DELFT UNIVERSITY OF TECHNOLOGY

REPORT 13-15

The eigenvectors corresponding to the second eigenvalue of the Google matrix and their relation to link spamming

Alex Sangers and Martin B. van Gijzen

ISSN 1389-6520

Reports of the Department of Applied Mathematical Analysis

Delft 2013

The eigenvectors corresponding to the second eigenvalue of the Google matrix and their relation to link spamming

A. Sangers* and M.B. van Gijzen†

December 10, 2013

Abstract

Google uses the PageRank algorithm to determine the relative importance of a website. Link spamming is the name for putting links between websites with no other purpose than to increase the PageRank value of a website. To give a fair result to a search query it is important to detect whether a website is link spammed so that it can be filtered out of the search result.

While the dominant eigenvector of the Google matrix determines the PageRank value, the second eigenvector can be used to detect a certain type of link spamming. We will describe an efficient algorithm for computing a complete set of independent eigenvectors for the second eigenvalue, and explain how this algorithm can be used to detect link spamming. We will illustrate the performance of the algorithm on web crawls of millions of pages.

Keywords: Google PageRank, link spamming, second eigenvector, Markov chains, irreducible closed subsets.

1 Introduction

Google's PageRank algorithm aims to return the best ranking of websites when searching on the web. The PageRank model assumes that a web surfer randomly follows one of the outgoing hyperlinks at a given website with a chance p or jump to a random website with chance $1 - p$. Mathematically this can be modeled by a Markov chain. The PageRank of a website is the probability to be on this website in the stationary distribution of the Markov chain. This stationary distribution is given by the first eigenvector of the transition matrix of the Markov chain.

According to Haveliwala and Kamvar [6] the eigenvectors for the second eigenvalue are also of importance: they can be used to detect link spam. Link spam is the name for putting links between web pages with no other purpose than to increase the PageRank of a website. Specifically, in the conclusions of [6] Haveliwala and Kamvar state that "The eigenvectors corresponding to the second eigenvalue $\lambda_2 = p$ are an artifact of certain structures in the web graph. In particular, each pair of leaf nodes in the SCC[1] graph for the chain $\mathbf{P}$ corresponds to an eigenvector of $\mathbf{A}$ with eigenvalue p. These leave nodes in the SCC are those subgraphs in the web link graph

*Delft University of Technology, Delft Institute of Applied Mathematics, Mekelweg 4, 2628 CD, The Netherlands. E-mail: `A.Sangers@student.tudelft.nl`

†Delft University of Technology, Delft Institute of Applied Mathematics, Mekelweg 4, 2628 CD, The Netherlands. E-mail: `M.B.vanGijzen@tudelft.nl`

[1] Strongly Connected Component

which have incoming edges, but have no edges to other components. Link spammers often generate such structures in attempts to hoard rank. Analysis of the nonprincipal eigenvectors of $\mathbf{A}$ may lead to strategies for combating link spam."

In this paper we will explain this remark. We will review the theory about the second eigenvalue of the Google Matrix that is described in [5] and in [6] and extend it with results for the corresponding eigenvectors. We will use our findings to propose an efficient algorithm to detect these structures in the web that may indicate link spamming. We will illustrate the performance of the algorithm on web crawls containing several millions of pages.

This structure of this paper is as follows. Section 2 explains the structure of the Google Matrix and gives different methods for computing the PageRank. Section 3 discusses the relation between irreducible closed subsets in a graph and link spamming. Section 4 gives the relevant theory for the second eigenvalue and the corresponding eigenvectors of the Google Matrix. It also explains how the second eigenvectors are related to the irreducible closed subsets. Section 5 describes two algorithms for computing the second eigenvectors. Section 6 compares the performance of the algorithms on web crawls of several millions of pages. Section 7 summarizes our findings and makes some concluding remarks.

Remarks on notation and terminology: The terms 'web sites', 'web pages' and 'nodes' as well as the terms 'hyperlinks' and 'web links' are used interchangeably.
The i-th eigenvector is written as $\mathbf{x^{(i)}}$ and the j-th element of vector $\mathbf{x}$ is written as x_j. A submatrix of matrix $\mathbf{A}$ will be denoted by $\mathbf{A_{ij}}$ and an element of $\mathbf{A}$ by a_{ij}.

2 The Google Matrix

We introduce W, a set of the web pages, that are connected to each other by hyperlinks, i.e., incoming and outgoing links between web pages. The mathematical representation of W is a directed graph, in which a directed link between nodes of the graph represents an incoming or outgoing link between web pages.

Let n be the number of websites. Further, let $\mathbf{G}$ be the n-by-n connectivity matrix with $g_{ij} = 1$ if there is an outgoing hyperlink from page j to i and $g_{ij} = 0$ otherwise. $\mathbf{G}$ is the matrix representation of W. The number of websites n is extremely large, hundreds of millions, while every website only contains a few outgoing links. The matrix $\mathbf{G}$ is therefore large and sparse.

We denote by c_j the column sums of $\mathbf{G}$, that is $c_j = \sum_i g_{ij}$. Note that c_j is the number of outgoing hyperlinks of website j. We will also call this the out-degree of page j.

Surfing the web can be modeled as a Markov process, where one state transitions into another state by following hyperlinks. In order to model this process we introduce the row-stochastic matrix $\mathbf{P}$. The entries p_{ji} of $\mathbf{P}$ are given by

$$p_{ji} = \begin{cases} g_{ij}/c_j & \text{if } c_j \neq 0, \\ 1/n & \text{if } c_j = 0. \end{cases} \tag{2.1}$$

Note that $\mathbf{P^T}$ is the column-stochastic transition probability matrix of the Markov process. Nodes without outgoing hyperlink are called *dangling nodes*. From (2.1) follows that from a dangling node all pages can be reached with equal probability. Following [9], we assume that self-referencing nodes, i.e., $g_{ii} = 1$ for node i, are not allowed.

The above Markov process does not capture the possibility that a web surfer jumps to another page without following an outlink. To include this behavior, called teleportation, Google's PageRank model assumes that an outlink is followed with chance p and a jump to a random page is made with chance $1-p$. Typically, p is chosen between 0.85 and 0.99.

Let $\mathbf{A}$ be the n-by-n column-stochastic transition matrix of this Markov process that includes teleportation. The elements a_{ij} of this matrix are given by

$$a_{ij} = \begin{cases} pg_{ij}/c_j + (1-p)/n & \text{if } c_j \neq 0. \\ 1/n & \text{if } c_j = 0. \end{cases}$$

In matrix notation this can be written as

$$\mathbf{A} = p\mathbf{P^T} + \frac{(1-p)}{n}\mathbf{ee^T},$$

with $\mathbf{e}$ is the n-vector of all ones. Also, recognize that if page j is a dangling node then each page has a chance $1/n$ to be chosen. Thus, if column $\mathbf{a_j} = \mathbf{e}/n$ then page j is a dangling node.

By introducing the diagonal matrix $\mathbf{D}$, of which the main diagonal elements d_{jj} are defined by

$$d_{jj} = \begin{cases} 1/c_j & \text{if } c_j \neq 0 \\ 0 & \text{if } c_j = 0, \end{cases}$$

and by defining the vector $\mathbf{z}$ with coefficients z_j given by

$$z_j = \begin{cases} (1-p)/n & \text{if } c_j \neq 0 \\ 1/n & \text{if } c_j = 0, \end{cases}$$

the matrix $\mathbf{A}$ can also be written as

$$\mathbf{A} = p\mathbf{GD} + \mathbf{ez^T}\ .$$

The matrix $\mathbf{ez^T}$ accounts for teleportation. Note that as a consequence of this teleportation matrix, $\mathbf{A}$ is positive, meaning that every entry is positive, and is irreducible.

The PageRank is determined as the eigenvector of the dominant eigenvalue of the following system:

$$\mathbf{Ax^{(1)}} = \lambda_1\mathbf{x^{(1)}}.$$

Intuitively, when recalling the random web surfer from Section 1, the eigenvector $\mathbf{x^{(1)}}$ is the distribution of the visiting frequency for each node. The more often the surfer passes node j, the higher its PageRank will be.

The matrix $\mathbf{A}$ has a simple dominant eigenvalue, with corresponding positive eigenvector $\mathbf{x^{(1)}}$. This follows from the well-known Perron-Frobenius theorem (see e.g. [8]) for irreducible, square, nonnegative matrices. A nonnegative matrix is a matrix of which all entries are nonnegative.

Theorem 2.1. *(Perron-Frobenius) Let* $\mathbf{A}$ *be a square irreducible nonnegative matrix. Then* $\mathbf{A}$ *has a unique positive real eigenvalue* λ_1 *equal to its spectral radius. The eigenvector corresponding to* λ_1 *is positive. If* $\mathbf{A}$ *is positive, then* λ_1 *is dominant.*

It can be shown [8] that the dominant eigenvalue λ_1 satisfies the following inequalities:

$$\min_j \sum_i a_{ij} \leq \lambda_1 \leq \max_j \sum_i a_{ij}.$$

All column sums of $\mathbf{A}$ are equal to one, so it immediately follows that $\lambda_1 = 1$. Since λ_1 is a simple eigenvalue

$$\mathbf{x^{(1)}} = \mathbf{A}\mathbf{x^{(1)}} \tag{2.2}$$

has a unique solution up to a scaling factor. If this scaling factor is chosen such that $\sum_i x_i^{(1)} = 1$ (or, by positivity: $||\mathbf{x^{(1)}}||_1 = 1$), then $\mathbf{x^{(1)}}$ is the stationary stochastic vector of the Markov chain and also, $\mathbf{x^{(1)}}$ is the Google PageRank vector.

2.1 Computing the PageRank vector

The most common way to solve a large system in Equation (2.2) is the power method. The power method starts with a guess $\mathbf{u_0}$ and then we iteratively compute $\mathbf{u_{k+1}} = \mathbf{A}\mathbf{u_k}$. After each iteration we scale $\mathbf{u_k}$ with $||\mathbf{u_k}||_1 = 1$ to make sure $\mathbf{u_k}$ sums up to 1 and thus is stochastic.

To perform a power iteration, only a matrix-vector multiplication with $\mathbf{A}$ needs to be performed. This operation can be performed cheaply as follows: $\mathbf{u_{k+1}} = p\mathbf{G}\mathbf{D}\mathbf{u_k} + \mathbf{e}(\mathbf{z^T}\mathbf{u_k})$. We refer to [9] for more information.

An alternative way to compute the PageRank is by rewriting Equation (2.2) as a linear system

$$(\mathbf{I} - p\mathbf{G}\mathbf{D})\mathbf{x^{(1)}} = \beta\mathbf{e} \tag{2.3}$$

with $\beta = \mathbf{z^T}\mathbf{x^{(1)}}$. Note that we do not know the value of scalar β, but we take $\beta = 1$ so the equation can be solved explicitly. Then $\mathbf{x^{(1)}}$ can be rescaled so that $\sum_i x_i^{(1)} = 1$.

3 Irreducible closed subsets and link spamming

A typical technique to increase the PageRank of a group of websites is to create many inlinks to the group, and to remove all outlinks. In this way, it is easy for the random surfer to enter the group, but difficult to leave since he can only escape from this group through teleportation.

To illustrate this we consider the example given by Figure 1. The PageRank vector for this example is given by

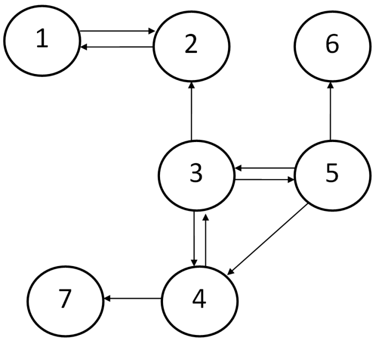

Figure 1: Simple directed graph

$$\mathbf{x^{(1)}}^T = \begin{pmatrix} 0.318 & 0.332 & 0.087 & 0.078 & 0.061 & 0.054 & 0.070 \end{pmatrix}.$$

The nodes with the highest PageRanks are number 1 and 2. Note that nodes 6 and 7 are dangling nodes. By definition, dangling nodes are connected to all other nodes.

Now we illustrate how to increase the PageRank of node 4. First we remove dangling node 7 by making a link back to node 4. Next we remove the outlink form node 4 to node 3. We refer to Figure 2 for the resulting graph. The PageRank vector after these modifications becomes

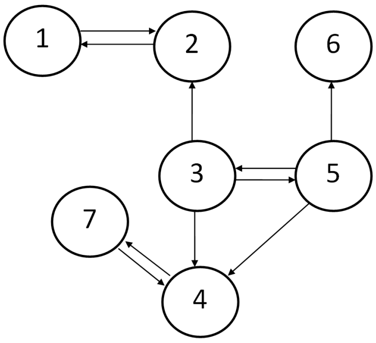

Figure 2: Changes to improve the PageRank for node 4.

$$\mathbf{x}^{(\mathbf{1})^T} = \begin{pmatrix} 0.203 & 0.209 & 0.036 & 0.246 & 0.036 & 0.036 & 0.235 \end{pmatrix}.$$

Clearly, node 4 now has the highest PageRank.

To analyse this we will recall some well known definitions.

Definition 3.1. *A set of states S is a closed subset of the Markov chain corresponding to $\mathbf{P^T}$ if and only if $i \in S$ and $j \notin S$ implies that $p_{ji} = 0$.*

Definition 3.1 tells us that a Markov chain is closed if it is not possible to get out of subset S as soon as you are in it. This means that any subset containing a dangling node cannot be closed, and in particular, any dangling node cannot be a a closed subset.

Definition 3.2. *A set of states S is an irreducible closed subset of the Markov chain corresponding to $\mathbf{P^T}$ if and only if S is a closed subset, and no proper subset of S is a closed subset.*

Let l be the number of irreducible closed subsets of $\mathbf{P}$. Then we can rewrite $\mathbf{P}$ in canonical form ([8]) by renumbering the nodes:

$$\mathbf{P} \sim \begin{pmatrix} \mathbf{T_{11}} & \mathbf{T_{12}} \\ \mathbf{0} & \mathbf{T_{22}} \end{pmatrix} = \left(\begin{array}{cccc|cccc} \mathbf{P_{11}} & \mathbf{P_{12}} & \cdots & \mathbf{P_{1r}} & \mathbf{P_{1,r+1}} & \mathbf{P_{1,r+2}} & \cdots & \mathbf{P_{1m}} \\ \mathbf{0} & \mathbf{P_{22}} & \cdots & \mathbf{P_{2r}} & \mathbf{P_{2,r+1}} & \mathbf{P_{2,r+2}} & \cdots & \mathbf{P_{2m}} \\ \vdots & & \ddots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{P_{rr}} & \mathbf{P_{r,r+1}} & \mathbf{P_{r,r+2}} & \cdots & \mathbf{P_{rm}} \\ \hline \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{P_{r+1,r+1}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{P_{r+2,r+2}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{P_{mm}} \end{array}\right), \tag{3.1}$$

where $l = m - r$ and each $\mathbf{P_{11}}, \ldots, \mathbf{P_{rr}}$ is either irreducible or $[\mathbf{0}]_{\mathbf{1\times1}}$, and $\mathbf{P_{r+1,r+1}}, \ldots, \mathbf{P_{mm}}$ are irreducible and closed. First, note that each $\mathbf{P_{ij}}$ is a submatrix of the n-by-n matrix $\mathbf{P}$. Let us call the dimension of the block $\mathbf{T_{11}}$ $\tilde{r}$-by-$\tilde{r}$ and thus, the dimension of the block $\mathbf{T_{22}}$ is $(n - \tilde{r})$-by-$(n - \tilde{r})$.

3.1 Example

We illustrate the theory by the graph displayed in Figure 2. Firstly, we will renumber the nodes to get the canonical form as in (3.1). For a graphical representation of the renumbering, we refer to Figure 3.

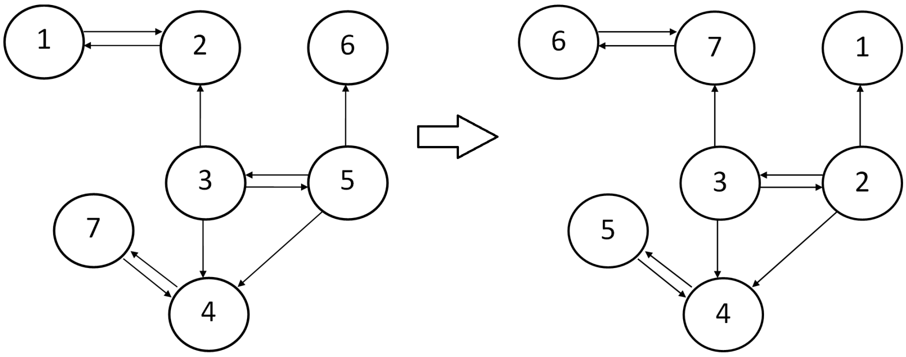

Figure 3: Renumbering the nodes of Figure 2 to canonical form.

Thus, rewriting $\mathbf{P}$ to $\mathbf{P_{canon}}$:

$$\begin{aligned}\mathbf{P} &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \\ &\sim \left(\begin{array}{ccc|cc|cc} \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \cline{4-7} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right) = \mathbf{P_{canon}}. \end{aligned} \tag{3.2}$$

Let us take a closer look at $\mathbf{P_{canon}}$ in (3.2). Firstly, we recognize the block on the lower left side of all zeros. Also, it is clear that we have two irreducible closed subsets ($\mathbf{P_{22}}$ and $\mathbf{P_{33}}$), which can be reached by $\mathbf{T_{12}}$. However, $\mathbf{T_{12}}$ includes all other nodes that are not in $\mathbf{T_{22}}$ and thus, $\mathbf{P_{11}}$ is the only block in the upper left side of $\mathbf{P_{canon}}$ (i.e., there are no nodes that do not refer to one of the irreducible closed subsets). Note that $\mathbf{P_{11}}$ is irreducible, but not closed. $\mathbf{P_{22}}$ and $\mathbf{P_{33}}$ are irreducible and closed.

4 The second eigenvector and its relation to link spamming

To explain the relation of the second eigenvector to link spamming we review some results from [5] and [6]. The following lemma can be found in [6]:

Lemma 4.1. *Every eigenvector* $\mathbf{x^{(2)}}$ *corresponding to the second eigenvalue of* $\mathbf{A}$ *is orthogonal to* $\mathbf{e}$*:* $\mathbf{e^T x^{(2)}} = 0$.

Below we give a sketch of the proof. For the complete proof we refer to [6].

Proof. Since $\mathbf{A}$ is column stochastic, $\mathbf{e}$ is a left eigenvector of $\mathbf{A}$ corresponding to the dominant eigenvalue $\lambda_1 = 1$. The lemma follows from the fact that the left and right eigenvectors are bi-orthogonal. □

Lemma 4.1 gives rise to the following theorem.

Theorem 4.2. *Every eigenvector* $\mathbf{x^{(2)}}$ *corresponding to the second eigenvalue* $\mathbf{A}$ *is an eigenvector of* $\mathbf{P^T}$*.*

Proof. The second eigenvector $\mathbf{x^{(2)}}$ of $\mathbf{A}$ satisfies

$$\left(p(\mathbf{P^T} + \frac{1-p}{n}\mathbf{ee^T}\right)\mathbf{x^{(2)}} = \lambda^{(2)}\mathbf{x^{(2)}}.$$

Using Lemma 4.1 yields

$$\mathbf{P^T x^{(2)}} = \frac{\lambda^{(2)}}{p}\mathbf{x^{(2)}} \ ,$$

which proves the theorem. □

The first left eigenvector(s) of $\mathbf{P}$ have a special structure, which becomes clear from the canonical form of $\mathbf{P}$. We assume that $\mathbf{T_{22}}$ is non-empty. The eigenvector(s) corresponding to eigenvalue $\gamma_i = 1$ for $\mathbf{P}$ in canonical form satisfy

$$\begin{gathered}\mathbf{y^T P} = \gamma_i \mathbf{y^T} \\ \begin{pmatrix} \mathbf{y_1^T}, & \mathbf{y_2^T} \end{pmatrix}\begin{pmatrix} \mathbf{T_{11}} & \mathbf{T_{12}} \\ \mathbf{0} & \mathbf{T_{22}} \end{pmatrix} = \begin{pmatrix} \mathbf{y_1^T}, & \mathbf{y_2^T} \end{pmatrix} \\ \Rightarrow \begin{cases} \mathbf{y_1^T T_{11}} & = \mathbf{y_1^T} \\ \mathbf{y_1^T T_{12}} + \mathbf{y_2^T T_{22}} & = \mathbf{y_2^T} \end{cases}\end{gathered} \tag{4.1}$$

We know that $(\mathbf{T_{11}} - \mathbf{I})$ is non-singular, since $|\gamma_i| < 1$ for $\mathbf{T_{11}}$ (refer to [8], page 698). Therefore, Equation (4.1) implies $\mathbf{y_1^T} = \mathbf{0}$. It follows that $\mathbf{y_2^T T_{22}} = \mathbf{y_2^T}$.

We get $\mathbf{y_2^T}(\mathbf{T_{22}} - \mathbf{I}) = \mathbf{0}$, where $(\mathbf{T_{22}} - \mathbf{I})$ is singular and thus, $\mathbf{y_2}$ is a left eigenvector of $\mathbf{T_{22}}$ corresponding to $\gamma_i = 1$. Each submatrix $\mathbf{P_{r+j,r+j}}$ $(1 \leq j \leq l)$ in $\mathbf{T_{22}}$ is row-stochastic and therefore has eigenvalue 1. This leads to the following lemma ([7], page 126):

Lemma 4.3. *The multiplicity of the eigenvalue* 1 *for* $\mathbf{P}$ *is equal to the number of irreducible closed subsets of* $\mathbf{P}$*.*

Let $\mathbf{y_{r+j}}$ be the dominant left eigenvector of $\mathbf{P_{r+j,r+j}}$. Since $\mathbf{P_{r+j,r+j}}$ is irreducible and has only nonnegative entries, this eigenvector can be scaled to be positive and stochastic by Theorem 2.1. Let $\mathbf{\bar{y}_{r+j}}$ be the vector that results from padding the stochastic vector $\mathbf{y_{r+j}}$ with zeros to get the appropriate size n. Every dominant left eigenvector of $\mathbf{P}$ can be written as a linear combination of the vectors $\mathbf{\bar{y}_{r+j}}$, $j = 1, \ldots, m - r$:

$$\mathbf{y} = \sum_{j=1}^{m-r} \alpha_j \mathbf{\bar{y}_{r+j}} \ . \tag{4.2}$$

Using Lemma 4.1 and Theorem 4.2 we can now construct $m-r-1$ independent second eigenvectors $\mathbf{x}^{(2)} \perp \mathbf{e}$ for $\mathbf{A}$:

$$\mathbf{x^{(2)}} = \mathbf{\bar{y}^T_{r+j}} - \mathbf{\bar{y}^T_{r+j+1}}, \quad j = 1, \cdots m-r-1 \ . \tag{4.3}$$

Here we have assumed that there are at least two irreducible closed subsets and we used that the eigenvectors $\mathbf{y_{r+j}}$ are stochastic.

The following theorem that can be found in [5] and [6] is a direct consequence of the discussion above.

Theorem 4.4. *If $\mathbf{P^T}$ has at least two irreducible closed subsets, then the second eigenvalue of $\mathbf{A}$ is $\lambda_2 = p$, with $1-p$ the teleportation chance as introduced in Section 2.*

These second eigenvectors of $\mathbf{A}$ have the following special nonzero structure that is characterized by Theorem 4.5.

Theorem 4.5. *Let $\mathbf{x^{(2)}} = (x_1, \cdots, x_n)^T$ be an eigenvector of $\mathbf{A}$ corresponding to the eigenvalue p. Then $x_j = 0$ if $j \notin$ irreducible closed subset.*

Proof. The proof follows from Equations (4.2) and (4.3). □

5 Computation of all the eigenvectors that correspond to the second eigenvalue of A

In this section we assume that we have a set W of websites with at least two irreducible closed subsets, so we know that p is the second eigenvalue of $\mathbf{A}$. We will present two algorithms for computing all the eigenvectors that correspond to this eigenvalue.

5.1 Computation of the eigenvectors for eigenvalue p of A by computing all the irreducible closed subsets of W

The first algorithm computes the eigenvectors for eigenvalue p of $\mathbf{A}$ by computing all the irreducible closed subsets of W. As we mentioned before, a directed graph is irreducible if, given any two nodes, there exists a directed path from the first node to the second. This is equivalent to the directed graph being strongly connected. Determining all the strongly connected components in the graph for W therefore allows us to determine the submatrices $\mathbf{P_{ii}}$ in Equation (3.1). Whether $\mathbf{P_{ii}}$ corresponds to a closed subset can be determined by inspecting whether there are outlinks to the subset corresponding to $\mathbf{P_{ii}}$. There are no outlinks to this set if $\mathbf{P_{ij}} = \mathbf{O}$, $j = 1, \ldots, n, j \neq i$. Several efficient algorithms exist for determining these strongly connected components. One of the most efficient ones is Tarjan's algorithm [11]. An efficient Matlab routine that implements Tarjan's algorithm is `graphconncomp` [1].

Once the the $m-r$ submatrices $\mathbf{P_{r+j,r+j}}$ have been determined, we can compute their dominant left eigenvectors $\mathbf{y_{r+j}}$. This can be done by solving the homogeneous equation

$$(\mathbf{P^T_{r+j,r+j}} - \mathbf{I})\mathbf{y_{r+j}} = \mathbf{0} \ . \tag{5.1}$$

A technique to compute a solution is to apply an unpreconditioned Krylov subspace method to this system with a nonzero initial guess $\mathbf{x_0}$. In our experiment we use IDR(s) [12] to solve Equation (5.1). The vector $\mathbf{y_{r+j}}$ must be normalized to make it stochastic and padded with zeros to give $\mathbf{\bar{y}_{r+j}}$. The $m-r-1$ eigenvectors $\mathbf{x^{(2)}}$ of $\mathbf{A}$ then follow from Equation (4.3).

We will denote the resulting algorithm by Tarjan-based algorithm. It is summarized as follows:

1. Apply Tarjan's algorithm to the graph W. The strongly connected components without outlinks are irreducible closed subsets;

2. Form the matrices $\mathbf{P_{r+j,r+j}}$ that correspond to the irreducible closed subsets;

3. Compute the dominant eigenvectors $\mathbf{y_{r+j,r+j}}$ of the matrices $\mathbf{P_{r+j,r+j}}$ by solving Equation (5.1), scale them to make them stochastic, and pad them with zeros to the appropriate size. This results in the vectors $\mathbf{\bar{y}_{r+j,r+j}}$;

4. Combine the vectors $\mathbf{\bar{y}_{r+j,r+j}}$ pairwise using Equation (4.3) to compute second eigenvectors of $\mathbf{A}$.

Remarks: To detect link spamming, only the irreducible closed subsets need to be computed in step 1. The eigenvectors for the second eigenvalue as computed in step 4 are sparse, the total amount of nonzeroes in these vectors cannot exceed $2n$.

5.2 Computation of all the eigenvectors for eigenvalue p of A by computing one second eigenvector of A

The second algorithm that we present uses the nonzero structure of the second eigenvectors of $\mathbf{A}$ that is given in Theorem 4.5. Nonzero components of the second eigenvector correspond to nodes in an irreducible closed subset. The idea is to compute one second eigenvector and determine all the nonzero elements. An arbitrary second eigenvector of $\mathbf{A}$ has with high probability nonzero values in all the entries that correspond to nodes in irreducible closed subsets. The second eigenvectors of $\mathbf{A}$ are eigenvectors of $\mathbf{P^T}$ corresponding to the eigenvalue 1. One second eigenvector of $\mathbf{A}$ can therefore be computed by solving the homogeneous system

$$(\mathbf{P^T} - \mathbf{I})\mathbf{y} = \mathbf{0} \quad . \tag{5.2}$$

To detect which nodes are in the same irreducible closed subset, we form a directed graph that only consists of the nodes that correspond to nonzero values in $\mathbf{y}$. We apply Tarjan's algorithm to this graph, that is of much smaller size than the original graph W. The strongly connected components in this graph correspond to irreducible closed subsets. Once we have found all the nodes that constitute an irreducible closed subset we can form the corresponding matrix $\mathbf{P_{r+j,r+j}}$. Of each of these matrices we compute the dominant left eigenvector $\mathbf{y_{r+j}}$, and these vectors are then combined to second eigenvectors of $\mathbf{A}$ using Equation (4.3).

We will denote the resulting algorithm by eigenvector-based algorithm. It is summarized as follows:

1. Compute one dominant eigenvector of $\mathbf{P^T}$ by solving Equation (5.2). This can be done by using applying a Krylov method to the homogeneous system with a nonzero initial guess;

2. Determine the nonzero coefficients;

3. Apply Tarjan's algorithm to the graph formed by the nonzero nodes. The strongly connected components in this graph are irreducible closed subsets;

4. Form the matrices $\mathbf{P_{r+j,r+j}}$ that correspond to the irreducible closed subsets;

5. Compute the dominant eigenvectors $\mathbf{y_{r+j,r+j}}$ of the matrices $\mathbf{P_{r+j,r+j}}$ by solving Equation (5.1), scale them to make them stochastic, and pad them with zeros to the appropriate size. This results in the vectors $\mathbf{\bar{y}_{r+j,r+j}}$;

Test problem	Size	IDR(1) iterations	CPU time
wb-cs-stanford	9914	57	0.2
flickr	820878	45	18.6
wikipedia-20051105	1634989	63	40.7
wikipedia-20060925	2983494	68	73.6
wikipedia-20061104	3148440	63	66.5
wikipedia-20070206	3566907	66	78.0
wb-edu	9845725	103	152.4

Table 1: Iterations and CPU-time to compute the first eigenvector.

6. Combine the vectors $\bar{\mathbf{y}}_{\mathbf{r+j,r+j}}$ pairwise using Equation (4.3) to compute second eigenvectors of $\mathbf{A}$.

Remarks: To detect link spamming, only the irreducible closed subsets need to be computed in step 1-3. The eigenvectors for the second eigenvalue as computed in step 6 are sparse, the total amount of nonzeroes in these vectors cannot exceed $2n$.

6 Numerical experiments

As test problems we consider 7 matrices from the University of Florida Sparse Matrix Collection [4]. These matrices correspond to web crawls and have been contributed by David Gleich. The problem sizes correspond to approximately 10^3 pages for the smallest test problem to 10^7 pages for the largest problem. The connectivity matrices $\mathbf{G}$ as included in the Florida Sparse Matrix Collection are defined as $g_{i,j} = 1$ if page i links to page j, which corresponds to the reversed direction with respect to the definition we use for the matrix $\mathbf{G}$. Moreover, the main diagonal elements of the matrices $\mathbf{G}$ are not all zero. Since we do not allow self-referencing, we set the main diagonal elements to zero. The matrices are therefore pre-prossed as follows

$$\mathbf{G} = \mathbf{G}^T - diag(\mathbf{G}) \ .$$

All computations have been performed using Matlab 7.13 on a workstation with 32 GB of memory and equipped with an 8 core Xeon processor.

We first determine the PageRank by solving system (2.3) using IDR(1). As termination criterion we use

$$\frac{\|\mathbf{r}_i\|}{\|\mathbf{r}_0\|} < 10^{-8},$$

in which $\mathbf{r}_i$ is the residual after i iterations. These systems are very well conditioned, which means that the convergence of IDR(s) is not influenced a lot by the choice of s. For this reason we have selected $s = 1$, the choice with lowest vector overhead. Table 1 gives in the first column the name of the test problem, in the second column the size of the matrix, (number of pages), in the third column the number of IDR(1) iterations, and in the fourth column the CPU-times. Note that the number of IDR(1) iterations only depends very mildly on the problem size.

We have applied the two algorithms of the previous section to detect the irreducible closed subsets and the second eigenvectors of $\mathbf{A}$. Table 2 gives the results for Tarjan's algorithm and Table 2 for the eigenvector-based algorithm.

System (5.2) can be quite ill-conditioned. For this reason, we use IDR(4) to solve Equation (5.2), which gives a considerable reduction of the number of iterations compared to IDR(1). The

Test problem	Size	Number of irreducible closed subsets	CPU-time
wb-cs-stanford	9914	113	0.3
flickr	820878	5394	399.3
wikipedia-20051105	1634989	68	1515.3
wikipedia-20060925	2983494	63	5077.1
wikipedia-20061104	3148440	59	5696.9
wikipedia-20070206	3566907	58	7462.7
wb-edu	9845725	49573	75703.2

Table 2: Results for Tarjan's algorithm to compute irreducible closed subsets.

Test problem	Size	Number of irreducible closed subsets	IDR(4) iterations	CPU-time
wb-cs-stanford	9914	350	113	1.4
flickr	820878	5394	312	160.8
wikipedia-20051105	1634989	68	169	140.2
wikipedia-20060925	2983494	63	126	166.6
wikipedia-20061104	3148440	59	112	155.1
wikipedia-20070206	3566907	58	176	313.6
wb-edu	9845725	85470	1000	2825.6

Table 3: Results for the eigenvector-based algorithm to compute irreducible closed subsets.

termination criterion we use is

$$\frac{\|\mathbf{r}_i\|}{\|\mathbf{r}_0\|} < 10^{-12},$$

which is more strict than for the computation of the PageRank, but needed in practice to determine if a coefficient of the solution vector equals zero. The iterative method was stopped if the number of iterations exceeded 1000, which was the case for test problem wb-edu. As a result an incorrect number of 85470 irreducible closed subesets was found, yielding 85469 computed eigenvectors for eigenvalue $p = 0.85$. After checking the Rayleigh quotients for these computed eigenvectors it turned out that of these 85469 vectors, 41605 corresponded to actual eigenvectors for p. After this correction, the number of detected irreducible closed subsets becomes 41606.

As is clear from the results in the above tables, the eigenvector-based algorithm gives a big computational advantage: the computing time is 10 to 20 times less for the larger test problem. For the eigenvector-based algorithm, the solution of the linear system (5.2) takes almost all of the computing time. This is similar to the computation of the PageRank, where the solution of Equation (2.3) takes all the computing time. However, since system (2.3) is much better conditioned than (5.2), the solution of Equation (2.3) is considerably more time consuming than of Equation (5.2), and hence the computation of the PageRank is much faster than the detection of possible link spamming.

7 Conclusion

In this paper we have examined the second eigenvector of the Google matrix and its relation to link spamming. Creating an irreducible closed subset is an effective way of link spamming.

Irreducible closed subsets can be found with the second eigenvector of the Google matrix. The second eigenvectors of $\mathbf{A}$ are first eigenvectors of $\mathbf{P^T}$. The elements of such eigenvectors have with high probability nonzero value in the nodes that correspond to irreducible closed subset and zero value in other nodes.

The second eigenvectors of $\mathbf{A}$ can all be found by an algorithm aiming to find the strongly connected components in matrix $\mathbf{P^T}$, such as Tarjan's algorithm. Another method is to first find a second eigenvector of $\mathbf{A}$. The entries with nonzero values in that eigenvector must correspond to a node in an irreducible closed subset of the graph. To detect the different irreducible closed subsets one can apply Tarjan's algorithm, but only to the nodes that correspond to nonzero values in the second eigenvector.

There are several ways to reduce the effectiveness of the type of link spamming that we considered in this paper. One way is to reduce the chance of teleporting to a node in an irreducible closed subset. This can be done by using a non-homogeneous teleportation vector $\mathbf{v}$, called personalization vector. Using a personalization vector, the transition matrix becomes $\mathbf{A} = p\mathbf{P^T} + (1-p)\mathbf{v}\mathbf{e^T}$. Although the original idea of the personalization vector [10] was to more accurately describes the surfing behavior of certain types of web surfers, this vector can also be used to combat link spamming, by giving small values to entries of $\mathbf{v}$ that corresponds to nodes that are suspected of being link spammed. Note that Theorem 4.1, which tells us that $\mathbf{e^T}\mathbf{x^{(2)}} = 0$, still holds after introducing a personalization vector. Therefore, our findings carry over to this case.

We used the second eigenvector for detecting link spamming that is based on irreducible closed subsets. However, this is not the only link spamming technique, and other techniques will require different approaches to combat them. See for a discussion for example [2, 3].

References

[1] http://www.mathworks.nl/help/bioinfo/ref/graphconncomp.html. The Mathworks, R2013b Documentation, 2013.

[2] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside PageRank. *ACM Trans. Internet Technol.*, 5(1):92–128, February 2005.

[3] Monica Bianchini, Marco Marco Gori, and Franco Scarselli. PageRank: A Circuital Analysis. In *In Proceedings of the Eleventh International World Wide Web (WWW) Conference*, 2002.

[4] Timothy A. Davis and Yifan Hu. The University of Florida Sparse Matrix Collection. *ACM Trans. Math. Softw.*, 38(1):1:1–1:25, December 2011.

[5] L. Eldén. A Note on the Eigenvalues of the Google Matrix. *ArXiv Mathematics e-prints*, January 2004.

[6] Taher Haveliwala and Sepandar Kamvar. The Second Eigenvalue of the Google Matrix. Technical Report 2003-20, Stanford InfoLab, 2003.

[7] Dean L. Isaacson and Richard W. Madsen. *Markov chains, theory and applications.* Wiley New York , 1976.

[8] Carl D. Meyer, editor. *Matrix Analysis and Applied Linear Algebra.* Chapter 7 and 8. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[9] Cleve Moler. Experiments with MATLAB. Chapter 7: Google PageRank, The MathWorks, 2011.

[10] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[11] Robert Endre Tarjan. Depth-First Search and Linear Graph Algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.

[12] Martin B. van Gijzen and Peter Sonneveld. Algorithm 913: An elegant IDR(s) variant that efficiently exploits bi-orthogonality properties. *ACM Transactions on Mathematical Software*, 38(1):5:1–5:19, November 2011.