PageRank algorithm

Due on January 30, 2021

*The report, illustrating the **Tasks 1, 2** and **3** described below, is due on January 23, 2021. Send the report and the python and/or FORTRAN codes to jose.lages@utinam.cnrs.fr. The use of LaTeX is mandatory. LaTeX packages like tikz and tikz-network can be used to draw graphs, diagrams, and networks. Results have to be commented in details and conclusions have to be drawn. Results will be presented by convenient, nice and meaningful figures. Figures can be produced directly within the LaTeX document using the tikz package, with python, with gnuplot... For each figure, send the code used to draw it. Do not forget to provide caption for each figure with exhaustive information. Generate vectorial figures (such as .pdf, .svg, .eps) which should be, in principle, lighter than heavy files such as .jpg, .png, ... Keep the report with a weight as low as possible. Any developments going further than the **Tasks 1, 2** and **3** are welcome.*

Networks are ubiquitous: social interactions, ie, persons related by acquaintances, can be modeled by social networks, the World Wide Web (WWW) is a collection of myriads of hyperlinked web-pages, billions of neurons communicate through electrical signals, ten thousand of genes interact with each other, economical sectors transform goods from other sectors and supplies goods to other sectors...

In many cases, (large amount of) data can be regarded as a network.[a] Eg, let us consider "Les Misérables" by V. Hugo, and let us consider the characters as the nodes of a network. We can think about many ways to relate each character to the others. A simple way is the following one: let us consider that if a character B is cited in the book just after a character A then a link $A - B$ is created between the two characters. Following this procedure for all the characters, we construct an undirected network. If we consider that A is cited before B, we can add this additional information by assigning a direction to the link which is now $A \to B$. Following this procedure for all the characters, we construct a directed network. If we count the number of times in the book that character B is cited just after A, it can serve to give a weight to the $A \to B$ link. Following this procedure for all the characters, we construct a directed and weighted network.

Once data is modeled by a network, a measure of centrality allows to determine how central or important is a node. For a social network, such a measure tells us which person is the most influential. For the WWW, it allows to rank the web-pages by relevance. For "Les Misérables", it tells which character plays the most central role in the story. Many measure of centrality exist. The most simple one is the degree centrality. The degree of a node is the number of links going to or going from this node. For degree centrality, the nodes of the network are ranked by their degree. According to this simple measure, the most (less) important node is the most (less) connected node. Note that the notion of importance is relative to the centrality measure we use. For two different centrality measures, the ranking of the nodes, and consequently their importance in the network, can be different.

Here, we will focus on the PageRank centrality which was at the heart of the Google search engine at its beginning in 1998 [1]. The associated algorithm is very powerful as it is able to rank billions of nodes in a reasonable number of computational steps. Let us consider a network with N nodes which are connected each other with ℓ links. Let us assign a number or a label to each node, eg, $1, 2, \dots, N$. The network structure is encoded in the adjacency matrix A the elements of which are

$$A_{ij} = \begin{cases} 1 & \text{if } j \text{ points to } i \\ 0 & \text{otherwise.} \end{cases} \qquad (1)$$

For an undirected network, the matrix A is symmetric, ie, $A_{ij} = A_{ji}$, $\forall (i,j)$. In their 1998 seminal paper [1], Brin and Page, co-founders of Google, described the PageRank algorithm without using the matrix formalism. They also omitted to talk about Markov chains [2], a class of stochastic models, introduced in 1906, to which the PageRank algorithm belongs. The adjacency matrix A is converted into a stochastic matrix S the elements of which are

$$S_{ij} = \begin{cases} A_{ij}/k_j^{\text{out}} & \text{if } k_j^{\text{out}} \neq 0 \\ 1/N & \text{otherwise.} \end{cases} \qquad (2)$$

Here, $k_j^{\text{out}} = \sum_{i=1}^{N} A_{ij}$ is the out-degree of the jth node. Otherwise stated, k_j^{out} is the number of ways to leave the node j. By extension, we can also define the in-degree of the ith node as $k_i^{\text{in}} = \sum_{j=1}^{N} A_{ij}$. The value k_i^{in} gives the number of ways to jump to the ith node. We easily observe that for all column j, $\sum_{i=1}^{N} S_{ij} = 1$, ie, each column of the stochastic

[a] In mathematics, a network is called a graph. A node of a network is a vertex of a graph, and a link between two nodes of a network is an edge between two vertices of a graph. It is just a renaming.

matrix S is normalized to unity. The stochastic matrix element S_{ij} is the transition probability from node j to node i. A random surfer, condemned to forever wander from node to node on the network, has a probability S_{ij} to jump to the node i if it is initially localized on the node j. If the number of ways out from the node j is $k_j^{\rm out}$, then the random surfer can jump from node j to any of the adjacent nodes with the same probability $S_{ij} = 1/k_j^{\rm out}$. If the node j is a dangling node, ie, there is no way out from node j, then one allows the random surfer to jump, with probability $1/N$, to any node of the network. Let us assume that the random surfer starts its journey from the node j_0. The vector $\mathbf{p}^{(1)} = S\mathbf{p}^{(0)}$, with $p_i^{(0)} = \delta_{ij_0}$, gives the probability distribution describing the localization of the random surfer after a jump, ie, an iteration of the stochastic process.[b] The ith component of the vector $\mathbf{p}^{(n)} = S^n\mathbf{p}^{(0)}$, ie, $p_i^{(n)}$, is the probability that the random surfer ends to the node i after n iterations.[c] The vector $\mathbf{p}^{(\infty)}$ gives the probability distribution after an infinite journey of the random surfer. This probability distribution is not unique as it depends on the initial probability distribution $\mathbf{p}^{(0)}$, ie, from where the random surfer starts its journey. In order to obtain an unique solution of the random walk on the network, we define the Perron-Frobenius operator G the elements of which are

$$G_{ij} = \alpha S_{ij} + (1-\alpha)\, v_i. \tag{3}$$

Here, the parameter $\alpha \in [0.5, 1[$ is the damping factor and $\mathbf{v}$ is a preferential vector. The equation (3) tells us that at each iteration, the random surfer follows, with the probability α, the structure of the network, or jump to any node, with the probability $(1-\alpha)$, according to the preferential vector $\mathbf{v}$. In the genuine article [1], the damping factor is taken as $\alpha = 0.85$ and the preferential vector components are $v_i = 1/N$, $\forall i$. The Perron-Frobenius operator G is called the Google matrix [3]. This damping factor allows the random surfer to be never stuck in a sub-network and, hence, only one solution exists $\mathbf{P}$ for the infinite journey probability distribution, $G^\infty\mathbf{p}^{(0)} = \mathbf{P}$ for all initial probability distribution $\mathbf{p}^{(0)}$. The vector $\mathbf{P}$ gives the steady state probability distribution of the stochastic process encoded by the stochastic matrix G since

$$G\mathbf{P} = \mathbf{P}. \tag{4}$$

The ith component of the PageRank vector $\mathbf{P}$ is proportional to the number of times the random surfer reach the node i. It is possible to define the PageRank index K. We assign the rank $K = 1$ to the node i_1 such as

$$P_{i_1} = \max_{i\in[1,\dots,N]} \{P_i\},$$

the rank $K = 2$ to the node i_2 such as

$$P_{i_2} = \max_{i\in[1,\dots,N]-\{i_1\}} \{P_i\},$$

the rank K to the node i_K such as

$$P_{i_K} = \max_{i\in[1,\dots,N]-\{i_1,\dots,i_K\}} \{P_i\},$$

and the rank $K = N$ to the node i_N such as

$$P_{i_N} = \min_{i\in[1,\dots,N]} \{P_i\}.$$

The nodes of the network are then ranked by descending order of their values of the PageRank probabilities. The node to which $K = 1$ ($K = N$) is assigned is the most (less) important or central node of the network according to the PageRank centrality measure.

Task 1 – PageRank computation: direct diagonalization

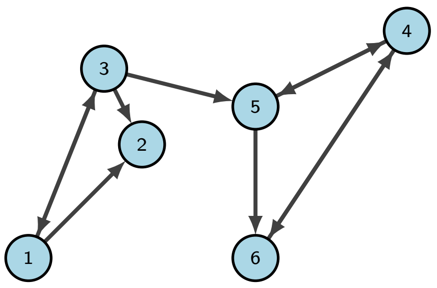

Figure 1: Network 1.

For the above displayed network,

[b]The Kronecker symbole δ_{ij} is equal to 1 if $i = j$ and is equal to 0 otherwise.

[c]Since the vector components of $\mathbf{p}^{(n)}$ are probabilities, we have $\left\|\mathbf{p}^{(n)}\right\|_1 = \sum_{i=1}^N p_i^{(n)} = 1$.

- rank the nodes according to the PageRank centrality by solving the eigenproblem (4). We take the original value $\alpha = 0.85$,
- what happens for the PageRank probabilities and the PageRank ranking when α varies in the interval $[0.5, 1[$,
- compare the PageRank centrality with the in-degree centrality,
- make relevant figures highlighting your results.

Equation (4) can be seen as an eigenproblem. The vector $\mathbf{P}$ is then the eigenvector of the matrix G associated to the eigenvalue 1. Hence, in principle for any network, the diagonalization of the matrix G provides the PageRank vector $\mathbf{P}$. But, the computational complexity for the matrix diagonalization is $O\left(N^2\right)$ which induces a CPU time cost unbearable for very large networks with $N \gg 1$. Taking as an example the hundreds of trillions of web-pages indexed by the Google search engine, the diagonalization is not an option. But, each web-page points to a relatively small number ($\sim 10-100$) of other web-pages. Hence, the WWW adjacency matrix is an extremely sparse matrix ($\ell \ll N^2$) as it is almost filled with only zeroes. Instead of working with matrices, it is convenient to work with lists of real positive numbers. There is no need to record and keep in memory the zeroes of the adjacency matrix A, only the list of the links is needed. In fact, one should be able to construct a PageRank algorithm with a computational complexity $O\left(nnz\left(S\right)\right) = O\left(\ell\right)$.[d] In the beginning of the 2000s, the low computational complexity of the PageRank algorithm gave the advantage of the Google search engine over the other web search engines. According to the Perron-Frobenius theorem, the leading eigenvalue of the matrix G is $\lambda_1 = 1$. The spectral radius of G is $\rho\left(G\right) = 1$, hence the other eigenvalues are $\{|\lambda_i| < 1\}_{i=2,\dots,N}$.[e] Let us consider an initial vector $\mathbf{p}^{(0)}$, *a priori*, not orthogonal to $\mathbf{P}$. Let us denote $\left\{\mathbf{P}^{(i)}\right\}_{i=2,\dots,N}$ the eigenvectors associated respectively to the non-leading eigenvalues $\{\lambda_i\}_{i=2,\dots,N}$. The initial vector can be written $\mathbf{p}^{(0)} = \sum_{i=1}^{N} \alpha_i \mathbf{P}^{(i)}$ where $\mathbf{P}^{(1)} = \mathbf{P}$ and $\{\alpha_i \in \mathbb{C}\}_{i=1,\dots,N}$. The infinite number of successive applications of G on $\mathbf{p}^{(0)}$ gives

$$\lim_{n\to\infty}\left(G^n \mathbf{p}^{(0)}\right) = \alpha_1 \mathbf{P} + \underbrace{\lim_{n\to\infty}\left(\sum_{i=2}^{N}\left(\alpha_i \lambda_i^n \mathbf{P}^{(i)}\right)\right)}_{\sim 0 \text{ as } |\lambda_i|<1, \forall i=2,\dots,N} = \alpha_1 \mathbf{P} \propto \mathbf{P}. \tag{5}$$

A simple power iteration method allows to compute efficiently the PageRank vector $\mathbf{P}$. Let us start with any norm-1-lized vector $\mathbf{p}^{(0)}$. The kth iteration of the power iteration method gives $\mathbf{p}^{(k)} = G\mathbf{p}^{(k-1)} / \left\|G\mathbf{p}^{(k-1)}\right\|_1$. The iterative process can be stopped at step m once $\mathbf{p}^{(m)} \sim \mathbf{p}^{(m-1)}$.

Task 2 – PageRank computation: power iteration method

For the network displayed in Figure 1, use the power iteration method to compute the PageRank vector and determine the PageRanking of the nodes. We still take the original value $\alpha = 0.85$.

Requirements:

- no matrices. Work only with the list of the links and the list of the dangling nodes,
- provide in the report a rigorous pseudo-code,
- carefully choose the convergence criterion,
- measure the CPU time needed to execute your code, compare this time with the one of **Task 1**.

The aim of the *Encyclopédie* [4], initially conceived by Diderot and d'Alembert, was to aggregate the whole available human knowledge and to make it accessible to all Citizens. The *Encyclopédie* was one of the most powerful catalyst of modern development of science and society. The knowledge transfer process have been considerably accelerated with the dawn of Wikipedia, the free online collaborative encyclopedia, which now supersedes the Encyclopedia Britannica in size and even in accuracy of articles devoted to many scientific domains [5]. Presently, Wikipedia contains more than three hundred linguistic editions representing different and complementary cultural points of view on human knowledge. This huge amount of encyclopedic data encodes information about how different cultures and societies are entangled. Nowadays, the efficient extraction of such a type of information is one of the great challenges of the Big Data era.

[d] Here, $nnz(S)$ stands for the number of non-zero elements of S.
[e] These eigenvalues are complex numbers.

Task 3 – PageRanking Wikipedia

Apply your code to Wikipedia editions. At http://perso.utinam.cnrs.fr/~lages/datasets/wiki/wiki/, you'll find the structure of 24 linguistic editions of Wikipedia collected in February 2013. These 24 editions are presented in Table 1. In each xxwiki folder, where xx denotes the edition, you'll find two zipped files. The xxwiki.titles gives the list of the titles of the articles in the xx edition. In this list, we assign the label i to a given title if it appears at the ith line. The xxwiki.txt gives the list of the directed (hyper)links between the articles. Each line corresponds to a link between two articles. A line "110 1058" means that the 110th article (in the xxwiki.titles list) points to the 1058th article. You should start with the small size editions such as the Thai, Greek, and Hindi ones. Maybe python code is too slow to handle bigger edition in a reasonable amount of time. If it is indeed the case, use a FORTRAN or C++ code instead. For the hugest editions, let your computer run at night, several hours of computation are needed. I can launch computations for you at the laboratory if you need (with your code of course).

Things to do:

- Determine and compare the top 10 articles according to the PageRank centrality for (as many as you can) different Wikipedia editions. I will test your code on few editions.
- Find few articles which are present in each of the editions you can consider. For example, the article devoted to Donald Trump or the one devoted to Philosophy. Compare the rank of these articles in the different editions.
- Make a plot of the CPU time vs the number of links ℓ. If you cannot handle the English edition, what is your estimate for the CPU time needed to apply the PageRank algorithm to this edition ?

Edition	Language	N	Edition	Language	N
EN	English	4212493	VI	Vietnamese	594089
DE	German	1532978	FA	Persian	295696
FR	French	1352825	HU	Hungarian	235212
NL	Dutch	1144615	KO	Korean	231959
IT	Italian	1017953	TR	Turkish	206311
ES	Spanish	974025	AR	Arabic	203328
RU	Russian	966284	MS	Malaysian	180886
PL	Polish	949153	DA	Danish	175228
JA	Japanese	852087	HE	Hebrew	144959
SV	Swedish	780872	HI	Hindi	96869
PT	Portuguese	758227	EL	Greek	82563
ZH	Chinese	663485	TH	Thai	78953

Table 1: Example of 24 Wikipedia editions collected in February 2013. Here N is the number of articles. These data are available at http://perso.utinam.cnrs.fr/~lages/datasets/wiki/

References

[1] S. Brin and L. Page, *The Anatomy of a Large-Scale Hypertextual Web Search Engine*. In: Seventh International World-Wide Web Conference (WWW 1998), April 14-18, 1998, Brisbane, Australia

[2] A.A. Markov, Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschie drug ot druga, Izvestiya Fiziko-matematicheskogo obschestva pri Kazanskom universitete, 2-ya seriya, 15, 135 (1906) (Extension of the limit theorems of probability theory to a sum of variables connected in a chain)

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

[4] *Encyclopédie, ou dictionnaire raisonné des sciences, des arts et des métiers*, etc., eds. Denis Diderot and Jean le Rond d'Alembert (1751)

[5] J. Giles, *Internet encyclopaedias go head to head: Jimmy Wales' Wikipedia comes close to Britannica in terms of the accuracy of its science entries*, Nature 438 (7070): 900-1 (2005)