Computing PageRank

IN3200/IN4200 Partial Exam, Spring 2019

Note: Each student should independently program the required code and write her/his own short report. These are to be submitted at Devilry within the announced deadline. The details about the submission can be found at the end of this document.

1 Introduction

An important "ingredient" of Google's search engine is the **PageRank** algorithm, which computes a numerical score for each webpage (inside a group of webpages). The score of a webpage is supposed to reflect its "importance", which depends on both the number of webpages that link to it and the "importance" of these inbound webpages.

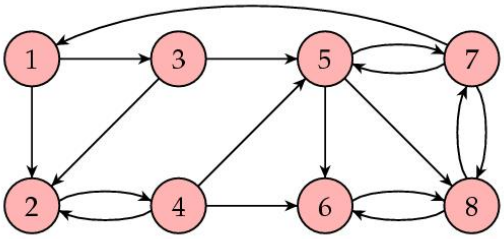

Figure 1: A very simple graphical representation example of eight webpages that are linked to each other.

Let us look at a concrete example of eight linked webpages as depicted in Figure 1, also called a **web graph**. For webpage No. 1, it has two outbound links (to webpages No. 2 and No. 3) and one inbound link (from webpage No. 7). For webpage No. 2, it has one outbound link (to webpage No. 4) and three inbound links (from webpages No. 1, No. 3 and No. 4). Similarly, the inbound and outbound links for the other six webpages can be identified in Figure 1. It turns out that webpage No. 8 has the highest score of PageRank, because it has inbound links from "important" webpages No. 5, No. 6 and No. 7, which have several inbound links of their own. (The details about how to compute the PageRank scores will be shown in the next section.)

2 The PageRank algorithm

Before diving into the numerical scheme of the PageRank algorithm, a few words of "warning" are in order. There are numerous online articles discussing the mathematical foundation of the PageRank algorithm, but you absolutely don't have to understand the mathematical details in order to implement this numerically practical algorithm. In the following, we will show a minimum numerical "skeleton" of the PageRank algorithm. If you want to understand more, a good reference can be the lecture notes authored by K. Shum [1].

2.1 The hyperlink matrix

For a group of linked webpages, with the total number of webpages being N, it is possible to represent the linkage connection between them by a so-called hyperlink matrix $\mathbf{A}$ of dimension $N \times N$. For $1 \leq i \leq N$, row i of $\mathbf{A}$ records all the inbound links towards webpage No. i. More specifically,

$$a_{ij} = \begin{cases} \frac{1}{L(j)} & \text{if there's an inbound link from webpage No. } j, \ (i \neq j) \\ 0 & \text{otherwise,} \end{cases}$$

where $L(j)$ denotes the number of outbound links from webpage No. j.

For example, the hyperlink matrix for the eight webpages shown in Figure 1 will be as follows:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 1 & \frac{1}{3} & 0 \end{bmatrix}$$

We remark that the main diagonal of a hyperlink matrix $\mathbf{A}$ contains only zeros. (Self-linkage should not affect the PageRank score.) The values of each column of $\mathbf{A}$ either sum up to 1, or are all zeros. The latter case corresponds to a webpage that has no outbound links, which is also called a **dangling webpage**. Another important remark is that $\mathbf{A}$ is a sparse matrix, where most of its values are zero (especially when the number of webpages N is large).

2.2 The iterative procedure

The objective of the PageRank algorithm is to find a vector of numerical values

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

where x_i is the true PageRank score of webpage No. i.

The algorithm is an iterative procedure that starts with an initial guess

$$\mathbf{x}^0 = \frac{1}{N} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

and the following formula computes $\mathbf{x}^k$ based on $\mathbf{x}^{k-1}$:

$$\mathbf{x}^k = \frac{\left(1 - d + d \cdot W^{k-1}\right)}{N} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} + d \cdot \mathbf{A}\mathbf{x}^{k-1} \tag{1}$$

Here, d is a prescribed scalar constant between 0 and 1, which is called the **damping constant**. (A typical choice of d is 0.85 in practice.) The term of $\mathbf{A}\mathbf{x}^{k-1}$ represents a sparse matrix-vector multiplication. The scalar value W^{k-1} is the summation of the PageRank scores, from iteration $k-1$, of all the dangling webpages, that is,

$$W^{k-1} = \sum_{m \in \mathcal{D}} x_m^{k-1} \quad \mathcal{D}\text{: set of indices for all the dangling webpages}$$

Note: if there is no dangling webpage, then the value of W^{k-1} is simply zero.

2.3 Stopping criterion

The iterations should be stopped if the difference between vectors $\mathbf{x}^{k-1}$ and $\mathbf{x}^k$ is small enough (that is, the iterations have converged). Then, the vector $\mathbf{x}^k$ is considered a good enough numerical approximation of the vector of true PageRank scores: $\mathbf{x}$.

The following is one possible way of testing that the difference between $\mathbf{x}^{k-1}$ and $\mathbf{x}^k$ is small enough:

$$\max_{1 \le j \le N} \left| x_j^k - x_j^{k-1} \right| < \varepsilon \tag{2}$$

where ε is a prescribed convergence threshold value (typically very small).

3 Requirements of the partial exam

3.1 Three functions

The following three functions, with suitable input and out arguments, should be implemented:

1. `read_graph_from_file` should read from a text file (filename prescribed as input) that contains the web graph, that is, the linkage connection information of a set of webpages. This function should set up the corresponding hyperlink matrix (as a sparse matrix in the CRS format, see Section 3.6 of the textbook), and identify the indices of all the dangling webpages (if any).

2. `PageRank_iterations` should implement the iterative procedure of the PageRank algorithm. The values of the damping constant d and the convergence threshold value ε should be among the input arguments. The converged PageRank score vector `x` should be among the output arguments.

3. `top_n_webpages` should go through the converged PageRank score vector `x` and list the top n webpages, with both their scores and page indices. The integer value of n should be among the input arguments to the function.

3.2 OpenMP parallelization

The `PageRank_iterations` function must be parallelized using OpenMP. IN4200 students must also parallelize the `top_n_webpages` function (optional for IN3200 students). The `read_graph_from_file` function does not need to be parallelized.

3.3 File format of a web graph

It can be assumed that a text file containing a web graph (that is, the webpage linkage connection information) has the following format:

- The first two lines both start with the `#` symbol and contain free text (listing the name of the data file, authors etc.);

- Line 3 is of the form "`# Nodes: integer1 Edges: integer2`", where `integer1` is the total number of webpages, and `integer2` is the total number of links. (Here, nodes mean the same as webpages, and edges mean the same as links.)

- Line 4 is of the form "`# FromNodeId ToNodeId`";

- The remaining part of the file consists of a number of lines, the total number equals the number of links. Each line simply contains two integers: the index of the outbound webpage and the index of the inbound webpage;

- Some of the links can be self-links (same outbound as inbound), these should be excluded when creating the hyperlink matrix;
- Note: the webpage indices start from 0 (C convention).

An example web graph file that contains the linkage connection information shown in Figure 1 is as follows:

```
# Directed graph (each unordered pair of nodes is saved once): 8-webpages.txt
# Just an example
# Nodes: 8 Edges: 17
# FromNodeId    ToNodeId
0       1
0       2
1       3
2       4
2       1
3       4
3       5
3       1
4       6
4       7
4       5
5       7
6       0
6       4
6       7
7       5
7       6
```

An example real-world web graph file (rather large) can be downloaded from `https://snap.stanford.edu/data/web-NotreDame.html`.

3.4 Submission

Each student should submit, via Devilry, a tarball (`.tar`) or a zip file (`.zip`). Upon unpacking/unzipping it should produce a folder named `IN3200_PE_xxx` or `IN4200_PE_xxx`, where `xxx` should be the UiO username of the student. Inside the folder, there should be *at least* the following files:

- `PE_functions_xxx.c` should contain the implementation of the three functions, one (or two) of those parallelized with OpenMP;
- `PE_main_xxx.c` should be a main program that accepts, on the command line, the filename of the webpage linkage information, the damping constant d, the convergence threshold value ε, and the n value for showing the top n webpages (see above). The main program should call the three functions implemented in `PE_functions_xxx.c`. Proper comments and output info (using `printf`) should be provided. Time measurement commands around all the three functions should also be included.
- `README.txt` should contain the most essential information about compilation and an example of how to run the main program.
- A short report (named `PE_report_xxx.pdf`) that describes the most important programming info (such as the main data structure and the basic

content of the three functions). Time measurements of the `PageRank_iterations` function, when varying the number of OpenMP threads, should be presented as well as the information of the compiler and hardware used. Programming issues relevant for performance optimization (if any) should also be included. IN4200 students should also provide a short discussion about the maximumly achievable (ideal) computing speed of the `PageRank_iterations` function, as well as the factors that prevent reaching the ideal speed in reality.

3.5 Grading

The grade of the submission will constitute 20% of the final grade of IN3200/IN4200. Grading of the submission will be based on the correctness, readability and speed of the implementations, in addition to the quality of the short report.

References

[1] Kenneth Shum. *Notes on PageRank Algorithm.* (Lecture notes of ENGG2012B Advanced Engineering Mathematics, The Chinese University of Hong Kong). 2013.
`http://home.ie.cuhk.edu.hk/~wkshum/papers/pagerank.pdf`