ACTA ET COMMENTATIONES UNIVERSITATIS TARTUENSIS DE MATHEMATICA
Volume 15, Number 2, 2011
Available online at www.math.ut.ee/acta/

Aggregation/disaggregation as a theoretical tool

Ivo Marek

Abstract. The main aim of this contribution is to establish convergence of some iterative procedures that play an important role in the PageRank computation. The problems we are interested in are considered in two recent papers by Ipsen and Selee, and Lee, Golub and Zenios. Both these papers present new ideas to solve the celebrated problem of the PageRank. Our aim is to show that the results and some generalizations of them can be proven via an application of the iterative aggregation/disaggregation methods. One of the results may be of particular interest. It concerns a proof that the two-stage algorithm proposed by Lee, Golub and Zenios does compute the PageRank. This problem has been raised in the literature. We answer this question in positive by showing appropriate necessary and sufficient conditions. In addition a short proof of the celebrated Google lemma is presented.

1. Introduction

As described in the monograph [6, Chapter 4] the GOOGLE search engine can be considered as a representation of the set of all web pages as an oriented graph. Each vertex represents just one page and an edge in the graph is directed from vertex a to vertex b if and only if page a refers to b. It is also well known that to this graph a Markov chain can be associated. To compute the PageRank means to compute a unique stationary probability vector of the Markov chain mentioned. The web pages referring to none of the pages are called *dangling pages* or *dangling nodes* [6, p. 37]. It is a fact that various research communities use different representations of the mentioned Markov chain. In this paper we use a representation whose stationary probability vectors are right eigenvectors of the Markov chain transition matrix. Since every finite dimensional Banach space is reflexive, a representation whose nontrivial stationary probability vectors are left eigenvectors

Received March 22, 2010.
2010 *Mathematics Subject Classification.* 15A15, 15B51, 65F08.
Key words and phrases. stationary probability vector of Markov chain, iterative aggregation/disaggregation, Google matrix.

are dual with respect to the representation possessing right eigenvectors as stationary probability vectors. Such models utilized by various communities are in the sense just described equivalent.

Let us consider the following two-by-two block matrix

$$H = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}$$

satisfying

$$H_{jk} \geq 0, \quad j,k = 1,2,$$

$$H_{11}^T e(K) + H_{21}^T e(N-K) \quad = e(K),$$

$$H_{12}^T e(K) + H_{22}^T e(N-K) \quad = e(N-K)$$

where $e^T(n) = (1, ..., 1) \in \mathbb{R}^n, n$ is a positive integer.

Here, H_{11} represents the links from the nondangling nodes into the nondangling nodes and H_{21} represents the links from the nondangling nodes to dangling nodes, respectively. The remaining blocks H_{12} and H_{22} can be treated in various ways. For both theoretical as well as computational purposes it is accepted to let $H_{12} = w^{(1)} e^T(N-K)$ and $H_{22} = w^{(2)} e^T(N-K)$.

Let $v \in \mathbb{R}^N_+, e^T v = 1$ and $\alpha \in (0,1)$ be fixed. Then stochastic matrix

$$G = G(\alpha) = \alpha H + (1-\alpha) v e^T,$$

with

$$H_{21} = u^{(2)} e^T, \ H_{22} = w^{(2)} e^T,$$

is called the *Google matrix* and vector v is called its *personalization vector*.

We note that our theory deals with the Google matrix without any additional assumption concerning the structure of the blocks distinct of the block representing the dangling nodes. We are thus able to handle models in which there appear dangling nodes of more than of one type in a manner similar to that invented in [3] and [7] with the same unifying iterative aggregation/disaggregation (IAD) methods.

It is worth to mention that due to the fact that the spectral radius of the Google matrix is a simple root of its characteristic polynomial for every $\alpha \in (0,1)$ [1], the matrix $G(\alpha)$ is convergent and the rate of convergence of the power method applied to $G(\alpha)$ equals α ([1], [2], [4], [11]). A short proof of these statements is presented in Section 7. The method of proof allows to obtain further generalizations.

2. Generalities

As standard, we denote by $\rho(C)$ the spectral radius of matrix C, i.e.

$$\rho(C) = \max \{ |\lambda| : \lambda \in \sigma(C) \},$$

where $\sigma(C)$ denotes the spectrum of C. We call

$$\gamma(C) = \sup\{|\lambda| : \lambda \in \sigma(C), \lambda \neq \rho(C)\}$$

the convergence factor of C.

2.1. Remark. Let C be any $N \times N$ matrix. Then obviously

$$\rho(C) \geq \gamma(C)$$

with possible strict inequality in place of the nonstrict one.

3. IAD communication maps

Let $\mathcal{E} = \mathbb{R}^N, \mathcal{F} = \mathbb{R}^n, n < N,\ e^T = e(N)^T = (1, ..., 1) \in \mathbb{R}^N$. Let $\mathcal{G}$ be a map defined on the index sets:

$$\mathcal{G} : \{1, ..., N\} \overset{onto}{\to} \{\overline{1}, ..., \overline{n}\}.$$

Iterative aggregation/disaggregation communication operators are defined as

$$(Rx)_j = \sum_{\mathcal{G}(j)=(j)} x_j,$$

$$S = S(u),\ (S(u)z)_j = \frac{u_j}{(Ru)_{(j)}}(Rx)_{(j)}.$$

We obviously have

$$RS(u) = I_{\overline{n}},$$

where $I_{\overline{n}}$ denotes the identity matrix of size $\overline{n}$. For the *aggregation projection* $P(x) = S(x)R$,

$$P(x)^T e = e \quad \forall x \in \mathbb{R}^N, x_j > 0, j = 1, ..., N,$$

and

$$P(x)x = x \quad \forall x \in \mathbb{R}^N, x_j > 0, j = 1, ..., N.$$

Define the *aggregated matrix* as

$$\mathcal{B}(x) = RBS(x)$$

and the error matrix as

$$J(B; T; t, s; \mathcal{G}; x^{(0)}; \varepsilon) = T^t \left[I - P(x)\left(B - Q\right)\right]^{-1} \left(I - P(x)\right),$$

where $Q = \hat{x}e^T(N)$ denotes the Perron projection of B in its spectral decomposition and

$$x^T = (x_1, ..., x_N),\ x_j > 0, j = 1, ..., N.$$

Operator $T = M^{-1}W$ is the iteration operator of the splitting

$$I - B = M - W,\ M \text{ invertible}.$$

4. Stationary Probability Vector Algorithm

4.1. Algorithm SPV$(\mathcal{G}; B; T; t, s; x^{(0)}; \varepsilon)$**.**

Let B be an $N \times N$ irreducible stochastic matrix and $\hat{x}$ its unique stationary probability vector. Further, let $T = M^{-1}W$, the iteration matrix corresponding to splitting $I - B = M - W$, be elementwise nonnegative. Finally, let t, s be positive integers, $x^{(0)} \in \mathbb{R}^N$ an elementwise positive vector and let $\varepsilon > 0$ be a tolerance.

Step 1. Set $k = 0$.

Step 2. Construct the aggregated matrix (*in case $s = 1$ irreducibility of B implies that of $\mathcal{B}(x^{(k)})$*)

$$B(x^{(k)}) = RB^s S(x^{(k)}).$$

Step 3. Find the unique stationary probability vector $z^{(k)}$ from

$$\mathcal{B}(x^{(k)})z^{(k)} = z^{(k)},\ e(p)^T z^{(k)} = 1,\ e(p) = (1, ..., 1)^T \in \mathbb{R}^p.$$

Step 4. Let

$$Mx^{(k+1,m)} = Wx^{(k+1,m-1)},\ x^{(k+1,0)} = S(x^{(k)})z^{(k)},\ m = 1, ..., t,$$

$$x^{(k+1)} = x^{(k+1,t)},\ e(N)^T x^{(k+1)} = 1.$$

Step 5. Test whether [1]

$$\|x^{(k+1)} - x^{(k)}\| < \epsilon.$$

Step 6. If NO in Step 6, then let

$$k + 1 \rightarrow k$$

and GO TO Step 2.
Step 7. If YES in Step 6, then set

$$\hat{x} := x^{(k+1)}$$

and STOP.

For the reader's information we present a convergence result needed in the next section.

4.2. Proposition ([9], [10])**.** *Assume the stochastic matrix B possesses a positive diagonal, i.e. $B \geq \beta I$ (elementwise) with some $\beta > 0$. Then the IAD algorithm SPV$(B; T = B; t = 1, s = 1; x^{(0)}; \varepsilon)$ returns convergent sequence: $\lim_{k\rightarrow\infty} x^{(k)} = \hat{x} = B\hat{x},\ e^T\hat{x} = 1$.*

[1] Here the symbol $\|.\|$ denotes any norm on $\mathbb{R}^N$. A good choice is the ℓ_1-norm.

5. Ipsen–Selee Algorithm

We want to find a vector x approximating the unique stationary probability vector $\hat{x}$ satisfying

$$\hat{x} = \begin{pmatrix} \hat{x}^{(1)} \\ \hat{x}^{(2)} \end{pmatrix} = G(\alpha) \begin{pmatrix} \hat{x}^{(1)} \\ \hat{x}^{(2)} \end{pmatrix} = \alpha \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} \begin{pmatrix} \hat{x}^{(1)} \\ \hat{x}^{(2)} \end{pmatrix} + (1-\alpha) \begin{pmatrix} v^{(1)} \\ v^{(2)} \end{pmatrix}.$$

The Ipsen–Selee [3] algorithm reads as follows.

5.1. IS Algorithm.
% Inputs $H, w, v, u^{(2)}, \alpha$. Output $\hat{x}$
% $H_{21} = u^{(2)}e^T, H_{22} = w^{(2)}e^T$
% Power method applied to $G^{(1)}(\alpha)$, where

$$G^{(1)} = \begin{bmatrix} \alpha H_{11} + (1-\alpha)v^{(1)}e^T & \alpha w^{(1)}e^T + (1-\alpha)v^{(1)}e^T \\ \alpha e^T H_{21} + (1-\alpha)e^T v^{(2)}e^T & \alpha e^T w^{(2)}e^T + (1-\alpha)e^T v^{(2)}e^T \end{bmatrix}.$$

Choose a starting vector $\hat{\sigma}^T = [\hat{\sigma}_{1:K}^T, \hat{\sigma}_{k+1}]$ with $\hat{\sigma}^T \geq 0, \|\hat{\sigma}\| = 1$.

While *not converged*

$\hat{\sigma}_{1:K} = \alpha H_{11}\hat{\sigma}_{1:K} + (1-\alpha)v^{(1)} + \alpha w^{(1)}\hat{\sigma}_{K+1}$
$\hat{\sigma}_{K+1} = 1 - e^T\hat{\sigma}_{1:K}$

end while

% Recover PageRank:

$$\hat{x} = \begin{bmatrix} \hat{\sigma}_{1:K} \\ \alpha H_{21}\hat{\sigma}_{1:K} + (1-\alpha)v^{(2)} + \alpha w^{(2)}\hat{\sigma}_{K+1} \end{bmatrix}.$$

The authors of [3] prove the convergence of Algorithm 5.1 by showing that the Google matrix $G(\alpha)$ and the matrix

$$\begin{bmatrix} G^{(1)} & 0 \\ * & 0 \end{bmatrix}, \text{ where } G^{(1)} = \begin{bmatrix} \alpha H_{11} & \alpha w^{(1)} \\ \alpha e^T H_{21} & \alpha e^T w^{(2)} \end{bmatrix},$$

are similar, and an important fact that a detailed knowledge of spectral properties of $G^{(1)}$ is available, and this forms a comfortable base for the elegant provision of the many steps needed in the proof. To this purpose also the lumpability of matrix $G(\alpha)$ is exploited and it is known that this is the case if all dangling nodes are lumped into a single node, or else if block H_{21} is aggregated into a single element [7], [3].

Our way to prove the convergence of the Ipsen–Selee algorithm consists of applying a particular version from a class of the IAD methods namely the algorithm SPV($\mathcal{G}; G(\alpha), T = G(\alpha); t = 1, s = 1; x_0; \varepsilon$) [8] with the following specifications

$$R = \begin{pmatrix} I_K & 0 \\ 0 & \tilde{R} \end{pmatrix} \tag{5.1}$$

where I_K is the identity matrix of order K and

$$\tilde{R}u = e^T u^{(2)},\ u^T = ((u^{(1)})^T,\ (u^{(2)})^T), u \in \mathbb{R}^N_+, u^{(1)} \in \mathbb{R}^K_+ \tag{5.2}$$

and

$$S(u) = \begin{pmatrix} I_K & 0 \\ 0 & \tilde{S}(u) \end{pmatrix}, \tag{5.3}$$

i.e.

$$\tilde{S}(u)z = \begin{pmatrix} z^{(1)} \\ \frac{u^{(2)}}{e^T u^{(2)}} z^{(2)} \end{pmatrix},\ z = \begin{pmatrix} z^{(1)} \\ z^{(2)} \end{pmatrix}. \tag{5.4}$$

The aggregated system then reads

$$\begin{aligned} (I - \alpha H_{11})x^{(1)} &= \alpha H_{12}\tilde{S}(x^{(2)})z^{(2)} + (1-\alpha)v^{(1)}, \\ e^T\left[I - \alpha H_{22}\right]\tilde{S}(x^{(2)})z^{(2)} &= \alpha e^T H_{21}x^{(1)} + (1-\alpha)e^T v^{(2)}. \end{aligned} \tag{5.5}$$

System (5.5) is solved approximately using the method of successive approximations or, alternatively, by the power method on $G^{(1)}$. The convergence of Algorithm 4.1 with the above specifications is guaranteed at first by the convergence of the Google matrix $G(\alpha)$, second by the irreducibility of the aggregated matrix $RG(\alpha)S(x)$, third, by the subsequent unique solvability of the coarse level system and fourth, by an additonal condition requiring the main diagonal diag$\{G(\alpha)\}$ to be positive. The last requirement is very easy to satisfy e.g. just by putting in place of $G(\alpha)$ matrix $\frac{1}{2}(I + G(\alpha))$. It follows that Algorithm 5.1 being equivalent to IAD Algorithm 4.1 with the specifications shown in (5.1) – (5.4) returns convergent sequences of PageRank vectors.

It is worth to recall the result [3] by mentioning that the work needed to get a solution to (5.5), or more precisely, an appropriate approximation requires a large number of multiplications of matrix H_{11} with a vector and a small number of simple algebraic operations implied by the formulae according to (5.5). This large number is proportional to $\tilde{\kappa}K$ where K is the dimension of H_{11}, $\tilde{\kappa}$ is independent of K; pay attention to the fact that the summands of the lower row in (5.5) are nonnegative reals.

Let us note that our approach is free of any lumpability conditions of the Google matrix and therefore, a convergence proof of a generalized problem in which there are $p > 1$ blocks of dangling nodes of different type as presented in [3] is not needed, because our proof utilizing IAD methods is free of requiring lumpability and does apply to an arbitrary block structure of the Google matrix.

6. Lee–Golub–Zenios Algorithm

This is a two-stage algorithm to compute the PageRank [7]. There matrix $G(\alpha)$ reads

$$G(\alpha) = \alpha \begin{pmatrix} H_{11} & u^{(1)}e^T \\ H_{21} & u^{(2)}e^T \end{pmatrix} + (1-\alpha) \begin{pmatrix} v^{(1}e^T \\ v^{(2)}e^T \end{pmatrix}, \quad \alpha \in (\frac{1}{2}, 1),$$

and the following matrices will be involved in the computation:

$$L^{(j)} = R^{(j)}_{LGZ} G(\alpha) S^{(j)}_{LGZ}(y) = \begin{pmatrix} L^{(j)}_{11} & L^{(j)}_{12} \\ L^{(j)}_{21} & L^{(j)}_{22} \end{pmatrix}, \quad j = 1, 2,$$

where $y^T = \left((y^{(1)})^T, (y^{(2)})^T\right), y^{(1)} \in \mathbb{R}^K, y^{(2)} \in \mathbb{R}^{N-K}$,

$$R^{(1)} = R^{(1)}_{LGZ} = \begin{pmatrix} I_K & 0 \\ 0 & e^T(K+1:N) \end{pmatrix} \tag{6.1}$$

and

$$S^{(1)} = S^{(1)}_{LGZ}(y(K+1:N)) = \begin{pmatrix} I_K & 0 \\ 0 & \frac{y(K+1:N)}{e^T(K+1:N)y(K+1:N)} \end{pmatrix}. \tag{6.2}$$

Similarly,

$$R^{(2)} = R^{(2)}_{LGZ} = \begin{pmatrix} e^T(1:K) & 0 \\ 0 & I_{N-K} \end{pmatrix} \tag{6.3}$$

and

$$S^{(2)} = S^{(2)}_{LGZ}(y(1:K)) = \begin{pmatrix} \frac{y(1:K)}{e^T(1:K)y(1:K)} & 0 \\ 0 & I_{N-K} \end{pmatrix}. \tag{6.4}$$

After substitution, we have

$$\begin{aligned} L^{(1)}(y) = \alpha & \begin{pmatrix} H_{11} & H_{12}S_1(y) \\ e^T(1:K)H_{21} & e^T(K+1:N)H_{22}S_1(y) \end{pmatrix} \\ +(1-\alpha) & \begin{pmatrix} v^{(1)}e^T(1:K) & v^{(1)}e^T(K+1:N)S_1(y) \\ R_1 v^{(2)}e^T(1:K) & R_1 v^{(2)}e^T(K+1:N)S_1(y) \end{pmatrix} \end{aligned} \tag{6.5}$$

and

$$\begin{aligned} L^{(2)}(y) = \alpha & \begin{pmatrix} e^T(1:K)H_{11}S_2(y) & e^T(1:K)H_{12} \\ H_{21}S_2(y) & H_{22} \end{pmatrix} \\ +(1-\alpha) & \begin{pmatrix} e^T(1:K)v^{(1)}e^T(1:K)S_2(y) & e^T(1:K)v^{(1)}e^T(K+1:N) \\ v^{(2)}e^T(1:K)S_2(y) & v^{(2)}e^T(K+1:N) \end{pmatrix}. \end{aligned} \tag{6.6}$$

As mentioned in [3], the LGZ-Algorithm is very similar to that of the Ipsen–Selee, actually, the LGZ-Algorithm is its direct predecessor and it reads as follows.

6.1. LGZ Two-stage Algorithm: Stage 1.
% Inputs $L^{(1)}, x(1:K+1)$, *where* $x \in \mathbb{R}^{K+1}$
% w, v, α
Output $\hat{x}(1:K)$
% $H_{21} = w^{(1)}e^T, H_{22} = w^{(2)}e^T$
% *Power method applied to* $L^{(1)}$
Form $L^{(1)}$ *via (6.5)*
select $y \in \mathbb{R}^{K+1}, y \geq 0, \|y\|_1 = 1,$ $\delta = \|y - x\|_1$;
while $\delta \geq \varepsilon$ **do**
$x = y$;

$$\begin{pmatrix} y(1:K) \\ y(K+1) \end{pmatrix} = L^{(1)}(x) \begin{pmatrix} x(1:K) \\ x(K+1) \end{pmatrix}$$

$\delta = \|y - x\|_1$
end

6.2. LGZ Two-stage Algorithm: Stage 2.
% Inputs $L^{(2)}$,
%$e^T(1:K)\hat{x}(1:K)$, *where* $\hat{x} \in \mathbb{R}^N$ is the PageRank
%w, v, α
%Output $\hat{x}(K+1:N)$
% $H_{21} = w^{(1)}e^T, H_{22} = w^{(2)}e^T$
% *Power method applied to* $L^{(2)}$
compute $y^{(k+1)}$ *by setting* $y^{(k+1)} = L^{(2)}(\hat{x}(1:K)y^{(k)}$
form $w = x(\overline{1})e^T(1:K)H_{11}S_2(\hat{x})$ *and* $c = 1 - e^T(1:K)w$ *via (6.6)*
select $x^{(0)} \in \mathbb{R}^{N-K+1}, x^{(0)} \geq 0, \|x^{(0)}\|_1 = 1$;
for $1:3$ **do**
$x^{(k)} = L^{(2)}(y)x^{(k-1)}$
end
if $\|x^{(3)} - x^{(2)}\|_1 < \varepsilon$ **then**
$z = x^{(3)}$
else
% *Aitken extrapolation*
for $k = 1:(N-K+1)$ **do**
$v(k) = \frac{x^{(2)}(k) - x^{(1)}(k)}{x^{(3)}(k) - 2x^{(2)}(k) + x^{(1)}(k)}$
end
$z = x^{(1)} - v$
end

6.3. Lemma. *Let* $\text{diag}G(\alpha)$ *be positive. Let* $\hat{x}^T = ([\hat{x}(1:K)]^T, [\hat{x}(K+1:N-K)]^T)$ *denote the unique stationary probability vector of the Google matrix* $G(\alpha)$. *Assume* $\mathcal{G}_1$ *and* $\mathcal{G}_2$ *are the maps of the set* $\{1,...,N\}$ *onto* $\{1,...,K,\overline{K+1}\}, K < N$, *and* $\{\overline{1}, K+1,...,N\}$, *respectively, and that* $R^{(j)}$

and $S^{(j)} j = 1, 2$, are the communication maps defined by (6.1) – (6.4) accordingly. This includes relations

$$\mathcal{G}_1(j) = \begin{cases} = j, & j = 1, ..., K, \\ = \overline{K+1}, & j \in \{K+1, ..., N\}, \end{cases}$$

and

$$\mathcal{G}_2(j) = \begin{cases} = \overline{1}, & j \in \{1, ..., K\}, \\ = \overline{K+j}, & j = 1, ...N-K\,. \end{cases}$$

Furthermore, let

$$e^T(1:K)\hat{x} = \hat{\xi} \text{ and } e^T(K+1:N)\hat{x}(K+1:N) = 1 - \hat{\xi}.$$

Then stage 1 of the LGZ-Algorithm returns vectors in the form $(x^T_{(1)}, \xi)$ with $x_{(1)} \in \mathbb{R}^K$ and $\xi \in \mathbb{R}^1$, and their first K components approach the vector $\hat{x}(1:K)$, the required stationary probability vector of $G(\alpha)$.

Similarly, stage 2 of the LGZ-Algorithm returns vectors in the form $(\eta, x_{(2)})$ with $\eta \in \mathbb{R}^1$, and $x_{(2)} \in \mathbb{R}^{N-K}$ approaches the appropriate component of the stationary probability vector $\hat{x}(K+1:N))$ if and only if

$$1 > \eta = e^T(K+1:N)\hat{x}(K+1:N)) = 1 - \hat{\xi} > 0. \tag{6.7}$$

Proof. The sufficiency of conditions (6.7) is a consequence of the convergence of both SPV algorithms appearing in our IAD interpretation of the LGZ-Algorithm according to Proposition 4.2. The uniqueness of the stationary probability vector $\hat{x}$ implies necessity of relation (6.7). The proof is thus complete. □

The authors of [3] remark that there is no proof that the two-stage LGZ-Algorithm does compute the PageRank. Lemma 6.3 shows that this is the case if the component representing the aggregated part of the coarse level vector ξ is chosen appropriately and namely

$$\xi = e^T(1:K)\hat{x}(1:K).$$

This adequate choice is available after having computed $\hat{x}(1:K)$ utilizing stage 1 of the LGZ-Algorithm in computing the remaining part $\hat{x}(K:N-K)$ utilizing the stage 2 of the LGZ-Algorithm. Thus, we have

6.4. Theorem. *The two-stage LGZ-Algorithm does compute the PageRank.*

7. A short proof of the Google lemma

It is known that the Google search engine opened unusual interest for its fundamental principles in many areas of research. Our contribution is concerned with the celebrated Google matrix whose importance in computing the PageRank is undisputable. A worldwide discussion concerning many aspects of search engines resulted in many journal publications as well as a monograph [6]. The above mentioned problem how to compute the PageRank efficiently led to an elementary but very interesting result in Linear Algebra, to the so-called Google lemma. Within short time many proofs and generalizations of this lemma have been proposed and with large probability some more will appear. An increasing interest to the specific disciplines of Mathematics and Computer Science as well as many other areas of research directions should be welcome.

We are going to examine the following system of problems parameterized by parameter $\alpha \in (\frac{1}{2}, 1)$:

$$G(\alpha) = \alpha G^{(1)} + (1-\alpha) G^{(2)},$$

where $G^{(1)}$ is a (column) stochastic matrix and $G^{(2)}$ a suitable (low rank) irreducible stochastic matrix.

We establish the following result and present it as follows.

7.1. Lemma. *Suppose $G^{(2)} = ve^T$, where $v = (v_1, ..., v_N)^T$ is a vector whose all components are nonnegative reals and $e^T = (1, ..., 1)$, $e^T v = 1$, i.e. $G^{(2)}$ represents a rank-one stochastic matrix. Then the convergence factor can be bounded as follows*

$$\gamma(G(\alpha)) \leq \alpha.$$

Proof. Let $\hat{x}(\alpha)$ denote the Perron eigenvector. It is easy to see that vector $\hat{x}(\alpha)$ has all its components nonnegative and it can be normalized by setting $e^T \hat{x}(\alpha) = 1$. It follows that

$$\hat{x}(\alpha) = G(\alpha)\hat{x}(\alpha) = \alpha G^{(1)} \hat{x}(\alpha) + (1-\alpha) v$$

and hence

$$\hat{x}(\alpha) = \left[\frac{1}{1-\alpha} (I - \alpha G^{(1)}) \right]^{-1} v.$$

Thus, the Perron projection of $G(\alpha)$ reads

$$Q(\alpha) = \hat{x}(\alpha) e^T.$$

We check easily that

$$Q(\alpha) G(\alpha) Q(\alpha) = G(\alpha) Q(\alpha) = Q(\alpha)$$

and

$$(I - Q(\alpha))\, G^{(2)}\, (I - Q(\alpha)) = \left(G^{(2)} - Q(\alpha)\right)(I - Q(\alpha))$$
$$= G^{(2)}\, (I - Q(\alpha)) = 0. \tag{7.1}$$

The validity of the statement of Lemma 7.1 follows from the relation representing the unique spectral decomposition of matrix $G(\alpha)$

$$G(\alpha) = Q(\alpha) + (I - Q(\alpha))\, \alpha G^{(1)}\, (I - Q(\alpha))\,.$$

□

The above proof opens a way to generalizations. A crucial point in the above proof is a special kind of relationship between the original transition matrix $G^{(1)}$ and the perturbation $G^{(2)}$ consisting of relations (7.1).

8. Concluding remarks

We wanted to show some capabilities of the IAD methods and in particular of the SPV algorithms as theoretical tools for proving convergence of computational procedures studied in [3] and [7]. The methodology of our approach carries characteristics of unifying and systemizing. The IAD methods offer some additional information in comparison with the results established already. In case of [3] this addition consists of showing that the IAD methods of proof are independent of the structure of the number of classes of dangling nodes.

The fact that the LGZ-Algorithm does compute the PageRank is another result that helps to complete our knowledge of Mathematics around the search engine Google. Another feature of our proofs is that convergence is guaranteed for more general models than are those applied in [3] and [7]. On the other hand, one must agree that the restrictive conditions applied in these papers are serious and needed to low computational cost. Seemingly, our proof is new.

The Google matrix lemma is very popular and many proofs have been presented so far. Our method of proof allows some generalizations as a theme for further research.

Acknowledgements. The author expresses his thanks to the anonymous referee for improving the author's English and for supplying the list of references by reference [7].

The author was partly supported by the Grant Agency of the Czech Republic contract number 201/09/1544 and by the Ministry of Schools, Youths and Sports of the Czech Republic via contract MSM 6840770010. The support is gratefully acknowledged.

References

[1] L. Eldén, *The eigenvalues of the Google matrix,* Technical Report LiTH-MAR-R-04-01, Department of Mathematics Linkőping University, Linkőping, Sweden, 2004.

[2] T. H. Haveliwala and S. D. Kamvar, *The second eigenvalue of the Google matrix,* Technical report, Computer Science Department, Stanford University, Palo Alto, 2003.

[3] I. Ipsen and T. Selee, *PageRank computation, with special attention to dangling nodes,* SIAM J. Matrix Anal. Appl. **29**(4) (2007), 1281–1296.

[4] S. D. Kamvar, T. H. Hawelivala, and G. H. Golub, *Extrapolation methods for accelerating PageRank computations.* In: Proceedings of the Twelfth International World Wide Web Conference (WWW03), Toronto, ACM Press, New York, 2003, pp. 261–273.

[5] A. N. Langville and C. D. Meyer, *A reordering for the PageRank problem,* SIAM J. Sci. Comput. **27** (2006), 2112–2120.

[6] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond. The Science of Search Engine Rankings,* Princeton University Press, 2006.

[7] C. P. Lee, G. H. Golub, and S. A. Zenios, *A two-stage algorithm for computing PageRank and multi-stage generalizations,* Internet Math. **4**(4) (2007), 299–328.

[8] I. Marek and P. Mayer, *Convergence theory of a class of aggregation/ disaggregation iterative methods for computing stationary probability vectors of stochastic matrices,* Linear Algebra Appl. **363** (2002), 177–200.

[9] I. Pultarová, *Local convergence analysis of aggregation/disaggregation methods with polynomial correction,* Linear Algebra Appl. **421** (2007), 122–137.

[10] I. Pultarová, *Necessary and sufficient local convergence condition of one class of aggregation-disaggregation methods,* Numer. Linear Algebra Appl. **15** (2008), 339–354.

[11] S. Serra-Capizzano, *Jordan canonical form of the Google matrix.* SIAM J. Matrix Appl. **27** (2005), 305–312.

[12] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains.* Princeton University Press, Princeton, NJ., 1994.

Czech Institute of Technology, School of Civil Engineering, Thakurova 7, 166 29 Praha 6, Czech Republic

E-mail address: `marek@ms.mff.cuni.cz`