

PageRank Convergence Proof

Course on “Algorithms for massive datasets”, University of Milan

Written by Jean-Baptiste Lepidi, revised by Dario Malchiodi

Released under CC-BY-NC-SA

1 Preliminary definitions

Consider a graph (V, E) , where V and $E \subseteq V \times V$ denote the set of nodes and edges, respectively. We will describe such a graph using a *transition matrix* defined as follows:

Definition 1 (Transition matrix). *The transition matrix of a graph (V, E) , with $V = \{v_1, \dots, v_n\}$, is the $n \times n$ matrix M having*

$$m_{ij} = \begin{cases} \frac{1}{o_j} & \text{if } (v_j, v_i) \in E, \\ 0 & \text{otherwise,} \end{cases}$$

as generic entry, where o_j denotes the out degree of the j -th node of the graph.

Example 1. *The graph depicted in Figure 1 is associated to the following transition matrix:*

$$M = \begin{pmatrix} A & B & C & D \\ 0 & 0 & 1 & 0 \\ \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ \frac{1}{3} & 1 & 0 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \end{pmatrix} \begin{matrix} A \\ B \\ C \\ D \end{matrix},$$

where each row and column of M has been labeled in order to highlight the starting and ending node (respectively, in columns and rows) in each edge of the graph.

Definition 2 (Row-wise and column-wise stochasticity). *A matrix A is row-wise stochastic (RWS for short) if and only if all its entries are non-negative and the sum of all the elements in any of its row equals 1. Analogously, A is column-wise stochastic (CWS for short) if its entries are non-null and the elements of any of its columns sum up to 1.*

By definition, the transition matrix of a graph is CWS, that is, all its entries are non-negative and the entries in any column sum up to 1. Therefore, each

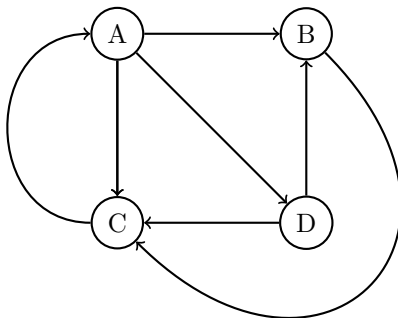


Figure 1: A simple graph.

column describes a probability distribution over V , and this probability distribution describes the destination of a random surfer in the conceptual process describing the PageRank computation.

We will also refer to some basic algebraic definitions and properties related to matrices, eigenvalues and eigenvectors.

Definition 3 (Determinant). *Let $A \in \mathbb{R}^{n \times n}$ be a square matrix. The determinant of A , denoted by $\det A$, is a scalar quantity associated with A such that $\det A = 0$ if and only if the rows (equivalently, the columns) of A are linearly dependent.*

For $n = 1$, if $A = (a_{11})$, we define $\det A = a_{11}$. For $n > 1$, the determinant can be computed recursively by Laplace expansion: fixing any row index $i \in \{1, \dots, n\}$,

$$\det A = \sum_{j=1}^n (-1)^{i+j} a_{ij} C(i, j),$$

where $C(i, j)$ denotes the (i, j) -minor of A , that is, the determinant of the submatrix obtained by deleting the i -th row and the j -th column of A . An analogous expansion holds by fixing a column index.

Definition 4 (Eigenvalues and eigenvectors). *Given a square matrix A of dimension n , a scalar λ and a vector $e \in \mathbb{R}^n$ are respectively called an eigenvalue and an eigenvector of the matrix if and only if $Ae = \lambda e$. For the sake of brevity, the pair (λ, e) is also called an eigenpair of A .*

Without diving into the mathematical details, we can state a property of the matrices which we will use in this document: when a matrix A of dimension n admits n distinct eigenpairs, and all the eigenvalues belong to $\mathbb{R}$, the corresponding eigenvectors form a basis for $\mathbb{R}^n$. Therefore, if $e_1, \dots, e_n$ are the involved eigenvectors, any $x \in \mathbb{R}^n$ can be expressed as the linear combination

$\sum_{i=1}^n \alpha_i e_i$, for a suitable choice of the weights in the sum. Finally, note that eigenvectors are strictly defined only by the direction they identify, and not by their magnitude: in other words, if (λ, e) and $(\lambda, \beta e)$ denote the same eigenpair for any $\beta \in \mathbb{R}$.

2 PageRank computation

The simplest possible computation of the PageRank index is described in Algorithm 1: basically, it consists in iterating the product of the transition matrix M of a graph with a vector, each time substituting the previous vector with the result of the multiplication. This is repeated until a suitable stopping condition occurs, e.g., the norm of the difference between two successive vectors is below a fixed threshold, or when a maximum number of iterations has been executed. Note that this algorithm can be interpreted as a *random surfing* process in which each element of a population of surfers, after having been assigned uniformly at random to one node of the graph (line 3), iterates the task of selecting uniformly at random one among the outgoing edges of the node it is sitting on, and following that edge. Indeed, considering the event occurring when a random surfer lies in a given node at a given time, by the total law of probabilities we have

$$\mathbb{P}(\text{in node } i \text{ at time } t+1) = \sum_{j=1}^n \mathbb{P}(\text{moving from } j \text{ to } i \mid \text{in node } i \text{ at time } t) \cdot \mathbb{P}(\text{in node } j \text{ at time } t) . \quad (1)$$

In the hypothesis that the graph under study does not change with time, and letting $v_i(t) := \mathbb{P}(\text{in node } i \text{ at time } t)$, Equation 1 reads $v_i(t+1) = (Mv(t))_i$, which is the component-wise version of the right-hand side of the assignment in line 5. Therefore, Algorithm 1 realizes the above mentioned random surfing process, stopping when the probabilities that surfers sit in a node converge to a limiting distribution.

Algorithm 1 PageRank computation algorithm.

Require: M (transition matrix)

- 1: $n \leftarrow \text{sizeof}(M)$
 - 2: $t \leftarrow 0$
 - 3: $v(t) \leftarrow (1/n, \dots, 1/n)^\top$
 - 4: **repeat**
 - 5: $v(t+1) \leftarrow Mv(t)$
 - 6: $t \leftarrow t+1$
 - 7: **until** convergence
-

3 Intermediate results

Theorem 1 (Power method). *Given a square matrix A , the succession of vectors $v(t)$ defined in Algorithm 1 converges to the principal eigenvector of A , defined as the eigenvector corresponding to the highest eigenvalue of the matrix.*

Proof. Without loss of generality, denote by $(\lambda_1, e_1), \dots, (\lambda_n, e_n)$ the eigenpairs of M , sorted by nonincreasing eigenvalue. As $e_1, \dots, e_n$ is a basis for $\mathbb{R}^n$, there exist $\alpha_1, \dots, \alpha_n \in \mathbb{R}$ such that

$$v(0) = \alpha_1 e_1 + \dots + \alpha_n e_n .$$

Recalling line 5 of the algorithm, and applying the distributive property of matrix-vector product followed by the definition of eigenpair we get that

$$\begin{aligned} v(1) = aV(0) &= A(\alpha_1 e_1 + \dots + \alpha_n e_n) \\ &= \alpha_1 A e_1 + \dots + \alpha_n A e_n \\ &= \alpha_1 \lambda_1 e_1 + \dots + \alpha_n \lambda_n e_n . \end{aligned}$$

By induction, it is easy to show that

$$\begin{aligned} v(t) &= \alpha_1 \lambda_1^t e_1 + \dots + \alpha_n \lambda_n^t e_n \\ &= \lambda_1^t \left(\alpha_1 e_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1} \right)^t e_2 + \dots + \alpha_n \left(\frac{\lambda_n}{\lambda_1} \right)^t e_n \right) . \end{aligned}$$

Having sorted the eigenvalues by non-decreasing order, it is obvious that $(\lambda_i/\lambda_1)^t \rightarrow 0$ for any $i > 1$. Thus

$$v(t) \stackrel{\text{as } t \text{ grows}}{\approx} \lambda_1^t \alpha_1 e_1 , \quad (2)$$

and the thesis follows, because eigenvectors are defined as directions. $\square$

Note that, in the previous proof, the last equation does not directly involves the limit of $v(t)$ for $t \rightarrow +\infty$, because the second hand depends on t : specifically, if $\lambda_1 > 1$ the limit would diverge, and if $\lambda_1 < 1$ the limit would converge to the null vector.

We will now prove some intermediate results which will be used in the following section to show that Algorithm 1 converges.

Lemma 1. *A square matrix A and its transpose have the same determinant.*

Proof. Let n denote the dimension of A . Recalling Definition 3,

$$\begin{aligned} \det A^\top &= \sum_{i=1}^n (-1)^{i+j} a_{ij}^\top C(i, j) \\ &= \sum_{i=1}^n (-1)^{i+j} a_{ji} C(j, i) \\ &\stackrel{i \leftrightarrow j}{=} \sum_{j=1}^n (-1)^{j+i} a_{ij} C(i, j) = \det A , \end{aligned}$$

where $i \leftrightarrow j$ is a shorthand meaning that the two indices have been swapped in the formula. $\square$

Lemma 2. *A square matrix A and its transpose have the same eigenvalues.*

Proof. The eigenvalues of a A are the roots of its characteristic polynomial $\det(A - \lambda I)$, expressed using λ as unknown and where I is the identity matrix having the same dimension as A . Now, by Lemma 1 $\det(A - \lambda I) = \det(A - \lambda I)^\top$ and, as

$$(A - \lambda I)^\top = A^\top - \lambda I ,$$

$\det(A - \lambda I) = 0$ and $\det(A^\top - \lambda I) = 0$ have the same solutions. The thesis follows. $\square$

Lemma 3. *A RWS matrix A admits $\mathbf{1}$ as eigenvalue.*

Proof. Let $\mathbf{1} = (1, \dots, 1)^\top$. The i -th element of $A \cdot \mathbf{1}$ is

$$(A \cdot \mathbf{1})_i = \sum_j a_{ij}(\mathbf{1})_j = \sum_j a_{ij} = 1 ,$$

by the RWS definition. Considering the equivalent vectorial equation, we have

$$A \cdot \mathbf{1} = \mathbf{1} = \mathbf{1} \cdot \mathbf{1},$$

meaning that $(\mathbf{1}, \mathbf{1})$ is an eigenpair of A . $\square$

Lemma 4. *A CWS matrix A admits $\mathbf{1} = (1, \dots, 1)^\top$ as eigenvalue.*

Proof. The thesis immediately follows by Lemmas 2 and 3, given the obvious fact that the transpose of a RWS matrix is a CWS matrix. $\square$

Lemma 5. *If a square matrix A is RWS, A^k is also RWS for each $k \in \mathbb{N}$.*

Proof. We can prove the thesis by induction on k , noting that when $k = 1$ the thesis is equal to the hypothesis. Concerning the induction step, stating that A^k is RWS means that

$$\sum_{i=1}^n a_{ij}^{(k)} = 1$$

for any j , where $a_{ij}^{(k)}$ denotes the (i, j) entry of A^k . Now, $A^{k+1} = A^k \cdot A$, therefore

$$a_{ij}^{(k+1)} = \sum_{s=1}^n a_{is}^{(k)} a_{sj} ,$$

so that, for fixed j ,

$$\sum_{j=1}^n a_{ij}^{(k+1)} = \sum_{j=1}^n \sum_{s=1}^n a_{is}^{(k)} a_{sj} = \sum_{s=1}^n a_{is}^{(k)} \sum_{j=1}^n a_{sj} .$$

As A^k is RWS for the induction hypothesis, the first sum equals 1. Analogously, by the initial hypothesis also the second sum equals 1. The thesis follows. $\square$

4 PageRank convergence

The following theorem allows us to easily show that Algorithm 1 converges.

Theorem 2. *The principal eigenvector of a CWS square matrix A is associated to the eigenvalue $\lambda_1 = 1$.*

Proof. (by contradiction) By Lemma 4 we know that 1 is among the eigenvalues of A . Let us suppose that the eigenvalue associated to the principal eigenvector is $\lambda_1 > 1$. By Lemma 2 we know that λ_1 is also an eigenvalue of $B = A^\top$, therefore there exists $e \in \mathbb{R}^n$ such that $Be = \lambda_1 e$ and

$$B^k e = \lambda_1^k e ,$$

Letting $C = B^k$, and recalling that by Lemma 5 C is RWS, the component-wise of the latter equation equivalent reads

$$\sum_{j=1}^n c_{ij} e_j = \lambda_1^k e_i .$$

Finally, let $e_{\max} = \max(e_1, \dots, e_n)$: we have that

$$\begin{aligned} \lambda_1^k e_i &= \sum_{j=1}^n c_{ij} e_j && \text{(by previous equation)} \\ &\leq \sum_{j=1}^n c_{ij} e_{\max} && \text{(by definition of } e_{\max}) \\ &= e_{\max} \sum_{j=1}^n c_{ij} \\ &= e_{\max} && \text{(as } C \text{ is RWS).} \end{aligned}$$

Finally, being $\lambda_1 > 1$, we can always choose k so that $\lambda_1^k e_i > G$ for a value G as big as we want, and therefore

$$G < e_{\max} ,$$

which is impossible, as $e_{\max}$ is fixed and G can be chosen as an arbitrarily big value. This means that the original hypothesis $\lambda_1 > 1$ should be rejected, that is we will have $\lambda_1 \leq 1$. Summing up, the eigenvalue associated to the principal eigenvector of A should not exceed 1, but as we already proved that $\lambda = 1$ is an eigenvalue of A , the thesis follows. $\square$

By Theorem 1 we know that, as the execution of Algorithm 1 proceeds, $v(t)$ tends to align to the eigenvector e_1 corresponding to the principal eigenvalue λ_1 . As the transition matrix M we are using is CWS, the previous theorem allows us to say that $\lambda_1 = 1$, therefore (2) now becomes

$$\lim_{t \rightarrow +\infty} v(t) = \alpha_1 e_1 ,$$

thus the algorithm converges. Note that if this happens eventually, meaning that for some t^* we have $v(t^*) = \alpha_1 e_1$, the algorithm stops its execution, because

$$v(t^* + 1) = Mv(t^*) = \alpha_1 M e_1 = \alpha_1 \lambda_1 e_1 = \alpha_1 e_1 = v(t^*) ,$$

thus the convergence condition of the algorithm will be met.