Mandatory Assignment 4
MATINF4130

Ivar Stangeby

November 9, 2017

1 Introduction

In this assignment we take a look at the PageRank algorithm, that helped establish Google as a powerful search engine. The algorithm assigns to each web page its "popularity" in a way that mimics how a human would define a popular web page. Before discussing the algorithm itself, we start with a mathematical intermezzo.

2 Mathematical framework

We first establish some notation. Let $\mathcal{S}$ denote the unit simplex

$$\mathcal{S} := \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid x_i \geq 0 \text{ for } i = 1, \ldots, n \text{ and } \sum_{i=1}^{n} x_i = 1 \right\}. \tag{1}$$

For later, we note that this is a closed and bounded set, which in $\mathbb{R}^n$ is equivalent to compact. We let $\boldsymbol{A}$ be a real $n \times n$ matrix with non-negative elements $a_{ij} \geq 0$, whose columns sum to one, and refer to this as a *stochastic matrix.* The image of $\mathcal{S}$ under $\boldsymbol{A}$ is denoted

$$\boldsymbol{A}(\mathcal{S}) := \{\boldsymbol{A}\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{S}\}. \tag{2}$$

a) If $\boldsymbol{y} \in \boldsymbol{A}(\mathcal{S})$ then $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$ for some $\boldsymbol{x}$. Note that since both x_i and a_{ij} are non-negative, we must have y_i non-negative for $i = 1, \ldots, n$. The sum

$$\sum_{i=1}^{n} y_i = \sum_{j=1}^{n} x_j \Big(\sum_{i=1}^{n} a_{ij} \Big) = \sum_{j=1}^{n} x_j = 1 \tag{3}$$

tells us that $\boldsymbol{y} \in \mathcal{S}$ and consequently $\boldsymbol{A}(\mathcal{S}) \subseteq \mathcal{S}$.

b) Considering $\boldsymbol{A} \colon \mathcal{S} \to \mathcal{S}$ as a linear operator, it suffices to show that it is bounded to show continuity. We have that

$$\|\boldsymbol{A}\|_1 = \max_{\|x\|=1} \|\boldsymbol{A}\boldsymbol{x}\|_1 = 1 \tag{4}$$

so $\boldsymbol{A}$ is bounded, and therefore also continuous in the $\|\cdot\|_1$ norm.

c) Assume that $(\lambda, \boldsymbol{v})$ is an eigenpair for $\boldsymbol{A}$. Since $\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v} \in \mathcal{S}$, we must have $|\lambda| \leq 1$. Since $\mathcal{S}$ is closed and bounded it is compact, and since $\mathcal{S}$ is continuous, it follows by Brouwer's fixed-point theorem that there exists a $\boldsymbol{w}$ such that

$$\boldsymbol{A}\boldsymbol{w} = \boldsymbol{w}. \tag{5}$$

Consequently, $(1, \boldsymbol{w})$ is a right eigenpair for $\boldsymbol{A}$.

From now on, we assume that the matrix entries a_{ij} are all strictly positive. Denote by $\mathcal{S}^\star$ the *interior* of $\mathcal{S}$:

$$\mathcal{S}^\star := \{\boldsymbol{x} \in \mathcal{S} \mid x_i > 0 \text{ for } i = 1, \ldots, n\}. \tag{6}$$

d) Let $\boldsymbol{x} \in \mathcal{S}$ and set $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$. Since at least one of the elements x_i are non-negative, and all a_{ij} are strictly positive, we have $y_i > 0$ for all $i = 1, \ldots, n$. This means that $\boldsymbol{y} \in \mathcal{S}^\star$, so $\boldsymbol{A}$ maps $\mathcal{S}$ to its interior.

e) Let $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{S}$ be two distinct vectors. Since the components of $\boldsymbol{x}$ and the components of $\boldsymbol{y}$ sum to one, we have that the components of $\boldsymbol{z} := \boldsymbol{x} - \boldsymbol{y}$ sum to zero. This means that since $\boldsymbol{x}$ and $\boldsymbol{y}$ are different, $\boldsymbol{z}$ is non-zero, hence $z_j < 0$ for at least one j. We will need this fact to achieve a strict inequality. We have that

$$\begin{aligned} \|\boldsymbol{Ax} - \boldsymbol{Ay}\|_1 = \|\boldsymbol{Az}\|_1 &= \sum_{i=1}^{n} |\sum_{j=1}^{n} a_{ij} z_j| \\ &< \sum_{j=1}^{n} \sum_{i=1}^{n} a_{ij} |z_j| \\ &= \sum_{j=1}^{n} |z_j| = \|\boldsymbol{z}\|_1 = \|\boldsymbol{x} - \boldsymbol{y}\|_1 \end{aligned} \tag{7}$$

Consequently, $\boldsymbol{A}\colon \mathcal{S} \to \mathcal{S}$ is a *contraction* in the $\|\cdot\|_1$ norm. Assume that $\boldsymbol{w}_1 \neq \boldsymbol{w}_2$ are two distinct eigenvectors with eigenvalue one. Then

$$\|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_1 = \|\boldsymbol{Aw}_1 - \boldsymbol{Aw}_2\|_1 < \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|_1, \tag{8}$$

which is a contradiction. We can therefore conclude that the geometric multiplicity $g(\lambda)$ of the eigenvalue $\lambda = 1$ is one.

3 The PageRank Algorithm

3.1 Motivation

Any web page of has a set of *forward links* which are links *to* other pages, and a set of *backlinks* which are links *from* other pages. Intuitively, pages with a large amount of backlinks should be deemed as more popular than those with few backlinks. There is however one flaw with this interpretation, and that is if the set of backlinks all come from obscure, rarely visited websites, the real popularity of the page is certainly lower. However, if a page has a few backlinks, but these come from large popular websites, they should be qualify as more important. The PageRank algorithm attempts to formalize this.

3.2 Notation

In the following, we will assume we have n websites, and that the *rank* or *popularity* of page j is denoted p_j. We store the page ranks in a popularity vector $\boldsymbol{p} :=$

$[p_1, \dots, p_n]^T$ which we assume to be normalized since we are only interested in the relative popularity. The total number of forward links from page j is denoted ℓ_j and the set of pages with forward links to page i is denoted

$$B_i \coloneqq \{j \mid \text{ there is a link from page } j \text{ to page } i\}\,. \tag{9}$$

We let $\boldsymbol{B}$ be the matrix with entries b_{ij} defined by

$$b_{ij} \coloneqq \begin{cases} \frac{1}{\ell_j} & \text{if there is a link from page } j \text{ to page } i, \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

3.3 The Algorithm

The idea is that the popularity of page i satisfies the following relation:

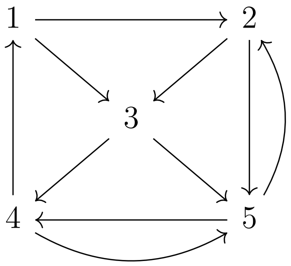
$$p_i = \sum_{j \in B_i} \frac{1}{\ell_j} p_j. \tag{11}$$

That is, every page distributes its popularity among its forward links. We are interested in finding the popularity vector $\boldsymbol{p}$. Notice that this relation is equivalent to the following matrix equation being satisfied

$$\boldsymbol{B}\boldsymbol{p} = \boldsymbol{p}, \tag{12}$$

hence we are looking for an eigenvector $\boldsymbol{p}$ for $\boldsymbol{B}$ with eigenvalue one.

a) An initial question is, when does such an eigenvector exist? Based on the work we did in earlier, we know that such an element exist if $\boldsymbol{B}$ turns out to be stochastic. The matrix $\boldsymbol{B}$ is stochastic precicely when each page has atleast one outgoing link, as this ensures that none of the columns of $\boldsymbol{B}$ are zero, and then each column sum to one by construction. As an example, the set of pages shown here



(13)

is associated to a stochastic matrix $\boldsymbol{B}$ given by

$$\boldsymbol{B} = \frac{1}{2}\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{14}$$

and solving the equation $\boldsymbol{Bp} = \boldsymbol{p}$ yields a popularity vector of

$$\boldsymbol{p} \approx \tag{15}$$

b) We want to formulate an iteration scheme for the approximate solution to the equation $\boldsymbol{Bp} = \boldsymbol{p}$. We may start with an arbitrary vector $\boldsymbol{p}^0 \in \mathcal{S}$, and define $\boldsymbol{p}^{k+1} := \boldsymbol{Bp}^k$ for $k = 0, 1, 2, \ldots$, however we are not guaranteed that $\boldsymbol{B}$ is stochastic. We therefore introduce the modified matrix $\boldsymbol{X}$ defined by

$$\boldsymbol{X} := \boldsymbol{B} + \frac{1}{n}\boldsymbol{C} \tag{16}$$

where $\boldsymbol{C}$ is an averaging matrix that fills out any zero columns of $\boldsymbol{B}$ so that the columns sum to one. That is,

$$c_{ij} := \begin{cases} 1, & \boldsymbol{B}_{:,j} = \boldsymbol{0}, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

This ensures that the matrix $\boldsymbol{X}$ is stochastic. In order to make sure that the iteration converges, we introduce another modified matrix $\boldsymbol{A}$

$$\boldsymbol{A} := \alpha\boldsymbol{X} + (1-\alpha)\frac{1}{n}\boldsymbol{1}, \tag{18}$$

where $\alpha \in (0, 1)$ and $\boldsymbol{1}$ is an $n \times n$ matrix filled with ones. This modification ensures that $\boldsymbol{A}$ is a contraction (cf. exercise Item e on page 2). The sequence $\left\{\boldsymbol{p}^k\right\}_{k=1}^{\infty}$ with $\boldsymbol{p}^0 \in \mathcal{S}$ and

$$\boldsymbol{p}^{k+1} = \boldsymbol{Ap}^k \tag{19}$$

therefore converges to a unique $\boldsymbol{p}$ that solves $\boldsymbol{Bp} = \boldsymbol{p}$. Writing out the iteration we have

$$\boldsymbol{p}^{k+1} = \alpha\boldsymbol{Bp}^k + \frac{\alpha}{n}\boldsymbol{Cp}^k + \frac{1-\alpha}{n}\boldsymbol{1p}^k \tag{20}$$

c) The matrix products $\boldsymbol{C}\boldsymbol{p}^k$ and $\mathbf{1}\boldsymbol{p}^k$ are large linear systems, and it is worthwile to spend some time considering them. Since $\boldsymbol{B}$ is known, we know exactly which columns of $\boldsymbol{B}$ are zero. Assume there are k zero-columns in $\boldsymbol{B}$ with indices of these columns as $i_1, \dots, i_k$. We may then set

$$\boldsymbol{C}\boldsymbol{p}^k(j) = p_{i_1} + \dots + p_{i_k} \tag{21}$$

for $j = 1, \dots, n$. Similarly, we may directly compute the elements of $\mathbf{1}\boldsymbol{p}^k$ as

$$\mathbf{1}\boldsymbol{p}^k(j) = p_1 + \dots + p_n \tag{22}$$

for all j.

d) We test the PageRank algorithm on the supplied set of data. For each iteration we compute the residual error, to see whether the method converges or not. Figure 1 shows the residual error plotted against the iterations. We used the PageRank algorithm to determine the ten most popular webpages, and these are displayed in Table 1.

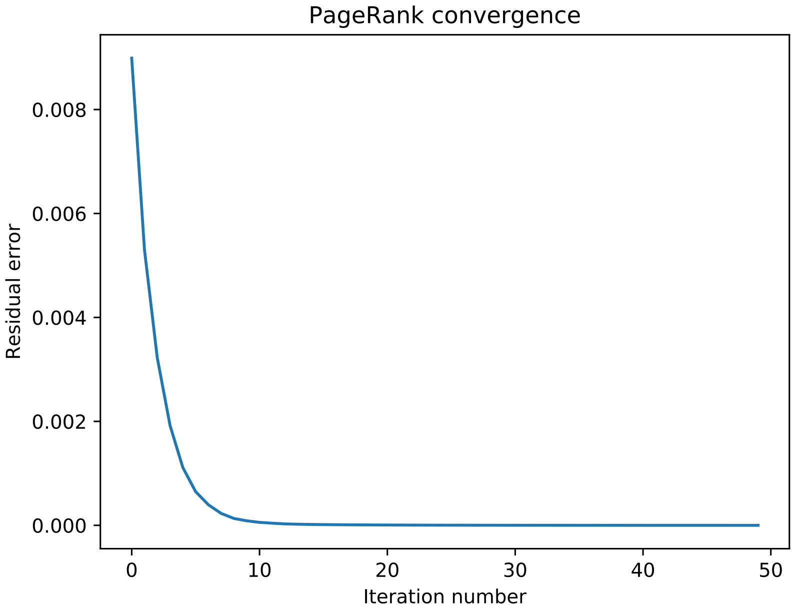


Figure 1: The residual error $E_{k+1} := \left\|\boldsymbol{p}^{k+1} - \boldsymbol{p}^k\right\|$. We see that the method has stabilized after approximately 20 iterations.

Table 1: The ten most popular webpages computed by the PageRank algorithm, with 50 iterations.

Position	ID	Popularity[%]
1	6982	0.103
2	2130	0.0807
3	1752	0.0756
4	8096	0.0706
5	8565	0.0702
6	3735	0.0635
7	1578	0.0585
8	8268	0.0579
9	6917	0.0572
10	6553	0.0569