Università
della
Svizzera
italiana

Institute of Computational Science ICS

Numerical Computing **2020**

Student: Claudio Maggioni Discussed with: –

Solution for Project 1

Due date: Thursday, 8 October 2020, 12:00 AM

Numerical Computing 2020 — Submission Instructions
(Please, notice that following instructions are mandatory: submissions that don't comply with, won't be considered)

- Assignments must be submitted to iCorsi (i.e. in electronic format).
- Provide both executable package and sources (e.g. C/C++ files, Matlab). If you are using libraries, please add them in the file. Sources must be organized in directories called:
  Project_number_lastname_firstname
  and the file must be called:
  project_number_lastname_firstname.zip
  project_number_lastname_firstname.pdf
- The TAs will grade your project by reviewing your project write-up, and looking at the implementation you attempted, and benchmarking your code's performance.
- You are allowed to discuss all questions with anyone you like; however: (i) your submission must list anyone you discussed problems with and (ii) you must write up your submission independently.

The purpose of this assignment[1] is to learn the importance of numerical linear algebra algorithms to solve fundamental linear algebra problems that occur in search engines.

1. Page-Rank Algorithm

1.1. Theory [20 points]

1.1.1. What assumptions should be made to guarantee convergence of the power method?

The first assumption to make is that the biggest eigenvalue in terms of absolute values should (let's name it λ_1) be strictly greater than all other eigenvectors, so:

$$|\lambda_1| < |\Lambda_i| \forall i \in \{2..n\}$$

Also, the eigenvector *guess* from which the power iteration starts must have a component in the direction of x_i, the eigenvector for the eigenvalue λ_1 from before.

[1] This document is originally based on a SIAM book chapter from *Numerical Computing with Matlab* from Clever B. Moler.

1.1.2. What is a shift and invert approach?

The shift and invert approach is a variant of the power method that may significantly increase the rate of convergence where some application of the vanilla method require large numbers of iterations. This improvement is achieved by taking the input matrix A and deriving a matrix B defined as:

$$B = (A - \alpha I)^{-1}$$

where α is an arbitrary constant that must be chosen wisely in order to increase the rate of convergence. Since the eigenvalues u_i of B can be derived from the eigenvalues λ_i of A, namely:

$$u_i = \frac{1}{\lambda_i - \alpha}$$

the rate of convergence of the power method on B is:

$$\left|\frac{u_2}{u_1}\right| = \left|\frac{\frac{1}{\lambda_2 - \alpha}}{\frac{1}{\lambda_1 - \alpha}}\right| = \left|\frac{\lambda_1 - \alpha}{\lambda_2 - \alpha}\right|$$

By choosing α close to λ_1, the convergence is sped up. To further increase the rate of convergence (up to a cubic rate), a new α, and thus a new B, may be chosen for every iteration.

1.1.3. What is the difference in cost of a single iteration of the power method, compared to the inverse iteration?

Inverse iteration is generally more expensive than a regular application of the power method, due to the overhead caused by the intermediate matrix B. One must either recompute B every time α changes, which is rather expensive due to the inverse operation in the definition of B, or one must solve the matrix equation $(A - \alpha I)v_k = v_{k-1}$ in every iteration.

1.1.4. What is a Rayleigh quotient and how can it be used for eigenvalue computations?

The Railegh quotient is an effective way to either compute the corresponding eigenvalue of an eigenvector or the corresponding eigenvalue approximation of an eigenvector approximation. I.e., if x is an eigenvector, then:

$$\lambda = \mu(x) = \frac{x^T A x}{x^T x}$$

is the corresponding eigenvalue, while if x is an eigenvector approximation, for example found through some iterations of the power method, then λ is the closest possible approximation to the corresponding eigenvalue in a least-square sense.

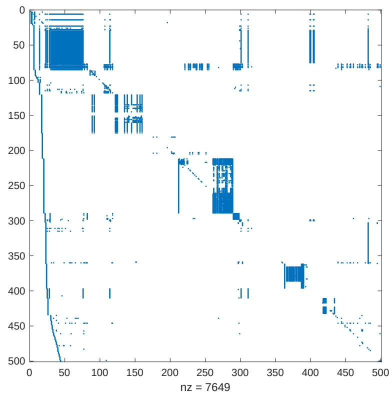
1.2. Other webgraphs [10 points]

The provided PageRank MATLAB implementation was run 3 times on the starting websites `http://atelier.inf.usi.ch/ maggicl`, `https://www.iisbadoni.edu.it`, and `https://www.usi.ch`, with results listed respectively in Figure **??**, Figure **??** and Figure **??**.

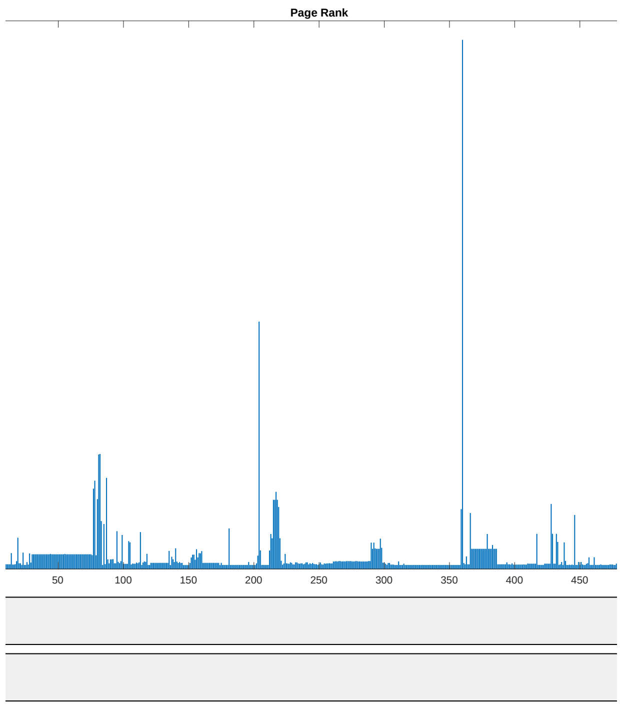
One patten that emerges on the first and third execution is the presence of 1s in the main diagonal. This indicates that several pages found have a link to themselves.

Another interesting pattern, this time observable in all executions, is the presence of contiguous rectangular regions filled with 1s, especially along the main diagonal. This may be due to the presence of pages belonging to the same website, thus having a common layout and perhaps linking to a common set of internal (when near to the main diagonal) or external pages.

Finally, we can always observe a line starting from the top-left of G and ending in its bottom-left, running along a steep path slighly going right. This may be a side effect of the way pages are discovered and numbered: if new pages are continuously discovered, these pages will be added at the end of U and a corresponding vertical strip on 1s will appear in the bottomest non-colored region of G. This continues until n unique pages are visited and the line reaches the bottom edge of the connectivity matrix. The steepness of the line thus formed depends on the amount of new pages discovered in each of the first iterations of the `surfer(...)` function.



(a) Spy plot of connectivity matrix



(b) Page rank bar graph

360	0.0869	31	1	`https://creativecommons.org/licenses/by-sa/3.0`
204	0.0406	8	1	`https://forum.gitlab.com`
82	0.0189	117	18	`https://www.mediawiki.org`
81	0.0188	117	4	`https://wikimediafoundation.org`
87	0.0150	6	1	`https://docs.gitea.io`
78	0.0145	114	9	`https://www.mediawiki.org/wiki/Special:MyLanguage/How_to_contribute`
77	0.0132	77	13	`https://foundation.wikimedia.org/wiki/Privacy_policy`
217	0.0127	40	8	`https://bugs.archlinux.org`
80	0.0115	107	6	`https://foundation.wikimedia.org/wiki/Cookie_statement`
215	0.0114	38	5	`https://bbs.archlinux.org`

(c) Top 10 webpages with highest PageRank

Figure 1: Results of first PageRank calculation (for starting website `http://atelier.inf.usi.ch/ maggicl/`)

1.3. Connectivity matrix and subcliques [10 points]

The following ETH organization are following for the near cliques along the diagonal of the connectivity matrix in `eth500.mat`. The clique approximate position on the diagonal is indicated throgh the ranges in parenthesis.

- `baug.ethz.ch` (74-100)
- `mat.ethz.ch` (114-129)
- `mavt.ethz.ch` (164-182)
- `biol.ethz.ch` (198-216)
- `chab.ethz.ch` (221-236)
- `math.ethz.ch` (264-278)
- `erdw.ethz.ch` (321-337)
- `usys.ethz.ch` (358-373)
- `mtec.ethz.ch` (396-416)
- `gess.ethz.ch` (436-462)

1.4. Connectivity matrix and disjoint subgraphs [10 points]

1.4.1. What is the connectivity matrix G (w.r.t figure 5)?

The connectivity matrix G, with U being defined as $\{"alpha", "beta", "gamma", "delta", "rho", "sigma"\}$ is:

$$G = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

1.4.2. What are the PageRanks if the hyperlink transition probability p is the default value 0.85?

First we compute the matrix A, finding:

$$A = \frac{1}{40} \begin{bmatrix} 1 & 1 & 1 & 35 & 1 & 1 \\ 18 & 1 & 1 & 1 & 1 & 1 \\ 18 & 18 & 1 & 1 & 1 & 1 \\ 1 & 18 & 35 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 35 \\ 1 & 1 & 1 & 1 & 35 & 1 \end{bmatrix}$$

We then find the eigenvectors and eigenvalues of A through MATLAB, finding that the solution of $Ax = 1x$ is:

$$x \approx \begin{bmatrix} 0.4771 \\ 0.2630 \\ 0.3747 \\ 0.4905 \\ 0.4013 \\ 0.4013 \end{bmatrix}$$

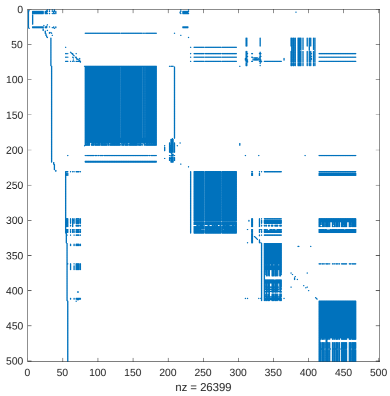
Thus the pageranks are the components of vector x, w.r.t. the order given in U.

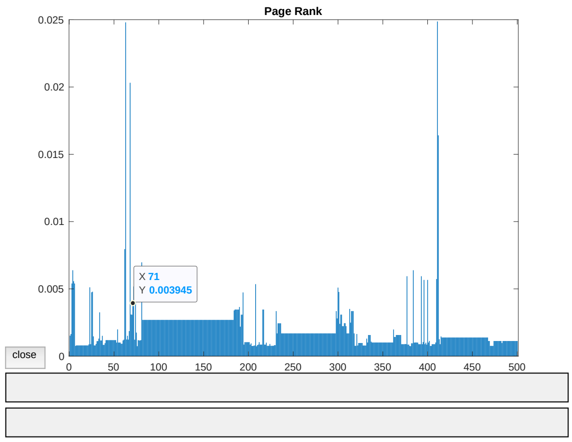
1.4.3. Describe what happens with this example to both the definition of PageRank and the computation done by pagerank in the limit $p \to 1$.

If p is closer to 1, then the probability a web user will visit a certain page randomly decreases, thus giving more weight in the computation of PageRank to the links between one page and another.

In the computation, increasing p decreases δ (which represents the probability of a user randomly visiting a page), eventually making it 0 when p is 1.

1.5. PageRanks by solving a sparse linear system [50 points]





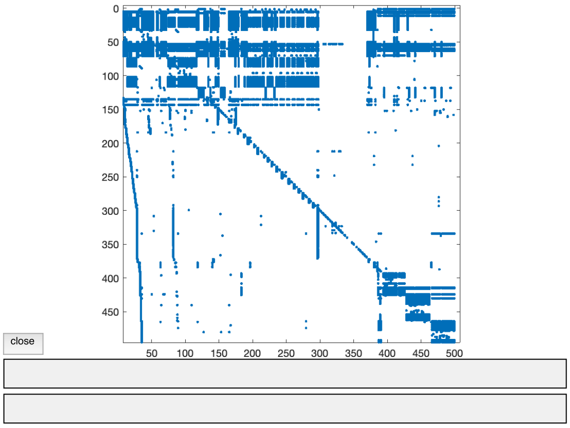
(a) Spy plot of connectivity matrix

(b) Page rank bar graph

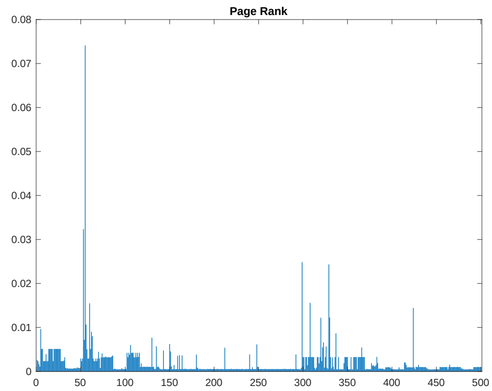
411	0.0249	42	1	https://twitter.com/mozilla
63	0.0248	145	1	https://twitter.com/firefox
68	0.0203	142	1	https://www.instagram.com/firefox
412	0.0164	37	1	https://www.instagram.com/mozilla
62	0.0080	21	1	https://github.com/mozilla/kitsune
81	0.0070	110	2	https://www.apple.com
384	0.0064	5	1	https://www.xfinity.com/privacy/policy/dns
4	0.0064	32	0	https:
377	0.0059	19	1	https://abouthome-snippets-service.readthedocs.io/en/latest/data_collection.html
393	0.0059	19	1	https://www.adjust.com/terms/privacy-policy
410	0.0057	16	1	https://wiki.mozilla.org/Firefox/Data_Collection

(c) Top 10 webpages with highest PageRank

Figure 2: Results of second PageRank calculation (for starting website https://www.iisbadoni.edu.it/)



(a) Spy plot of connectivity matrix



(b) Page rank bar graph

55	0.0741	354	1	`https://www.instagram.com/usiuniversity`
53	0.0324	366	3	`https://www.facebook.com/usiuniversity`
299	0.0248	6	1	`https://twitter.com/usi_en`
329	0.0243	8	1	`https://www.facebook.com/USIeLab`
308	0.0156	7	3	`https://www.facebook.com/USIFinancialCommunication`
60	0.0155	316	2	`https://www.swissuniversities.ch`
424	0.0144	96	1	`https://it.bul.sbu.usi.ch`
330	0.0123	6	4	`https://www.facebook.com/USI.ITDxC`
320	0.0122	7	1	`https://www.facebook.com/usiimeg`
56	0.0107	320	0	`https://www.youtube.com/usiuniversity`

(c) Top 10 webpages with highest PageRank

Figure 3: Results of third PageRank calculation (for starting website `https://www.usi.ch/`)