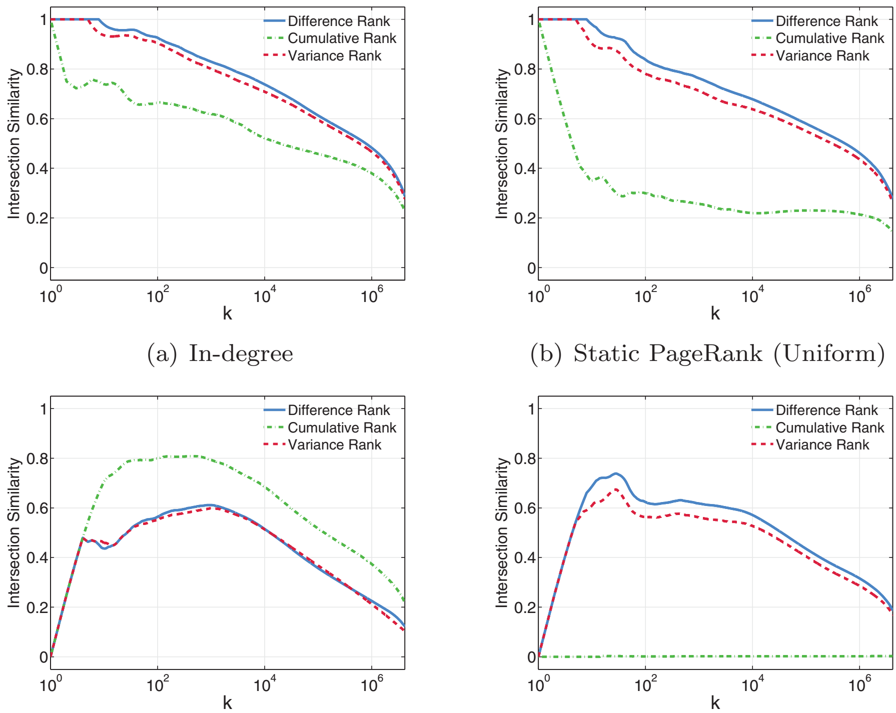
5.1.2. Twitter Social Network and Monthly Tweet Rates. The Twitter social network consists of a set of users that *follow* each other's tweets, or small 140-character messages. Thus, Twitter has both a network structure, the follower graph, and activity on top of this graph, the tweet stream. We built the follower graph by starting with a few seed users and crawling follow links for several iterations. (The particular crawl we used is from 2008.) We then took the set of users from the follower graph and extracted their tweets for a period of six months between 2008 and 2009. Each tweet is time-stamped, and we construct a sequence of vectors to represent the number of tweets from each user in each month. (We briefly explored finer levels of granularity for Twitter activity, but these choices led to sparser vectors.) These vectors are the basis for the teleportation time series in our time-dependent PageRank.

5.2. Rankings from Transient Scores

First, we evaluate the rankings from dynamic PageRank using the intersection similarity measure [Boldi 05]. Given two vectors $\mathbf{x}$ and $\mathbf{y}$, the intersection

[2] See http://en.wikipedia.org/wiki/Wikipedia:Database_download.

[3] See http://dumps.wikimedia.org/other/pagecounts-raw/.



(a) In-degree (b) Static PageRank (Uniform)

(c) Average page views (d) Static PageRank (avg. page views)

Figure 4. Intersection similarity of rankings derived from dynamic PageRank. We compute the intersection similarity of the difference, variance, and cumulative rankings given by dynamic PageRank and compare these with the rankings given by the in-degree, average page views, static PageRank with uniform teleportation, and static PageRank with average page views as the teleportation vector. For dynamic PageRank, we set the initial value $\mathbf{x}(0)$ to be the solution of the static PageRank system, which uses $\mathbf{v}(0)$ as the teleportation vector.

similarity metric at k is the average symmetric difference over the top j sets for each $j \leq k$. If $\mathcal{X}_k$ and $\mathcal{Y}_k$ are the top k sets for $\mathbf{x}$ and $\mathbf{y}$, then

$$\operatorname{isim}_k(\mathbf{x}, \mathbf{y}) = \frac{1}{k} \sum_{j=1}^{k} \frac{|\mathcal{X}_j \Delta \mathcal{Y}_j|}{2j},$$

where Δ is the symmetric set-difference operation. Identical vectors have an intersection similarity of 0.

For the Wikipedia graph, Figure 4 shows the similarity profile comparing a few ranking measures from dynamic PageRank to reasonable baselines. In particular, we compare $\mathbf{d}, \mathbf{r}, \mathbf{c}$ (from Section 2.4) to in-degree, average page views, static PageRank with uniform teleportation, and static PageRank using average page views as the teleportation vector. The results suggest that dynamic PageRank is different from the other measures, even for small values of k. In particular, combining the external influence with the graph appears to produce something new. The only exception is in Figure 4(d), where the cumulative rank is shown to give a similar ordering to static PageRank using average page views as the teleportation.

5.3. Difference Ranks

Figures 5 and 6 show the time series of the top 100 pages by the difference measure for Wikipedia with $s = 1$ and $s = 4$ without smoothing. Many of these pages reveal the ability of dynamic PageRank to mesh the network structure with changes in external interest. For instance, in Figure 6, we find pages related to an Australian earthquake (43, 84, 82), the "recently" released movie "Watchmen" (98, 23–24), a famous musician who died (2, 75), recent "American Idol" gossip (34, 63), a remembrance of Eve Carson from a contestant on "American Idol" (88, 96, 34), news about the murder of a Harry Potter actor (60), and the Skittles social media mishap (94). These results demonstrate the effectiveness of dynamic PageRank to identify interesting pages that pertain to external interest. The influence of the graph results in the promotion of pages such as the Richter magnitude scale (84). That page was not in the top 200 from page views.

6. Applications of Time-Dependent Teleportation

This section explores the opportunity of using dynamic PageRank for a variety of applications outside of the context of ranking.

6.1. Predicting Future Page Views and Tweets

We begin by studying how well the dynamical system can predict the future. Formally, given a lagged time series $\mathbf{p}_{t-w}, \ldots, \mathbf{p}_{t-1}, \mathbf{p}_t$ [Ahmed et al. 10], the goal is to predict the future value $\mathbf{p}_{t+1}$ (actual page views or number of tweets). This type of temporal prediction task has many applications, such as actively adapting caches in large database systems or dynamically recommending pages.

Suppose that $\bar{\mathbf{f}}$ is a feature vector derived from $\mathbf{p}_t$ or any other information that should be correlated with $\mathbf{p}_t$. We performed one-step-ahead predictions $(t + 1)$

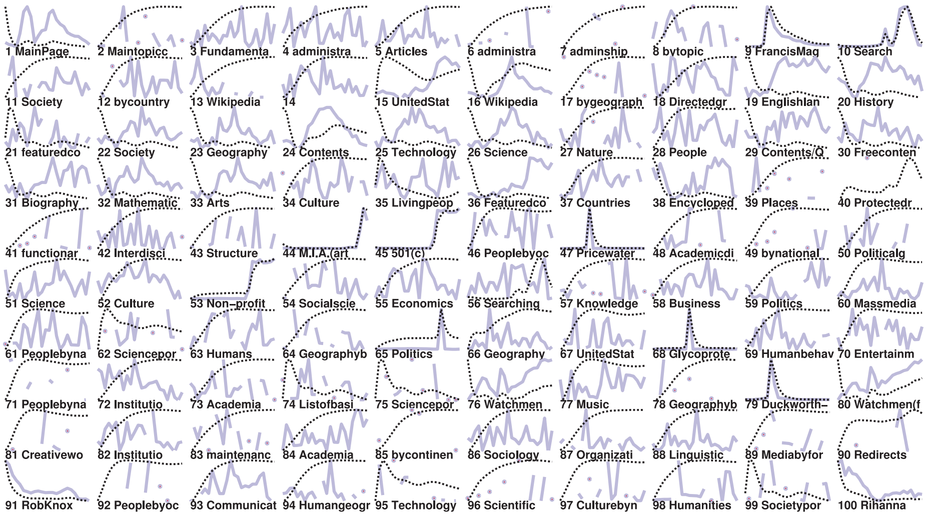


Figure 5. The top-100 Wikipedia pages that fluctuate the most as determined by the difference ranking from our dynamic PageRank approach. The dashed curves are the transient scores, and the light solid curves are the raw page-view data. The horizontal axis is 24 hours, and the vertical axes are normalized to show the range of the data.

using linear regression. That is, we learn a model of the form

$$\left[\bar{\mathbf{f}}(t-1)\ \bar{\mathbf{f}}(t-2)\ \ldots\ \bar{\mathbf{f}}(t-w)\right]\ \mathbf{b}\ \approx\ \mathbf{p}(t),$$

where w is the window size, and $\bar{\mathbf{f}}(\cdot)$ is either page views or both page views and transient scores. After fitting $\mathbf{b}$, we see that the model predicts $\mathbf{p}(t+1)$ to be

$$\left[\bar{\mathbf{f}}(t)\ \bar{\mathbf{f}}(t-1)\ \cdots\ \bar{\mathbf{f}}(t-w+1)\right]\ \mathbf{b}.$$

We use the symmetric mean absolute percentage error (sMAPE) [Ahmed et al. 10] measure to evaluate the prediction:

$$\text{sMAPE} = \frac{1}{|T|}\sum_{t=1}^{|T|}\frac{|p_t - \hat{p}_t|}{(p_t + \hat{p}_t)/2}.$$

This relative error measure averages all the relative prediction errors over all the time steps. We then average it over nodes.

There are three relevant details in how we applied this methodology. First, we created two models. The *base model* uses only the time series of page views or tweets as the feature vectors $\bar{f}$. The *dynamic teleportation model* uses both the

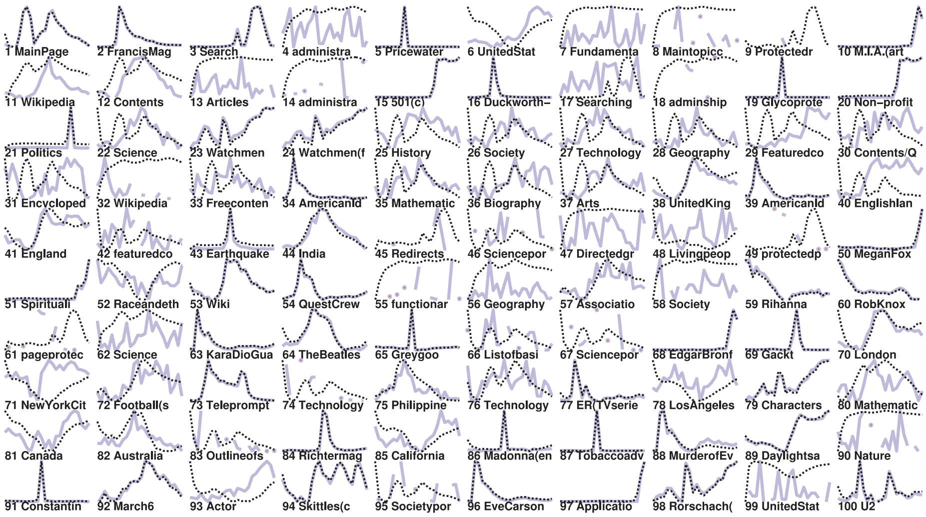


Figure 6. We again plot the top 100 pages from the difference rank, now using $s = 4$. The dashed curves are the transient scores, and the light solid curves are the raw page-view data. The horizontal axis is 24 hours, and the vertical axes are normalized to show the range of the data. This choice gives a similar ordering to that from our previous forward Euler iteration method from [Rossi and Gleich 12]. Note the large change between the transient scores for "MainPage" between this figure and Figure 5.

transient scores with smoothing and page views as the feature vectors $\bar{f}$. We sampled the transient scores once for each change in $\mathbf{v}(t)$ based on the raw data. In our final numbers, we looked at the relative decrease in error after adding these new features to indicate whether adding the dynamic teleportation information improved the prediction, that is, we checked whether the quantity dynamic teleportation sMAPE divided by baseline sMAPE was less than 1. Second, we built a model for blocks of 1000 nodes at a time and averaged the sMAPE scores over those predictions. Third, we varied w through all choices in order to demonstrate that our results are not sensitive to a particular choice. We did this by setting $w = 1$ and using a model learned on the first time step to predict the second; then we set $w = 2$ and used the first two time steps to predict the third, and so on. We repeated this until we had predicted all time steps (up to $t = t_{\max} - 1$ and $w = t_{\max} - 2$). Each choice of w resulted in a single sMAPE score, which we again averaged.

As an aside, we note that setting w to a fixed constant and using AIC to learn the value of w for each one-step-ahead prediction did not change our results. We

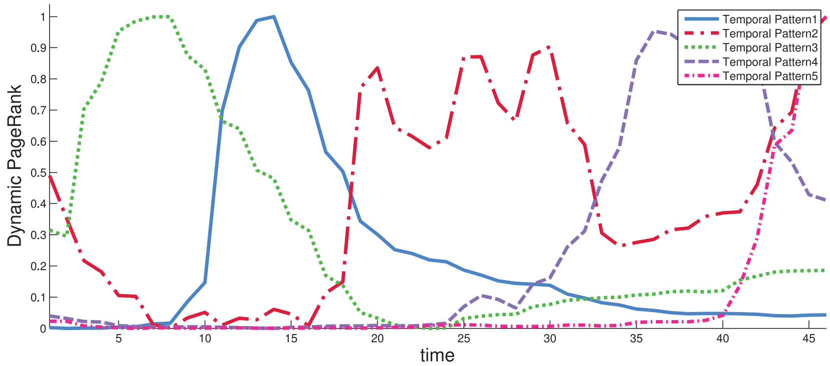
Dataset	Type	θ	Error Ratio s (timescale) 1	2	6	∞
TWITTER	stationary	**0.01**	0.450	0.898	0.836	0.967
		0.50	0.258	0.611	0.858	0.775
		1.00	0.527	0.531	0.849	0.791
	nonstationary	**0.01**	0.500	0.874	0.662	1.240
		0.50	0.461	0.499	0.658	0.835
		1.00	0.458	0.489	0.652	0.848
WIKIPEDIA	stationary	**0.01**	0.978	0.991	0.989	0.978
		0.50	1.140	1.130	1.004	0.990
		1.00	1.084	0.976	1.010	0.990
	nonstationary	**0.01**	0.968	1.011	0.968	1.004
		0.50	1.218	0.994	1.030	1.031
		1.00	1.241	0.996	0.957	0.998

Table 3. The ratio between the base model and the model with dynamic teleportation scores with $s = 1, 2, 6, \infty$ for three smoothing parameters. (Here, $s = \infty$ corresponds to solving the PageRank problem exactly for each change in teleportation.) If this ratio is less than 1, then the model with the dynamic teleportation scores improves the prediction performance. We also distinguish between prediction problems with highly volatile nodes (nonstationary) and nodes with relatively stable behavior (stationary). The results show a much stronger benefit for Twitter than for Wikipedia.

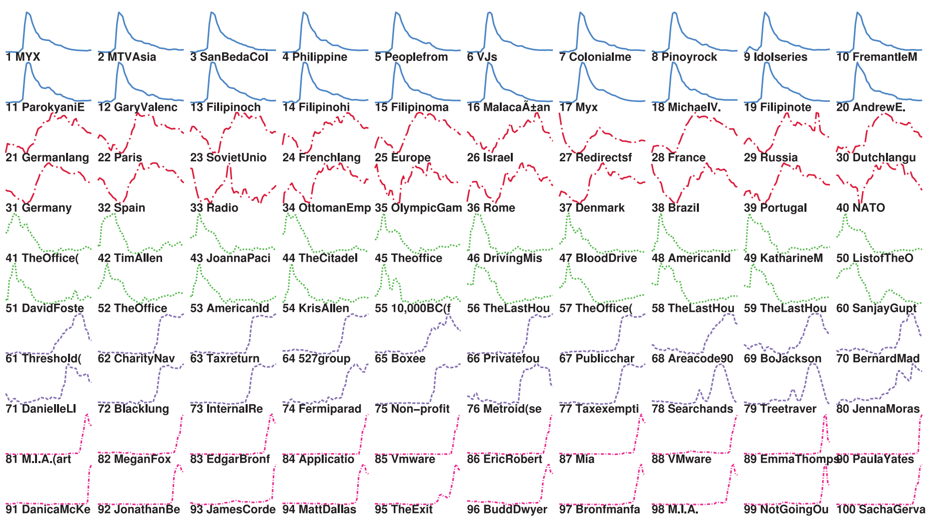
felt that using all possible w allowed us to better understand the effectiveness of using dynamic PageRank for prediction. For practical applications, there may be other benefits to choosing w carefully.

We evaluated these models for prediction on stationary and nonstationary time series. Informally, a time series is *weakly stationary* if it has properties (mean and covariance) similar to those of the time-shifted time series. We considered the top and bottom 10 000 nodes from the difference ranking as nodes that are approximately nonstationary (volatile) and stationary (stable), respectively. Table 3 compares the predictions of the models across time for nonstationary and stationary prediction tasks. Our findings indicate that the dynamic PageRank time series provides valuable information for forecasting future tweet rates; however, it adds little (if any) accuracy in forecasting future page views on Wikipedia.

For Twitter, the dynamic teleportation model improves predictions the most with the nonstationary nodes. The diffusion of activity captured by the model allows our model to detect, early on, when the external interest of vertices will change before that change becomes apparent in the external interest of the vertices. This is easiest to detect when there is a large sudden change in external interest of a neighboring vertex.



(a) Temporal Patterns



(b) Vertices with Similar Dynamics

Figure 7. Vertices with similar dynamic properties are grouped together. The visualization reveals the important dynamic patterns (spikes, trends) present from March 6, 2009, in our large collection of time series from Wikipedia. For each hour, we sample twice from the continuous function $\mathbf{x}(t)$ and use these intermediate values in the clustering.

6.2. Clustering Transient Score Trends

Identifying vertices with similar time series is important for modeling and understanding large collections of multivariate time series. We now group vertices according to their transient scores. Using the difference rank measure $\mathbf{d}$ for $s = 4$, we cluster the top 5000 vertices using k-means with $k = 5$, repeat the clustering 2000 times, and take the minimum distance clustering identified.

The cluster centroids are temporal patterns, and the main patterns in the dynamic PageRanks are visualized in Figure 7(a). Pattern 2 represents Eurocentric behavior, whereas the others correspond to spikes or unusual events occurring within the dynamic PageRank system. Figure 7(b) plots the 20 closest vertices matching the patterns above. A few pages from the five groups are consistent with our previously discussed results from Figure 6. One such unusual event is related to the death of a famous musician/actor from the Philippines (see pages 1–20).

The pages from the third cluster (41–60) are related to "American Idol" and other TV shows/actors. Also, some of the pages from the fourth cluster relate to Bernard Madoff ($63, 66, 67, 70, 73$) six days before he pleaded guilty in the largest financial fraud in U.S. history. This grouping reveals many of the standard patterns in time series such as spikes and increasing/decreasing trends [Yang and Leskovec 11].

6.3. Toward Causal Link Relationships

In this section, we use Granger causality tests [Granger 69] on the collection of transient scores to attempt to understand which links are most important. The Granger causality model, briefly described below, ought to identify a causal relationship between the time series of any two vertices connected by a directed edge. This is because there is a causal relationship between their time series in our dynamical system. However, due to the impact of the time-dependent teleportation, only some of these links will be identified as causal. We wish to investigate this smaller subset of links.

Intuitively, if a time series X causally affects another Y, then the past values of X should be helpful in predicting the future values of Y, above what can be predicted based on the past values of Y alone. This is formalized as follows: the error in predicting $\hat{y}_{t+s}$ from $y_t, y_{t-1}, \ldots$ should be larger than the error in predicting $\hat{y}_{t+s}$ from the joint data $y_t, y_{t_1}, \ldots, x_t, x_{t-1}, \ldots$ if X causes Y. As our model, we chose to use the standard vector-autoregressive (VAR) model from econometrics [Box et al. 11]. This is implemented in MATLAB in [LeSage 99]. The standard p-lag VAR model takes the form

$$\begin{bmatrix} y_t \\ x_t \end{bmatrix} = \mathbf{c} + \sum_{i=1}^{p} \mathbf{M}_i \begin{bmatrix} y_{t-i} \\ x_{t-i} \end{bmatrix} + \mathbf{e}_t,$$

Earthquake Granger Causes	p-value
Seismic hazard	0.003535
extensional tectonics	0.003033
landslide dam	0.002406
earthquake preparedness	0.001157
Richter magnitude scale	0.000584
fault (geology)	0.000437
aseismic creep	0.000419
seismometer	0.000284
epicenter	0.000020
seismology	0.000001

Table 4. Example of causality in Wikipedia. We consider only pages with a p-value less than 0.01 to be statistically significant. The page with values "caused" by "Earthquake" represent ideas related to earthquakes. All pages below are significant with p-value < 0.01.

where $\mathbf{c}$ is a vector of constants, $\boldsymbol{M}_i$ are the $n \times n$ coefficient (or autoregressive) mixing matrices, and $\mathbf{e}_t$ is the unobservable white noise. For the results shown below, $p = 2$. We then use the standard F-test to determine significance.

In Table 4, we show the causal relationships identified among the out-links of the article "Earthquake." Recall that there was a major earthquake in Australia during our time window. We wish to understand which of the out-links appeared to be sensitive to this large change in interest in "Earthquake." We used a significance cutoff of 0.01 and tested for Granger causality among the time series with $s = 4$.

7. Conclusion

PageRank is one of the most widely used network centrality measures. Our dynamical system reformulation of PageRank permits us to incorporate time-dependent teleportation in a relatively seamless manner. Based on the results presented here, we believe this to be an interesting variation on the PageRank model. For instance, we can analyze certain choices of oscillating teleportation functions (Lemma 2.6). Our empirical results show that the maximum change in the transient rank values identifies interesting sets of pages. Furthermore, this method is simple to implement in an online setting using either a forward Euler or Runge–Kutta integrator for the dynamical system. We hope that it might find a use in online monitoring systems.

One important direction for future work is to treat the inverse problem. That is, suppose that the observed page views reflect the behavior of these random surfers. Formally, suppose that we equate page views with samples of $\mathbf{x}(t)$. Then the goal would be to find $\mathbf{v}(t)$ that produces this $\mathbf{x}(t)$. This may not be a problem for websites such as Wikipedia, due to our argument that the majority of page views reflect search-engine traffic. But for many other cases, we suspect that $\mathbf{x}(t)$ may be much easier to observe.

References

[Abiteboul et al. 03] S. Abiteboul, M. Preda, and G. Cobena. "Adaptive On-Line Page Importance Computation." In *WWW*, pp. 280–290. ACM, 2003.

[Ahmed et al. 10] N. K. Ahmed, A. F. Atiya, N. El Gayar, and H. El-Shishiny. "An Empirical Comparison of Machine Learning Models for Time Series Forecasting." *Econ. Rev.* 29 (2010), 594–621.

[Andersen et al. 06] Reid Andersen, Fan Chung, and Kevin Lang. "Local Graph Partitioning Using PageRank Vectors." In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006.

[Bagrow et al. 11] J. P. Bagrow, D. Wang, and A. L. Barabási. "Collective Response of Human Populations to Large-Scale Emergencies." *PloS One* 6 (2011), e17680.

[Bahmani et al. 12] B. Bahmani, R. Kumar, M. Mahdian, and E. Upfal. "PageRank on an Evolving Graph." In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 24–32. ACM, 2012.

[Becchetti et al. 08] Luca Becchetti, Carlos Castillo, Debora Donato, Ricardo Baeza-Yates, and Stefano Leonardi. "Link Analysis for Web Spam Detection." *ACM Trans. Web* 2 (2008), 1–42.

[Berman et al. 89] Abraham Berman, Michael Neumann, and Ronald J. Stern. *Nonnegative Matrices in Dynamic Systems*. Wiley, 1989.

[Bianchini et al. 05] M. Bianchini, M. Gori, and F. Scarselli. "Inside PageRank." *ACM Transactions on Internet Technologies* 5 (2005), 92–128.

[Boldi 05] Paolo Boldi. "TotalRank: Ranking without Damping." In *WWW*, pp. 898–899, 2005.

[Boldi et al. 05] Paolo Boldi, Massimo Santini, and Sebastiano Vigna. "Paradoxical Effects in PageRank Incremental Computations." *Internet Mathematics* 2 (2005) 387–404.

[Boldi et al. 07] Paolo Boldi, Roberto Posenato, Massimo Santini, and Sebastiano Vigna. "Traps and Pitfalls of Topic-Biased PageRank." In *WAW2006, Fourth International Workshop on Algorithms and Models for the Web-Graph*, LNCS, pp. 107–116. Springer-Verlag, 2007.

[Box et al. 11] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time Series Analysis: Forecasting and Control*. Wiley, 2011.

[Chien et al. 04] S. Chien, C. Dwork, R. Kumar, D. R. Simon, and D. Sivakumar. "Link Evolution: Analysis and Algorithms." *Internet Mathematics* 1 (2004), 277–304.

[Constantine and Gleich 10] Paul G. Constantine and David F. Gleich. "Random Alpha PageRank." *Internet Mathematics* 6 (2010), 189–236.

[Das Sarma et al. 08] A. Das Sarma, S. Gollapudi, and R. Panigrahy. "Estimating PageRank on Graph Streams." In *SIGMOD*, pp. 69–78. ACM, 2008.

[Dunlavy et al. 11] Daniel M. Dunlavy, Tamara G. Kolda, and Evrim Acar. "Temporal Link Prediction Using Matrix and Tensor Factorizations." *TKDD* 5 (2011), 10:1–10:27.

[Embree and Lehoucq 09] Mark Embree and Richard B. Lehoucq. "Dynamical Systems and Non-Hermitian Iterative Eigensolvers." *SIAM Journal on Numerical Analysis* 47 (2009), 1445–1473.

[Gleich et al. 07] D. Gleich, P. Glynn, G. Golub, and C. Greif. "Three Results on the PageRank Vector: Eigenstructure, Sensitivity, and the Derivative." In *Web Information Retrieval and Linear Algebra Algorithms*. IBFI, 2007.

[Gleich et al. 10] D. F. Gleich, P. G. Constantine, A. D. Flaxman, and A. Gunawardana. "Tracking the Random Surfer: Empirically Measured Teleportation Parameters in PageRank." In *WWW*, pp. 381–390. ACM, 2010.

[Granger 69] C. W. J. Granger. "Investigating Causal Relations by Econometric Models and Cross-Spectral Methods." *Econometrica: Journal of the Econometric Society* 37 (1969), 424–438.

[Grindrod et al. 11] P. Grindrod, M. C. Parsons, D. J. Higham, and E. Estrada. "Communicability across Evolving Networks." *Physical Review E* 83 (2011), 046120.

[Gyöngyi et al. 04] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. "Combating Web Spam with TrustRank." In *Proceedings of the 30th International Very Large Database Conference*, Toronto, Canada, 2004.

[Horn and Serra-Capizzano 07] Roger A. Horn and Stefano Serra-Capizzano. "A General Setting for the Parametric Google Matrix." *Internet Mathematics* 3 (2007), 385–411.

[Kutta 01] W. Kutta. "Beitrag zur näherungsweisen Integration totaler Differentialgleichungen." *Z. Math. Phys.* 46 (1901), 435–453.

[Langville and Meyer 04] Amy N. Langville and Carl D. Meyer. "Updating PageRank with Iterative Aggregation." In *WWW*, pp. 392–393, 2004.

[Langville and Meyer 06] Amy N. Langville and Carl D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2006.

[LeSage 99] J. P. LeSage. "Applied Econometrics Using MATLAB." Manuscript, Dept. of Economics, University of Toronto, 1999.

[O'Madadhain and Smyth 05] J. O'Madadhain and P. Smyth. "EventRank: A Framework for Ranking Time-Varying Networks." In *LinkKDD*, pp. 9–16. ACM, 2005.

[Page et al. 99] L. Page, S. Brin, R. Motwani, and T. Winograd. "The PageRank Citation Ranking: Bringing Order to the Web." Technical Report 1999-66, Stanford University, 1999.

[Ratkiewicz et al. 10] J. Ratkiewicz, S. Fortunato, A. Flammini, F. Menczer, and A. Vespignani. "Characterizing and Modeling the Dynamics of Online Popularity." *PRL* 105 (2010), 158701.

[Rossi and Gleich 12] Ryan A. Rossi and David F. Gleich. "Dynamic PageRank Using Evolving Teleportation." In *Algorithms and Models for the Web Graph*, edited by Anthony Bonato and Jeannette Janssen, Lecture Notes in Computer Science 7323, pp. 126–137. Springer, 2012.

[Runge 95] C. Runge. "Über die numerische Auflösung von Differentialgleichungen." *Mathematische Annalen* 46 (1895), 167–178.

[Shampine and Reichelt 97] Lawrence F. Shampine and Mark W. Reichelt. "The MATLAB ODE Suite." *SIAM Journal on Scientific Computing* 18 (1997), 1–22.

[Singh et al. 07] Rohit Singh, Jinbo Xu, and Bonnie Berger. "Pairwise Global Alignment of Protein Interaction Networks by Matching Neighborhood Topology." In *Proceedings of the 11th Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, Lecture Notes in Computer Science 4453, pp. 16–31. Springer, 2007.

[Sun et al. 06] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. "Beyond Streams and Graphs: Dynamic Tensor Analysis." In *SIGKDD*, KDD '06, pp. 374–383. ACM, 2006.

[Tong et al. 06] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. "Fast Random Walk with Restart and Its Applications." In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pp. 613–622. IEEE Computer Society, 2006.

[Tsaparas 04] Panayiotis Tsaparas. "Using Non-linear Dynamical Systems for Web Searching and Ranking." In *Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 59–70. ACM, 2004.

[Wills and Ipsen 09] Rebecca S. Wills and Ilse C. F. Ipsen. "Ordinal Ranking for Google's PageRank." *SIAM Journal on Matrix Analysis and Applications* 30 (2009), 1677–1696.

[Yang and Leskovec 11] Jaewon Yang and Jure Leskovec. "Patterns of Temporal Variation in Online Media." In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pp. 177–186. ACM, 2011.

David F. Gleich, Purdue University, Department of Computer Science, 305 N. University St., West Lafayette, IN 47906, USA (dgleich@purdue.edu)

Ryan A. Rossi, Purdue University, Department of Computer Science, 305 N. University St., West Lafayette, IN 47906, USA (rrossi@purdue.edu)