Perron–based algorithms for the multilinear PageRank

Beatrice Meini∗ and Federico Poloni†

Abstract

We consider the multilinear PageRank problem studied in [Gleich, Lim and Yu, Multilinear PageRank, 2015], which is a system of quadratic equations with stochasticity and nonnegativity constraints. We use the theory of quadratic vector equations to prove several properties of its solutions and suggest new numerical algorithms. In particular, we prove the existence of a certain minimal solution, which does not always coincide with the stochastic one that is required by the problem. We use an interpretation of the solution as a Perron eigenvector to devise new fixed-point algorithms for its computation, and pair them with a continuation strategy based on a perturbative approach. The resulting numerical method is more reliable than the existing alternatives, being able to solve a larger number of problems.

Keywords: Multilinear PageRank; Perron vector; fixed point iteration; Newton’s method

1 Introduction

Gleich, Lim and Yu [1] consider the following problem, arising as an approximate computation of the stationary measure of an order-2 Markov chain: given \( \mathbf{v} \in \mathbb{R}^n, R \in \mathbb{R}^{n \times n^2}, \alpha \in \mathbb{R} \) with \( \mathbf{v} \geq 0, R \geq 0, \alpha \in (0, 1) \) and
\[
\mathbf{1}_n^\top \mathbf{v} = 1, \quad \mathbf{1}_n^\top R = \mathbf{1}_n^\top, \tag{1}
\]
solve for \( \mathbf{x} \) the equation
\[
\mathbf{x} = \alpha R (\mathbf{x} \otimes \mathbf{x}) + (1 - \alpha) \mathbf{v}. \tag{2}
\]
The solution of interest \( \mathbf{s} \) is stochastic, i.e., \( \mathbf{s} \geq 0 \) and \( \mathbf{1}_n^\top \mathbf{s} = 1 \). Here \( \mathbf{1} \) denotes a column vector of all ones (with an optional subscript to specify its length), and inequalities between vectors and matrices are intended in the componentwise sense.
In the paper [1], they prove some theoretical properties, consider several solution algorithms, and evaluate their performance. In the more recent paper [2], the authors improve some results concerning the uniqueness of the solution.

This problem originally appeared in [3], and is a variation of problems related to tensor eigenvalue problems and Perron–Frobenius theory for tensors; see, e.g., [3] [5] [6]. However, it also fits in the framework of quadratic vector equations derived from Markovian binary tree models introduced in [7] and later considered in [8] [9] [10]. Indeed, the paper [10] considers a more general problem, which is essentially [2] without the hypotheses [1]. Hence, all of its results apply here, and they can be
used in the context of multilinear PageRank. In particular, [10] considers the minimal nonnegative solution of (2) (in the componentwise sense), which is not necessarily stochastic as the one sought in [1].

In this paper, we use the theory of quadratic vector equations in [8, 9, 10] to better understand the behavior of the solutions of (2) and suggest new algorithms for computing the stochastic solution. More specifically, we show that if one considers the minimal nonnegative solution of (2) as well, the theoretical properties of (2) become clearer, even if one is only interested in stochastic solutions. Indeed we prove that there always exists a minimal nonnegative solution, which is the unique stochastic solution when $\alpha \leq \frac{1}{2}$. When $\alpha > \frac{1}{2}$, the minimal nonnegative solution $m$ is not stochastic and there is at least one stochastic solution $s \geq m$. Note that [1, Theorem 4.3] already proves that when $\alpha < \frac{1}{2}$ the stochastic solution is unique and [2, Theorem 1] slightly improves this bound; our results give a broader characterization. All this is in Section 2.

When $\alpha \leq \frac{1}{2}$, as already pointed out in [1], computing the stochastic solution of (2) is easy. Indeed, this is also due to the fact that the stochastic solution is the minimal solution, and for instance the numerical methods proposed in [7, 8, 9] perform very well. The most difficult case is when $\alpha > \frac{1}{2}$, in particular when $\alpha \approx 1$. Since the minimal solution $m$ of (2) can be easily computed, the idea is to compute and deflate it, with a similar strategy to the one developed in [8, 9], hence allowing us to compute stochastic solutions even when they are not minimal. The main tool in this approach is rearranging (2) to show that (after a change of variables) a solution $x$ corresponds to the Perron eigenvector of a certain matrix that depends on $x$ itself. This interpretation in terms of Perron vector allows to devise new algorithms based on fixed point iteration and on Newton's method. Sections 3 and 4 describe this deflation technique and the algorithms based on the Perron vector computation.

Finally, we propose in Section 5 a continuation strategy based on a perturbative approach that allows one to solve the problem for values $\hat{\alpha} < \alpha$ in order to obtain better starting values for the more challenging cases when $\alpha \approx 1$.

We report several numerical experiments in Section 6 to show the effectiveness of these new techniques for the set of small-scale benchmark problems introduced in [1], and draw some conclusions in Section 7.

## 2 Properties of the nonnegative solutions

In this section, we show properties of the nonnegative solutions of the equation (2). In particular, we prove that there always exists a minimal nonnegative solution, which is stochastic when $\alpha \leq \frac{1}{2}$. These properties can be derived by specializing the results of [10], which apply to more general vector equations defined by bilinear forms.

We introduce the map

$$G(x) := \alpha R(x \otimes x) + (1 - \alpha)v,$$

and its Jacobian

$$G'_x := \alpha R(x \otimes I_n) + \alpha R(I_n \otimes x).$$

We have the following result.

**Lemma 1.** Consider the fixed-point iteration

$$x_{k+1} = G(x_k), \quad k = 0, 1, \ldots,$$  \hspace{1cm} (3)
Consider the fixed-point iteration \( \{x_k\} \), with a certain initial value \( x_0 \geq 0 \), for the problem (2) with \( \alpha > \frac{1}{2} \). Define \( z_k := 1^\top x_k \). Then,

- If \( z_0 \in (0, \frac{1-\alpha}{\alpha}) \), then \( \lim_{k \to \infty} z_k = \frac{1-\alpha}{\alpha} \), and the iteration (3) can converge only to the minimal solution \( m \) (if it converges).

2.1 Sum of entries and criticality

In this specific problem, the hypotheses (1) enforce a stronger structure on the iterates of (3): the sum of the entries of \( G(x) \) is a function of the sum of the entries of \( x \) only.

**Lemma 2.** Let \( g(u) := \alpha u^2 + (1 - \alpha) \). Then, \( 1^\top G(x) = g(1^\top x) \) for any \( x \in \mathbb{R}^n \).

**Proof.**

\[
1^\top G(x) = 1^\top (\alpha R(x \otimes x) + (1 - \alpha) v) = \alpha 1^\top R(x \otimes x) + (1 - \alpha) 1^\top v = \alpha 1^\top (x \otimes x) + (1 - \alpha) = \alpha (1^\top x)^2 + (1 - \alpha).
\]

This fact has important consequences for the sum of the entries of the solutions of (2).

**Corollary 3.** For each solution \( x \) of (2), \( 1^\top x \) is one of the two solutions of the quadratic \( u = g(u) \), i.e., \( u = 1 \) or \( u = \frac{1-\alpha}{\alpha} \).

Let \( u \) be one of the solutions of \( u = g(u) \) and define the level set \( \ell_u = \{ x : 1^\top x = u, x \geq 0 \} \). Since \( \ell_u \) is convex and compact, and since \( G(x) \) maps \( \ell_u \) to itself by Lemma 2, then the Brouwer fixed-point theorem implies the following result.

**Corollary 4.** There exists at least a solution \( x \geq 0 \) to (2) with \( 1^\top x = 1 \) and a solution \( x \geq 0 \) with \( 1^\top x = \frac{1-\alpha}{\alpha} \).

Hence we can have two different settings, for which we borrow the terminology from (10).

**Subcritical case** \( \alpha \leq \frac{1}{2} \), hence the minimal nonnegative solution \( m = s \) is the unique stochastic solution.

**Supercritical case** \( \alpha > \frac{1}{2} \), hence the minimal nonnegative solution \( m \) satisfies \( 1^\top m = \frac{1-\alpha}{\alpha} < 1 \) and there is at least one stochastic solution \( s \geq m \).

Note that (1) Theorem 4.3 already proves that when \( \alpha \leq \frac{1}{2} \) the stochastic solution is unique; these results give a broader characterization.

The tools that we have introduced can already be used to determine the behavior of simple iterations such as (3).

**Theorem 5.** Consider the fixed-point iteration (3), with a certain initial value \( x_0 \geq 0 \), for the problem (2) with \( \alpha > \frac{1}{2} \). Define \( z_k := 1^\top x_k \). Then,

- If \( z_0 \in (0, \frac{1-\alpha}{\alpha}) \), then \( \lim_{k \to \infty} z_k = \frac{1-\alpha}{\alpha} \), and the iteration (3) can converge only to the minimal solution \( m \) (if it converges).
• If \( z_0 \in (\frac{1-\alpha}{\alpha}, 1) \), then \( \lim_{k \to \infty} z_k = 1 \), hence the iteration (3) can converge only to a stochastic solution (if it converges).

• If \( z_0 \in (1, +\infty) \), then \( \lim_{k \to \infty} z_k = +\infty \), hence the iteration (3) diverges.

Proof. Thanks to Lemma (2) the quantity \( z_k := 1^\top x_k \) evolves according to \( z_{k+1} = g(z_k) \). So the result follows by the theory of scalar fixed-point iterations, since this iteration converges to \( \frac{1-\alpha}{\alpha} \) for \( z_0 \in (0, \frac{1-\alpha}{\alpha}) \), to 1 for \( z_0 \in (\frac{1-\alpha}{\alpha}, 1) \), and diverges for \( z_0 \in (1, +\infty) \).

An analogous result holds for the subcritical case.

The papers [8, 9, 10] describe several methods to compute the minimal solution \( m \). In particular, all the ones described in [10] exhibit monotonic convergence, that is, \( 0 = x_0 \leq x_1 \leq x_2 \leq \cdots \leq x_k \leq \cdots \leq m \). Due to the uniqueness and the monotonic convergence properties, computing the minimal solution \( m \) is typically simple, fast, and free of numerical issues. Hence in the subcritical case the multilinear PageRank problem is easy to solve. The supercritical case is more problematic.

Among all available algorithms to compute the minimal solution \( m \), we recall Newton’s method, which is one of the most efficient ones. The Newton–Raphson method applied to the function \( F(x) = x - G(x) \) generates the sequence

\[
(I - G'_x) x_{k+1} = (1 - \alpha) v - \alpha R(x_k \otimes x_k), \quad k = 0, 1, \ldots.
\]

(4)

The following result holds (10, Theorem 13).

Lemma 6. Suppose that \( m > 0 \), and that \( G'_m \) is irreducible. Then, Newton’s method (4) starting from \( x_0 = 0 \) is well defined and converges monotonically to \( m \) (i.e., \( 0 = x_0 \leq x_1 \leq x_2 \leq \cdots \leq x_k \leq \cdots \leq m \)).

Algorithm (1) shows a straightforward implementation of Newton’s method as described above.

Input: \( R, \alpha, v \) as above, a tolerance \( \varepsilon \).

Output: An approximation to the minimal solution \( m \) of (2).

\[
x \leftarrow 0;
\]

while \( ||G(x) - x||_1 > \varepsilon \) do
\[
G'_x \leftarrow \alpha R(x \otimes I_n) + \alpha R(I_n \otimes x);
\]
\[
x \leftarrow (I_n - G'_x)^{-1}((1 - \alpha)v - \alpha R(x \otimes x));
\]
end

\( m = x; \)

Algorithm 1: Newton’s method for the computation of the minimal solution \( m \) to (2).

Note that the theory in (10, Section 9) shows how one can predict where zeros appear in \( m \) and eliminate them reducing the problem to a smaller one. Indeed, in view of the probabilistic interpretation of multilinear PageRank, zero entries in \( m \) can appear only when the second-order Markov chain associated to \( R \) is not irreducible. So we can assume in the following that \( m > 0 \), and that the nonnegative matrix \( G'_m \) is irreducible. In particular, in this case \( G'_m \) is also substochastic (as proved in [10, Theorem 6]).
3 Deflating the minimal solution

From now on, we assume to be in the supercritical case, i.e., $\alpha > 1/2$, and that $m$ has been already computed and is explicitly available.

We wish adapt to this setting the deflation strategy introduced in [9]. Since all solutions $s$ to (2) satisfy $s \geq m$, it makes sense to change variable to obtain an equation in $y := s - m \geq 0$. After a few manipulations, using bilinearity of $(m + y) \otimes (m + y)$ and the fact that $m = \alpha R(m \otimes m) + (1 - \alpha)v$, one gets

$$y = \alpha R(y \otimes y) + \alpha R(m \otimes y) = \alpha R(y \otimes y) + G_m' y = (\alpha R(y \otimes I_n) + G_m') y.$$  

Moreover,

$$1^T y = 1^T (s - m) = 1 - \frac{1 - \alpha}{\alpha} = \frac{2\alpha - 1}{\alpha},$$  

(6)

We set $P_y := \alpha R(y \otimes I_n) + G_m'$. Note that $P_y \geq G_m'$ for each $y \geq 0$, hence it is irreducible. In addition, if $y$ is chosen such that $1^T y = \frac{2\alpha - 1}{\alpha}$ as in (6), then

$$1^T P_y = \alpha 1^T R(y \otimes I) + \alpha 1^T R(I \otimes m) + \alpha 1^T R(m \otimes I)$$

$$= \alpha 1^T (y \otimes I) + \alpha 1^T (I \otimes m) + \alpha 1^T (m \otimes I)$$

$$= \frac{2\alpha - 1}{\alpha} 1^T + \frac{1 - \alpha}{\alpha} 1^T + \frac{1 - \alpha}{\alpha} 1^T = 1^T,$$

(7)

so $P_y$ is a stochastic matrix.

Let us introduce the map $P(A)$ that gives the Perron vector $w$ of an irreducible matrix $A \geq 0$, normalized such that $1^T w = 1$. Then, since $P_y$ is irreducible and stochastic, (5) and (6) show that

$$y = \frac{2\alpha - 1}{\alpha} P(\alpha R(y \otimes y) + \alpha R(m \otimes y) = \alpha R(y \otimes y) + G_m' y,$$

(8)

i.e., the sought vector $y$ is the Perron vector of the matrix $P_y$, multiplied by the constant $\frac{2\alpha - 1}{\alpha}$.

4 Perron-based algorithms

Equation (8) suggests a new fixed-point iteration for computing $y$, which is analogous to the one appearing in [9],

$$y_{k+1} = \frac{2\alpha - 1}{\alpha} P(\alpha R(y \otimes y) + \alpha R(m \otimes y) = \alpha R(y \otimes y) + G_m' y,$$

(9)

starting from a given nonnegative vector $y_0$ such that $1^T y_0 = \frac{2\alpha - 1}{\alpha}$. This iteration is implemented in Algorithm 2.

We may also apply Newton’s method to find a solution to (8), following [8]. To this end, we first compute the Jacobian of the function $w(y) := P(A)$.

**Lemma 7.** Let $w(y) := P(A)$, with $y \geq 0$ such that $1^T y = \frac{2\alpha - 1}{\alpha}$. Then, its Jacobian is given by

$$w'(y) = \alpha \left( (I - P_y + w(y)1^T)^{-1} R(I \otimes w(y)) - w(y)1^T \right).$$

(10)
The matrix in the left-hand side is nonsingular, since it can be obtained by replacing $v$ is a simple eigenvalue, and hence we can define locally smooth functions $P$ Perron eigenvalue of the eigenvalue 0 with 1 in the eigendecomposition of the singular irreducible M-matrix $I - P$. Hence, we can write $G_m' = \alpha R(m \otimes I_n) + \alpha R(I_n \otimes m)$; $\|G(x) - x\|_1 > \varepsilon$ do

\begin{align*}
P_y &\leftarrow \alpha R(y \otimes I_n) + G_m'; \\
y &\leftarrow \frac{2\alpha - 1}{\alpha} v; \\
x &\leftarrow m + y;
\end{align*}

end

$s = x$.

Algorithm 2: The Perron-based iteration for the computation of a stochastic solution $s$ to (2).

Proof. Let us compute its directional derivative along the direction $z$. We set $y(h) = y + hz$, hence $\frac{d}{dh} P_y(h) = \alpha R(z \otimes I)$. Since $P_y$ is irreducible, its Perron eigenvalue is a simple eigenvalue, and hence we can define locally smooth functions $\lambda(h)$ as the Perron eigenvalue of $P_y(h)$ and $v(h) = \mathcal{P}(P_y(h))$. A computation analogous to (7) shows that $1^T P_y(h) = (1 + h \alpha 1^T) 1^T$, hence $\lambda(h) = 1 + h \alpha 1^T$ and $\frac{d}{dh} \lambda(h) = \alpha (1^T z)$.

By differentiating $P_y(h) v(h) = \lambda(h) v(h)$ and evaluating at $h = 0$, we get

$$\alpha R(z \otimes I)v(0) + P_y \frac{dv}{dh}(0) = \alpha (1^T z)v(0) + \frac{dv}{dh}(0),$$

or, rearranging terms,

$$(I - P_y) \frac{dv}{dh}(0) = \alpha \left( R(I \otimes v(0)) - v(0) 1^T \right) z,$$

where $v(0) = w(y)$. Since we defined $v(h)$ so that $1^T v(h) = 1$, we have $1^T \frac{dv}{dh}(0) = 0$, and hence also

$$(I - P_y + v(0) 1^T) \frac{dv}{dh}(0) = \left( R(I \otimes v(0)) - v(0) 1^T \right) z.$$

The matrix in the left-hand side is nonsingular, since it can be obtained by replacing the eigenvalue 0 with 1 in the eigendecomposition of the singular irreducible M-matrix $I - P_y$. Thus we can write

$$\frac{dv}{dh}(0) = \alpha (I - P_y + v(0) 1^T)^{-1} \left( R(I \otimes v(0)) - v(0) 1^T \right) z.$$

Since $(I - P_y + v(0) 1^T)v(0) = v(0)$, we can simplify this expression further to

$$\frac{dv}{dh}(0) = \alpha \left( (I - P_y + v(0) 1^T)^{-1} R(I \otimes v(0)) - v(0) 1^T \right) z,$$

from which we get (10).
From the above result, the Jacobian of the function $H(y) = y - \frac{2\alpha - 1}{\alpha} P V(P y)$ is given by

$$H'_y = I_n + (2\alpha - 1)w(y)1^T - (2\alpha - 1)(I - P_y + w(y)1^T)^{-1} R(I \otimes w(y))$$  \hspace{1cm} (11)$$

and Newton’s method consists of generating the sequence of vectors

$$y_{k+1} = y_k - (H'_y y_k)^{-1} H(y_k)$$

The Perron–Newton method is described in Algorithm 3.

**Input:** $R, \alpha, v$ as above, with $\alpha > \frac{1}{2}$, a tolerance $\varepsilon$, an initial value $x_0$.

**Output:** An approximation to a stochastic solution $s$ of (2).

Compute $m$ with Algorithm 1;

Normalize $x_0$ (if needed): $x_0 \leftarrow \max(x_0, 0)$, $x_0 \leftarrow \frac{x_0}{\|x_0\|_1}$;

$y \leftarrow x_0 - m$;

Normalize $y$ (if needed): $y \leftarrow \max(y, 0)$, $y \leftarrow \frac{2\alpha - 1}{\alpha} y^\top 1$;

$G'_m \leftarrow \alpha R(m \otimes I_n) + \alpha R(I_n \otimes m)$;

while $\|G(x) - x\|_1 > \varepsilon$ do

$P_y \leftarrow \alpha R(y \otimes I_n) + G'_m$;

$w \leftarrow PV(P_y)$;

$H'_y \leftarrow I_n + (2\alpha - 1)w1^T - (2\alpha - 1)(I - P_y + w1^T)^{-1} R(I \otimes w)$;

$y \leftarrow y - H'_y^{-1}(y - \frac{2\alpha - 1}{\alpha} w)$;

Normalize $y$ (if needed): $y \leftarrow \frac{2\alpha - 1}{\alpha} y^\top 1$;

$x \leftarrow m + y$;

end

$s = x$;

**Algorithm 3:** The Perron–Newton method for the computation of a stochastic solution $s$ to (2).

The standard theorems [11] on local convergence of Newton’s method imply the following result.

**Theorem 8.** Suppose that the matrix $H'_s-m$ is nonsingular. Then the Perron–Newton method is locally convergent to a stochastic solution $s$ of (2).

**Remark 9.** Since $1^T (w(y)1^T - (I - P_y + w(y)1^T)^{-1} R(I \otimes w(y))) = 0$, the matrix $H'_y$ has an eigenvalue 1 with left eigenvector $1^T$, for each value of $y$.

5 Continuation techniques

The above algorithms, as well as those in [1], are sufficient to solve most of the test problems that are explored in [1]. However, especially when $\alpha \approx 1$, the algorithms may converge very slowly or stagnate far away from the minimal solution. For this reason, we explore an additional technique based on a perturbation expansion of the solution as a function of $\alpha$, inspired loosely by the ideas of homotopy continuation [12], which is a well-known strategy to derive approximate solutions for parameter-dependent equations.

The main result is the following.
Lemma 10. Let $s$ be a stochastic solution of problem (2) (for a certain $\alpha > \frac{1}{2}$), and suppose that $I - G'_{\alpha}$ is nonsingular. Then, there is a stochastic solution $s_{\hat{\alpha}}$ to (2) with the parameter $\alpha$ replaced by $\hat{\alpha}$ such that

$$s_{\hat{\alpha}} = s + s^{(1)}(\hat{\alpha} - \alpha) + s^{(2)}(\hat{\alpha} - \alpha)^2 + O((\hat{\alpha} - \alpha)^3),$$

with $s^{(1)} = (I - G'_{\alpha,s_{\hat{\alpha}}})^{-1}(R(s_{\hat{\alpha}} \otimes s_{\hat{\alpha}}) - v)$, $s^{(2)} = (I - G''_{\alpha,s_{\hat{\alpha}}})^{-1}R\left(t_{\alpha} \otimes s^{(1)} + s^{(1)} \otimes t_{\alpha}\right)$, where $t_{\alpha} := 2s_{\alpha} + \alpha s^{(1)}$.

Proof. Let us make the dependence of the various quantities on the parameter $\alpha$ explicit, i.e., we write $s_{\alpha}$ to denote a stochastic solution of (2) for a certain value of the parameter $\alpha$, and similarly $G_{\alpha}, G'_{\alpha,s}$, and $F_{\alpha}$.

We apply the implicit function theorem [13, Theorem 9.28] to

$$0 = F_{\alpha}(s_{\alpha}) = s_{\alpha} - \alpha R(s_{\alpha} \otimes s_{\alpha}) - (1 - \alpha)v,$$

obtaining

$$\frac{\partial F_{\alpha}}{\partial s_{\alpha}} = I - \alpha R(s_{\alpha} \otimes I) - \alpha R(I \otimes s_{\alpha}) = I - G'_{\alpha,s_{\alpha}},$$

$$\frac{\partial F_{\alpha}}{\partial \alpha} = v - R(s_{\alpha} \otimes s_{\alpha}),$$

$$s^{(1)} = \frac{d}{d\alpha} s_{\alpha} = -\left(\frac{\partial F_{\alpha}}{\partial s_{\alpha}}\right)^{-1}\frac{\partial F_{\alpha}}{\partial \alpha} = -(I - G'_{\alpha,s_{\alpha}})^{-1}(v - R(s_{\alpha} \otimes s_{\alpha})).$$

Differentiating (13a), we get

$$\frac{d}{d\alpha} s_{\alpha} = -R(s_{\alpha} \otimes I) - R(I \otimes s_{\alpha}) = \alpha R\left(s^{(1)} \otimes I\right) - \alpha R\left(I \otimes s^{(1)}\right),$$

$$\frac{d}{d\alpha} F_{\alpha} = -R\left(s^{(1)} \otimes s_{\alpha}\right) - R\left(s_{\alpha} \otimes s^{(1)}\right),$$

$$s^{(2)} = \frac{d^2 s_{\alpha}}{d\alpha^2} = \left(\frac{\partial F_{\alpha}}{\partial s_{\alpha}}\right)^{-1}\left(\frac{d}{d\alpha} \frac{\partial F_{\alpha}}{\partial s_{\alpha}}\right)\frac{\partial F_{\alpha}}{\partial \alpha} - \left(\frac{\partial F_{\alpha}}{\partial s_{\alpha}}\right)^{-1}\frac{d}{d\alpha} \frac{\partial F_{\alpha}}{\partial \alpha},$$

$$= -(I - G'_{\alpha,s_{\alpha}})^{-1}\left(R\left(t_{\alpha} \otimes s^{(1)}\right) + R\left(s^{(1)} \otimes t_{\alpha}\right)\right).$$

The function $s_{\hat{\alpha}}$ obtained by the theorem must satisfy $1^Ts_{\hat{\alpha}} = 1$ for each $\hat{\alpha} > \frac{1}{2}$. Indeed, by Corollary [3] for a solution $s_{\hat{\alpha}}$ of the equation $x = G_{\alpha}(x)$ it must hold either $1^Ts_{\hat{\alpha}} = 1$ or $1^Ts_{\hat{\alpha}} = \frac{1}{\alpha_{\hat{\alpha}}} < 1$, and the continuous function $1^Ts_{\hat{\alpha}}$ cannot jump from 1 to $\frac{1}{\alpha_{\hat{\alpha}}}$ without taking any intermediate value.

The formula (12) now follows by Taylor expansion. □

This result suggests a strategy to improve convergence for $\alpha \approx 1$: we start solving the problem with a small value of $\alpha = \alpha_0$, e.g., 0.6, then we solve it repeatedly for increasing values $\alpha_0 < \alpha_1 < \alpha_2 < \cdots < \alpha_k = \alpha$; at each step $h$ we use as a starting point for our iterative algorithms an approximation of $s_{\alpha_{h+1}}$ constructed from $s_{\alpha_h}$.

We tried several different strategies to construct this approximation.

T2 Second-order Taylor expansion $s_{\alpha_{h+1}} \approx s_{\alpha_h} + \frac{\partial s_{\alpha}}{\partial \alpha}\big|_{\alpha_h} (\alpha_{h+1} - \alpha_h) + \frac{1}{2} \frac{d^2 s_{\alpha}}{d\alpha^2}\big|_{\alpha_h} (\alpha_{h+1} - \alpha_h)^2$.
T1 First-order Taylor expansion $s_{\alpha_{h+1}} \approx s_{\alpha_h} + \frac{ds}{d\alpha} \alpha_h (\alpha_{h+1} - \alpha_h)$;

EXT Linear extrapolation from the previous two values $s_{\alpha_{h+1}} \approx s_{\alpha_h} + \frac{s_{\alpha_h} - s_{\alpha_{h-1}}}{\alpha_{h-1} - \alpha_h} (\alpha_{h+1} - \alpha_h)$;

IMP An attempt at building a ‘semi-implicit method’ with only the values available: ideally, one would like to set $s_{\alpha_{h+1}} \approx s_{\alpha_h} + \frac{ds}{d\alpha} \alpha_h (\alpha_{h+1} - \alpha_h)$ (which would correspond to expressing $s_{\alpha_h}$ with a Taylor expansion around $\alpha_{h+1}$); however, we do not know $\frac{ds}{d\alpha} \alpha_h$. Instead, we take $s_{\alpha_{h+1}} \approx s_{\alpha_h} + s^{(1)} (\alpha_{h+1} - \alpha_h)$ with

$$s^{(1)} = (I - \alpha_{h+1} R(s_{\alpha_h} \otimes I) - \alpha_{h+1} R(I \otimes s_{\alpha_h}))^{-1} R(s_{\alpha_h} \otimes s_{\alpha_h} - v),$$

i.e., we use the new value of $\alpha$ and the old value of $s$ in the expression for the first derivative.

The only remaining choice is designing an effective strategy to choose the next point $\alpha_{h+1}$. We adopt the following one. Note that for T1,EXT,IMP we expect $s_{\alpha_{h+1}} - s_{\alpha_h} = O((\alpha_{h+1} - \alpha_h)^2)$. Hence one can expect

$$\frac{s_{\alpha_{h+1}} - s_{\alpha_{h-1}}}{(\alpha_{h+1} - \alpha_{h-1})^2} \approx \frac{s_{\alpha_{h+1}} - s_{\alpha_h}}{(\alpha_{h+1} - \alpha_h)^2}.$$

Thus we choose a ‘target’ value $\tau$ (e.g., $\tau = 0.01$) for $s_{\alpha_{h+1}} - s_{\alpha_h}$, and then we choose $\alpha_{h+1}$ by solving the equation

$$\frac{s_{\alpha_{h+1}} - s_{\alpha_{h-1}}}{(\alpha_{h+1} - \alpha_{h-1})^2} = \frac{\tau}{(\alpha_{h+1} - \alpha_h)^2}.$$

The resulting continuation algorithm is described in Algorithm 4. Note that we start from $\alpha_0 = 0.6$, to steer clear of the double solution for $\alpha = 0.5$. Convergence for such a value of $\alpha_0$ is typically not problematic.

Note that for strategy T2 one would expect $s_{\alpha_{h+1}} - s_{\alpha_h} = O((\alpha_{h+1} - \alpha_h)^3)$ instead, but for simplicity (and to ensure a fair comparison between the methods) we use the same step size selection strategy.

6 Numerical experiments

We have implemented the described methods using Matlab, and compared them to some of the algorithms in [14] on the same benchmark set of small-size models ($n \in \{3, 4, 6\}$) used in [1]. We have used tolerance $\varepsilon = \sqrt{u}$, where $u$ is the machine precision, $x_0 = v$ as an initial value unless differently specified, and $\tau = 0.01$ for Algorithm 1. To compute Perron vectors, we have used the output of Matlab’s eig. For problems with larger $n$, different methods (eigs, inverse iteration, Gaussian elimination for kernel computation, etc.) can be considered [15].

The results, for various values of $\alpha$, are reported in Tables 1 to 3. Each of the 29 rows represents a different experiment in the benchmark set, and the columns stand for the different algorithms, according to the following legend.

F The fixed-point iteration [3], from an initial value such that $1^T x_0 = 1$ (and with renormalization to $1^T x_k = 1$ at each step).

IO The inner-outer iteration method, as described in [1].

N Newton’s method with normalization to $1^T x_k = 1$ at each step, as described in [4].

Note that this is not Algorithm 4 which would converge instead to the minimal solution $m$. 
Input: \( R, \alpha, v \) as above, with \( \alpha > 0.6 \), a tolerance \( \varepsilon \), a continuation ‘speed’ parameter \( \tau \).

Output: An approximation to a stochastic solution \( s \) of (2).

\[
\alpha_0 \leftarrow 0.6; \\
h = 0; \\
\text{while } \alpha_h < \alpha \text{ do} \\
\quad \text{if } h = 0 \text{ then} \\
\quad \quad x_{\text{guess}} \leftarrow v; \\
\quad \quad \text{else} \\
\quad \quad \quad x_{\text{guess}} \text{ determined using } T1, T2, \text{IMP or (if } h > 1 \text{) EXT;} \\
\quad \quad \text{end} \\
\quad x_h \leftarrow \text{a stochastic solution to (2) with parameters } R, \alpha_h, v \\
\quad \quad \text{(determined using a the Newton or Perron-Newton method, with initial value } x_{\text{guess}}); \\
\quad \text{if } h = 0 \text{ then} \\
\quad \quad \quad \alpha_{h+1} \leftarrow \alpha_h + 0.01; \\
\quad \quad \quad \text{else} \\
\quad \quad \quad \quad \alpha_{h+1} = \alpha_h + \sqrt{\frac{2(\alpha_h - \alpha_{h-1})^2}{x_h - x_{h-1}}}; \\
\quad \quad \text{end} \\
\quad \text{if } \alpha_{h+1} > \alpha \text{ then} \\
\quad \quad \quad \alpha_{h+1} \leftarrow \alpha; \\
\quad \quad \text{end} \\
\quad h \leftarrow h + 1; \\
\text{end} \\
s \leftarrow x_{h-1};
\]

Algorithm 4: The continuation method for the computation of a stochastic solution \( s \) to (2).
Fig. 1: Performance profiles for the 29 benchmark tensors and $\alpha = 0.9$

$P$ The Perron method (Algorithm 2).

$PN$ The Perron–Newton method (Algorithm 3).

$N$-xxx, $PN$-xxx $N$ or $PN$ methods combined with continuation with one of the four strategies described in Section 5.

For each experiment, we report the number of iterations required, and the CPU times in seconds (obtained on Matlab R2017a on a computer equipped with a 64-bit Intel core i5-6200U processor). The value NaN inside a table represents failure to converge after 10,000 iterations. For $P$ and $PN$, the number of iterations is defined as the sum of the number of Newton iterations required to compute $\mathbf{m}$ with Algorithm 1 and the number of iterations in the main algorithm. For the continuation algorithms, we report the number of inner iterations, that is, the total number of iterations performed by $PN$ or $N$ (respectively), summing along all calls to the subroutine.

In addition to the tables, performance profiles [16, Section 22.4] are reported in Figures 1–3.

Note that neither the number of iterations nor the CPU time are particularly indicative of the true performance of the algorithms: indeed, the iterations in the different methods amount to different quantity of work, since some require merely linear system solutions and some require eigenvalue computations. Moreover, Matlab introduces overheads which are difficult to predict for several instructions (such as variable access and function calls). In order to make the time comparisons more fair, for methods $IO$ and $N$ we did not use the code in [14], but rather rewrote it without making use of Matlab’s object-oriented features, which made the original functions significantly slower. The only change introduced with respect to their implementations is that we changed the stopping criterion to $\|G(x) - x\|_1 \leq \sqrt{u}$, so that the stopping criterion is the same one for all tested methods.

In any case, the performance comparison between the various methods should be taken with a pinch of salt.

We comment briefly on the results. Newton-based methods typically require a
| F | IO | N | O | ON | ON-T1 | ON-IMP | ON-T2 | ON-EXT | N-T1 | N-IMP | N-T2 | N-EXT |
|---|---|---|---|----|------|-------|------|-------|-----|------|-----|------|
| 184 | 45 | 4 | 19 | 7 | 30 | 23 | 23 | 47 | 8 | 6 | 23 | 13 |
| 18 | 40 | 5 | 22 | 9 | 40 | 32 | 25 | 64 | 11 | 9 | 25 | 21 |
| 18 | 40 | 5 | 14 | 8 | 40 | 32 | 25 | 64 | 11 | 9 | 25 | 21 |
| 67 | 28 | 4 | 17 | 7 | 23 | 23 | 23 | 40 | 6 | 6 | 23 | 11 |
| 54 | 83 | 6 | 29 | 10 | 62 | 30 | 26 | 94 | 19 | 13 | 26 | 30 |
| 227 | 169 | 7 | 59 | 9 | 55 | 32 | 27 | 82 | 19 | 10 | 27 | 28 |
| 680 | 142 | 5 | 63 | 8 | 39 | 25 | 25 | 57 | 11 | 7 | 25 | 17 |
| 117 | 153 | 6 | 63 | 8 | 32 | 25 | 25 | 51 | 9 | 7 | 25 | 15 |
| 512 | 166 | 5 | 63 | 8 | 37 | 25 | 25 | 51 | 11 | 7 | 25 | 15 |
| 407 | 160 | 6 | 95 | 9 | 47 | 25 | 26 | 73 | 15 | 7 | 26 | 23 |
| 654 | 140 | 5 | 97 | 8 | 39 | 24 | 25 | 58 | 12 | 6 | 25 | 17 |
| 186 | 114 | 6 | 51 | 9 | 47 | 31 | 25 | 72 | 15 | 8 | 25 | 22 |
| 248 | 134 | 6 | 58 | 8 | 53 | 32 | 25 | 79 | 17 | 11 | 25 | 27 |
| 147 | 300 | 7 | 92 | 7 | 31 | 23 | 23 | 82 | 9 | 6 | 23 | 12 |
| 114 | 155 | 6 | 43 | 8 | 32 | 31 | 25 | 51 | 11 | 10 | 25 | 17 |
| 240 | 175 | 6 | 68 | 8 | 47 | 25 | 26 | 72 | 16 | 8 | 26 | 24 |
| 204 | 124 | 6 | 52 | 8 | 47 | 32 | 26 | 66 | 2 | 6 | 26 | 16 |
| 594 | 140 | 5 | 57 | 8 | 31 | 24 | 25 | 50 | 9 | 6 | 25 | 14 |
| 227 | 167 | 5 | 79 | 8 | 47 | 25 | 26 | 66 | 14 | 7 | 26 | 21 |
| 1,514 | 207 | 8 | 103 | 7 | 31 | 25 | 24 | 50 | 9 | 8 | 24 | 16 |
| 136 | 127 | 8 | 55 | 10 | 40 | 27 | 28 | 59 | 14 | 10 | 28 | 20 |
| 161 | 109 | 5 | 51 | 7 | 40 | 24 | 25 | 58 | 12 | 7 | 25 | 17 |
| 170 | 225 | 7 | 46 | 8 | 48 | 33 | 25 | 65 | 15 | 13 | 25 | 23 |
| 91 | 110 | 6 | 40 | 9 | 48 | 26 | 26 | 71 | 14 | 8 | 26 | 22 |
| 69 | 99 | 5 | 38 | 7 | 31 | 24 | 24 | 43 | 8 | 6 | 24 | 12 |
| 112 | 131 | 6 | 58 | 8 | 38 | 31 | 25 | 65 | 13 | 10 | 25 | 22 |
| 70 | 106 | 6 | 38 | 7 | 30 | 24 | 24 | 48 | 10 | 8 | 24 | 16 |

Tab. 1: Iteration counts and times for the 29 benchmark tensors and $\alpha = 0.90$
Numerical experiments

| F | IO | N | O | ON | ON-T1 | ON-IMP | ON-T2 | ON-EXT | N-T1 | N-IMP | N-T2 | N-EXT |
|---|----|---|---|----|-------|--------|-------|--------|------|-------|------|--------|
| NaN | 42 | 5 | 21 | 6 | 35 | 22 | 22 | 46 | 10 | 6 | 22 | 13 |
| 23 | 48 | 6 | 26 | 7 | 50 | 36 | 24 | 80 | 17 | 13 | 24 | 29 |
| 23 | 48 | 6 | 10 | 7 | 50 | 36 | 24 | 80 | 17 | 13 | 24 | 29 |
| 114 | 24 | 4 | 19 | 6 | 28 | 22 | 22 | 39 | 8 | 6 | 22 | 11 |
| 22 | 41 | 17 | 15 | 7 | 74 | 26 | 24 | 110 | 23 | 29 | 24 | 37 |
| NaN | 639 | 7 | NaN | NaN | 73 | 39 | 25 | 109 | 26 | 14 | 25 | 38 |
| NaN | 234 | 5 | 369 | 8 | 44 | 24 | 24 | 63 | 13 | 7 | 24 | 29 |
| NaN | 1,221 | 6 | 1,845 | 9 | 46 | 26 | 26 | 69 | 16 | 9 | 26 | 24 |
| NaN | 532 | 5 | NaN | 7 | 44 | 24 | 25 | 63 | 15 | 7 | 25 | 20 |
| NaN | 339 | 6 | NaN | 7 | 59 | 30 | 25 | 85 | 20 | 9 | 25 | 29 |
| NaN | 228 | 5 | NaN | 7 | 45 | 23 | 25 | 64 | 15 | 7 | 25 | 20 |
| NaN | 357 | 6 | 362 | NaN | 62 | 31 | 26 | 90 | 21 | 9 | 26 | 30 |
| NaN | 156 | 6 | 7,509 | 6 | 52 | 31 | 25 | 77 | 18 | 10 | 25 | 27 |
| NaN | 329 | 6 | NaN | NaN | 65 | 32 | 25 | 92 | 22 | 11 | 25 | 33 |
| NaN | 550 | 6 | NaN | 8 | 38 | 23 | 25 | 56 | 13 | 7 | 25 | 19 |
| NaN | NaN | 5 | NaN | 7 | 31 | 23 | 24 | 47 | 10 | 7 | 23 | 14 |
| NaN | 1,173 | NaN | NaN | 71 | 99 | 32 | NaN | 87 | NaN | 11 | NaN | NaN |
| NaN | 541 | 6 | 1,595 | NaN | 60 | 31 | 25 | 89 | 21 | 10 | 25 | 32 |
| NaN | 273 | NaN | 248 | 8 | 53 | 37 | NaN | 78 | 17 | 12 | NaN | 27 |
| NaN | 230 | 7 | 196 | 7 | 37 | 24 | 24 | 56 | 12 | 7 | 24 | 17 |
| NaN | 533 | 6 | NaN | 9 | 59 | 31 | 26 | 90 | 19 | 10 | 26 | 29 |
| NaN | NaN | 7 | 205 | 6 | 31 | 32 | 24 | 50 | 16 | 10 | 24 | 17 |
| NaN | 435 | NaN | 504 | NaN | 59 | 35 | NaN | 89 | 23 | NaN | NaN | 31 |
| NaN | 147 | 5 | 137 | 7 | 46 | 29 | 25 | 64 | 15 | 8 | 25 | 20 |
| NaN | 835 | NaN | NaN | 44 | 766 | 50 | 41 | 124 | 40 | 36 | 71 | 44 |
| NaN | 654 | NaN | NaN | 72 | 33 | NaN | 103 | 24 | 12 | NaN | 37 |
| NaN | NaN | NaN | 8,025 | 1,326 | NaN | 903 | 652 | 2,788 | NaN | NaN | 652 | NaN |
| NaN | 334 | 6 | 307 | NaN | 57 | 37 | 24 | 91 | 21 | 13 | 24 | 33 |
| 2,005 | 2,100 | 9 | 909 | 10 | 34 | 33 | 27 | 52 | 14 | 14 | 27 | 20 |

Tab. 2: Iteration counts and times for the 29 benchmark tasks and $\alpha = 0.99$
| F | IO | N | O | ON | ON-T1 | ON-IMP | ON-T2 | ON-EXT | N-T1 | N-IMP | N-T2 | N-EXT |
|---|----|---|---|-----|------|-------|------|-------|------|-------|------|------|
| NaN | 41 | 5 | 21 | 6 | 35 | 22 | 22 | 46 | 10 | 6 | 22 | 13 |
| 29 | 51 | 6 | 28 | 7 | 50 | 36 | 24 | 80 | 18 | 13 | 24 | 29 |
| 29 | 51 | 6 | 9 | 6 | 50 | 36 | 24 | 80 | 18 | 13 | 24 | 29 |
| 122 | 24 | 4 | 19 | 6 | 28 | 22 | 22 | 39 | 8 | 6 | 22 | 11 |
| 18 | 38 | 336 | 13 | 8 | 74 | 26 | 23 | 111 | 23 | 71 | 23 | 37 |
| NaN | NaN | NaN | 7 | NaN | 73 | 39 | 25 | 110 | 26 | 14 | 25 | 39 |
| NaN | 251 | 5 | 723 | 8 | 45 | 24 | 25 | 63 | 14 | 7 | 25 | 20 |
| NaN | NaN | 6 | 9 | 46 | 26 | 26 | 69 | 16 | 9 | 26 | 24 |
| NaN | NaN | 5 | NaN | 7 | 44 | 24 | 25 | 63 | 15 | 7 | 25 | 20 |
| NaN | 383 | 6 | NaN | 7 | 59 | 31 | 25 | 85 | 20 | 9 | 25 | 29 |
| NaN | 243 | 5 | NaN | 7 | 45 | 23 | 25 | 64 | 15 | 7 | 25 | 20 |
| NaN | 442 | 6 | 824 | NaN | 63 | 31 | 26 | 95 | 22 | 9 | 26 | 32 |
| NaN | 164 | 6 | NaN | 6 | 52 | 31 | 25 | 77 | 18 | 10 | 25 | 27 |
| NaN | 396 | 6 | NaN | NaN | 66 | 36 | 26 | 97 | 23 | 12 | 26 | 35 |
| NaN | 812 | 6 | NaN | 8 | 38 | 23 | 25 | 60 | 13 | 7 | 25 | 21 |
| NaN | NaN | 5 | NaN | 7 | 31 | 23 | 23 | 47 | 10 | 7 | 23 | 14 |
| NaN | NaN | NaN | NaN | NaN | 100 | 33 | NaN | 92 | NaN | 12 | NaN | NaN |
| NaN | 848 | 6 | NaN | NaN | 65 | 31 | 25 | 93 | 23 | 10 | 25 | 33 |
| NaN | 323 | NaN | 388 | 8 | 54 | 37 | NaN | 78 | 18 | 12 | NaN | 27 |
| NaN | 253 | 7 | 258 | 7 | 37 | 24 | 24 | 56 | 12 | 7 | 24 | 17 |
| NaN | 857 | 6 | NaN | 9 | 59 | 31 | 26 | 90 | 19 | 10 | 26 | 29 |
| NaN | 334 | 7 | 165 | 6 | 30 | 32 | 24 | 50 | 10 | 16 | 24 | 17 |
| NaN | 577 | NaN | 1.185 | NaN | 59 | 35 | NaN | 94 | 23 | NaN | NaN | 33 |
| NaN | 150 | 7 | 5 | 160 | 46 | 29 | 25 | 69 | 15 | 8 | 25 | 22 |
| NaN | NaN | NaN | NaN | NaN | 81 | 60 | NaN | 126 | 42 | 37 | NaN | 46 |
| NaN | NaN | NaN | NaN | NaN | 72 | 38 | NaN | 108 | 25 | 14 | NaN | 39 |
| 348 | 358 | 50 | 276 | 192 | 108 | 64 | 53 | 71 | 232 | 54 | 53 | 131 |
| NaN | 406 | 7 | 522 | NaN | 57 | 38 | 24 | 91 | 21 | 13 | 24 | 33 |
| 205 | 224 | 12 | 108 | NaN | 48 | NaN | 30 | 56 | 31 | 17 | 30 | NaN |

Tab. 3: Iteration counts and times for the 29 benchmark tensors and $\alpha = 0.999$
Fig. 2: Performance profiles for the 29 benchmark tensors and $\alpha = 0.99$.

Fig. 3: Performance profiles for the 29 benchmark tensors and $\alpha = 0.999$. 
constant number of iterations to converge, but on some of the benchmarks they fail to converge or require a much larger number of iterations. From the point of view of reliability, combining the Perron–Newton algorithm with continuation strategies gives a definite advantage: the resulting methods are the only ones (among the ones considered) that can solve successfully all the problems in the original benchmark set in [1], which contained the experiments with all values of \( \alpha \) up to 0.99. Raising \( \alpha \) to the more extreme value of 0.999 reveals failure cases for these methods as well. (However, if one reduces the step-size selection parameter \( \tau \) to 0.001, methods PN-EXT and PN-T1 succeed on all examples also for \( \alpha = 0.999 \).)

Our results confirm the findings of [1] that problem \( \text{R6.3} \) (the third to last one) for \( \alpha = 0.99 \) is the hardest problem of the benchmark set; Perron-based algorithms can solve it successfully in the end, but (like the algorithms in [1]) they stagnate for a large number of iterations around a point which is far away from the true solution. The analysis in [1, End of Section 6.3 and caption of Figure 9] also attributes this difficulty to the presence of a 'pseudo-solution' that is an attraction basin for a large set of starting points.

To understand the specific difficulties encountered by continuation methods on problem \( \text{R6.3} \), it is instructive to consider the plot in Figure 4. This figure, which has been generated with the help of the software Bertini [17], shows the first entry of each valid stochastic solution \( s \) to (2) (recall that there may be more than one solution), for varying values of \( \alpha \). It turns out that the problem has a single solution for all values of \( \alpha \) up to \( \alpha \approx 0.975 \), then two more solutions appear, initially coincident and then slowly separating one from the other. Two of these three solutions merge at about \( \alpha \approx 0.9899 \) and then disappear (i.e., they are replaced by a pair of complex conjugate solutions), so for \( \alpha = 0.99 \) there is only one real stochastic solution remaining. Note that the points where the number of solutions changes are points in which the Jacobian \( I - G' \alpha \) is singular, in accordance with the implicit function theorem.
Continuation methods ‘track’ the upper solution, and hence fail to provide a good starting point for $\alpha = 0.99$, when it disappears suddenly and is replaced by a completely different solution. Moreover, for $\alpha = 0.99$, the numerical behavior of these iterations is affected by the presence of a point with a very small residual. For an intuition, the reader may consider the behavior of Newton's method (on a function of a single variable) in presence of a local minimum very close to 0, e.g., on $f(x) = x^2 + \varepsilon$ for a very small $\varepsilon$.

7 Conclusions

We have used the theory of quadratic vector equations in [8, 9, 10] to attack the multilinear PageRank problem described in [1]. The tools that we have introduced improve the theoretical understanding of this equation and of its solutions. In particular, considering the minimal solution in addition to the stochastic ones allows one to use a broader array of algorithms, with computational advantage. The new algorithms achieve better results when $\alpha \approx 1$, which is the most interesting and computationally challenging case.

Possible future improvements to these algorithms include experimenting with further step-size selection strategies, and more importantly integrating them with inexact eigenvalue solvers that are more suitable to computation with large sparse matrices.

References

[1] Gleich DF, Lim LH, Yu Y. Multilinear PageRank. *SIAM J. Matrix Anal. Appl.* 2015; 36(4):1507–1541, doi:10.1137/140985160. URL http://dx.doi.org/10.1137/140985160

[2] Li W, Liu D, Ng MK, Vong SW. The uniqueness of multilinear pagerank vectors. *Numerical Linear Algebra with Applications* 2017; 24(6):e2107–n/a, doi:10.1002/nla.2107. URL http://dx.doi.org/10.1002/nla.2107

[3] Li W, Ng MK. On the limiting probability distribution of a transition probability tensor. *Linear Multilinear Algebra* 2014; 62(3):362–385, doi:10.1080/03081087.2013.777436. URL http://dx.doi.org/10.1080/03081087.2013.777436

[4] Lim LH. Singular values and eigenvalues of tensors: A variational approach. *Proceedings of the 1st IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2005; 129–132, doi:10.1109/CAMAP.2005.1574201.

[5] Chang KC, Pearson K, Zhang T. Perron-Frobenius theorem for non-negative tensors. *Commun. Math. Sci.* 2008; 6(2):507–520. URL http://projecteuclid.org/euclid.cms/1214949934

[6] Friedland S, Gaubert S, Han L. Perron-Frobenius theorem for nonnegative multilinear forms and extensions. *Linear Algebra Appl.* 2013; 438(2):738–749, doi:10.1016/j.laa.2011.02.042. URL http://dx.doi.org/10.1016/j.laa.2011.02.042

[7] Hautphenne S, Latouche G, Remiche MA. Newton’s iteration for the extinction probability of a Markovian binary tree. *Linear Algebra Appl.* 2008; 428(11-12):2791–2804, doi:10.1016/j.laa.2007.12.024. URL http://dx.doi.org/10.1016/j.laa.2007.12.024
[8] Bini DA, Meini B, Poloni F. On the solution of a quadratic vector equation arising in Markovian binary trees. *Numer. Linear Algebra Appl.* 2011; 18(6):981–991, doi: 10.1002/nla.809. URL [http://dx.doi.org/10.1002/nla.809](http://dx.doi.org/10.1002/nla.809).

[9] Meini B, Poloni F. A Perron iteration for the solution of a quadratic vector equation arising in Markovian binary trees. *SIAM J. Matrix Anal. Appl.* 2011; 32(1):248–261, doi:10.1137/100796765. URL [http://dx.doi.org/10.1137/100796765](http://dx.doi.org/10.1137/100796765).

[10] Poloni F. Quadratic vector equations. *Linear Algebra and its Applications* 2013; 438(4):1627–1644, doi:10.1016/j.laa.2011.05.036. URL [http://dx.doi.org/10.1016/j.laa.2011.05.036](http://dx.doi.org/10.1016/j.laa.2011.05.036).

[11] Ortega JM, Rheinboldt WC. *Iterative solution of nonlinear equations in several variables*, Classics in Applied Mathematics, vol. 30. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000, doi:10.1137/1.9780898719468. URL [http://dx.doi.org/10.1137/1.9780898719468](http://dx.doi.org/10.1137/1.9780898719468), reprint of the 1970 original.

[12] Li TY. Numerical solution of multivariate polynomial systems by homotopy continuation methods. *Acta numerica, 1997*, *Acta Numer.*, vol. 6. Cambridge Univ. Press, Cambridge, 1997; 399–436, doi:10.1017/S09624929000002749. URL [http://dx.doi.org/10.1017/S09624929000002749](http://dx.doi.org/10.1017/S09624929000002749).

[13] Rudin W. *Principles of mathematical analysis*. Third edn., McGraw-Hill Book Co., New York-Auckland-Düsseldorf, 1976. International Series in Pure and Applied Mathematics.

[14] Gleich DF, Yu Y. *mlpagerank*. Github repository 2015. URL [https://github.com/dgleich/mlpagerank](https://github.com/dgleich/mlpagerank).

[15] Stewart WJ. *Introduction to the numerical solution of Markov chains*. Princeton University Press, Princeton, NJ, 1994.

[16] Higham DJ, Higham NJ. *MATLAB Guide*. Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2000.

[17] Bates DJ, Hauenstein JD, Sommese AJ, Wampler CW. Bertini: Software for numerical algebraic geometry. Available at [http://bertini.nd.edu](http://bertini.nd.edu), doi:10.7274/R0H41PB5.