Notes on PageRank

Gabriele Tolomei

May 28, 2020

The PageRank algorithm was introduced by Larry Page and Sergey Brin around 1998[1], and it has been at the core of the Google search engine since then. PageRank is an example of a *link analysis* technique applied to the graph of the Web, with the aim of estimating the popularity - also referred to as *importance score* - of every web page. In other words, the trustworthiness of web pages can be inferred from the link structure of the Web graph. The rationale behind it is that: (*i*) a web page with more incoming links is more important than a page with less incoming links, and (*ii*) a page with a link from a page which is known to be of high importance is also important.
In this notes, we study the *existence* (i.e., convergence) and *uniqueness* of the solution to the problem defined by PageRank.

1 Preliminaries

We let $G = (V, E)$ be a *graph* where V is the set of *vertices* (or *nodes*) and $E \subseteq V \times V$ is the set of *edges* (or *links*) between nodes. Moreover, we assume $|V| = N$.
We use such a representation to model the Web graph, where the set of vertices are web pages and the set of edges the hyperlinks between those.
Moreover, we indicate with $O_v \subseteq V$ the set of nodes (i.e., web pages) that are directly reachable from node v, and with o_v the size of such a set:

$$O_v = \{w \in V : (v, w) \in E\},\ |O_v| = o_v$$

In other words, o_v simply indicates the number of *outgoing links* (i.e., *out-degree*) from node v. Note that similarly we can define $I_v \subseteq V$ as the set of

[1] `https://snap.stanford.edu/class/cs224w-readings/Brin98Anatomy.pdf`

nodes from which v can be reached directly, and i_v as the *in-degree* of that node v:

$$I_v = \{w \in V : (w, v) \in E\},\ |I_v| = i_v$$

A natural way to encode the Web graph - in fact, *any* graph - is by using an *adjacency matrix* $\mathbf{A}_{N \times N}$, which is an N-by-N squared matrix whose (v, w)-th entry $a_{v,w}$ is defined as follows:

$$a_{v,w} = \begin{cases} 1 & \text{if } w \in O_v \\ 0 & \text{otherwise} \end{cases}$$

In addition to that, we can define the out-degree matrix as another square diagonal matrix $\mathbf{L}_{N \times N}$, defined as follows:

$$l_{v,w} = \begin{cases} o_v & \text{if } v = w \\ 0 & \text{otherwise} \end{cases}$$

We can therefore define another square matrix $\mathbf{M}_{N \times N}$, as follows:

$$m_{v,w} = \begin{cases} \frac{1}{o_w} & \text{if } v \in O_w \text{ or, equivalently, if } w \in I_v \\ 0 & \text{otherwise} \end{cases}$$

Note that $\mathbf{M} = (\mathbf{L}^{-1}\mathbf{A})^T$.

2 The PageRank Problem

Informally, each web page is associated with an *importance score* (i.e., a PageRank score), which is just the sum of the PageRank scores of the nodes which point to it. Furthermore, each node propagates its own PageRank score to its adjacent nodes by uniformly distributing it over each of its outgoing links. More formally, the PageRank score r_v associated with the web page v is computed according to the following *recursive* definition:

$$r_v = \sum_{w \in I_v} \frac{r_w}{o_w}$$

We define $\mathbf{r}$ as the $N \times 1$ column vector of PageRank, with one entry for each web page. Moreover, let $\mathbf{m}_v^T$ be the $1 \times N$ row vector corresponding to the v-th row of the matrix $\mathbf{M}$ above, which contains all zeros except for the entries corresponding to the nodes belonging to I_v. As such, we can compute the v-th entry of the PageRank vector $\mathbf{r}$, as follows:

$$r_v = \mathbf{m}_v^T \cdot \mathbf{r} = \sum_{w=1}^{N} m_{v,w} \times r_w = \sum_{w=1}^{N} \frac{1}{o_w} \times r_w = \sum_{w=1}^{N} \frac{r_w}{o_w} = \sum_{w \in I_v} \frac{r_w}{o_w}$$

The last equality derives from the fact that only the non-zero entries of $\mathbf{m}_v^T$ contributes to the summation. If we extend the above computation to all the N entries of $\mathbf{r}$, we will obtain the following matrix equation:

$$\underbrace{\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_v \\ \vdots \\ r_N \end{bmatrix}}_{\mathbf{r}} = \underbrace{\begin{bmatrix} - & \mathbf{m}_1^T & - \\ - & \mathbf{m}_2^T & - \\ & \vdots & \\ - & \mathbf{m}_v^T & - \\ & \vdots & \\ - & \mathbf{m}_N^T & - \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_v \\ \vdots \\ r_N \end{bmatrix}}_{\mathbf{r}}$$

The PageRank problem can be thus formulated as the solution $\mathbf{r}^*$ to the following problem:

$$\mathbf{Mr} = \mathbf{r}$$

Notice that the above formulation is a special case of the more general problem of finding the *eigenvalues* and associated *eigenvectors* of a matrix $\mathbf{A}$:

$$\mathbf{Ax} = \lambda \mathbf{x}$$

In other words, to finding the solution $\mathbf{x}^* \neq 0$ which validates the equation above. In the case of PageRank, this corresponds to finding the *eigenvector* $\mathbf{r}^*$ which solves the above system when the *eigenvalue* is $\lambda = 1$. Note that, for a fixed eigenvalue all the eigenvectors are just scalar multiples of each other. By convention, and since we want the PageRank vector $\mathbf{r}^*$ to capture the relative importance of each web page, we restrict to the eigenvector whose entries sum up to 1, i.e., $|\mathbf{r}^*| = 1$.
We typically compute $\mathbf{r}^*$ using an *iterative solution* named **power method**, which basically execute the following steps:

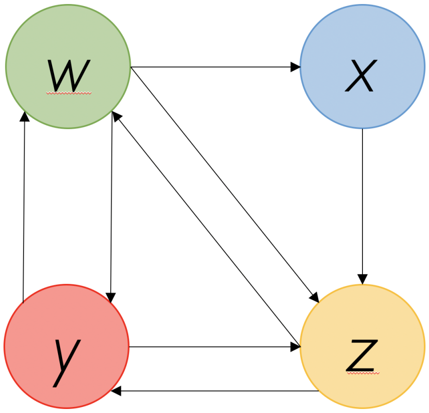
1. Initially, at time $t = 0$, set $\mathbf{r}(t) = (1/N, \dots, 1/N)^T$ (that is, PageRank score is the same for all the N web pages);

2. Compute a new vector $\mathbf{r}(t+1) = \mathbf{M}\mathbf{r}(t)$

3. Go back to step 2. until $\mathbf{r}(t+1) \approx \mathbf{r}(t)$, namely the difference between the current vector and the one obtained at the iteration before is negligible (e.g., $|\mathbf{r}(t+1) - \mathbf{r}(t)| < \epsilon$, $\epsilon > 0$).

Example 1

Consider the small graph below consisting of 4 web pages: w, x, y, and z.



The matrix $\mathbf{M}$ corresponding to this graph is a 4×4 square matrix defined as follows.

$$\mathbf{M}_{4\times 4} = \begin{bmatrix} | & | & | & | \\ \mathbf{m}_w & \mathbf{m}_x & \mathbf{m}_y & \mathbf{m}_z \\ | & | & | & | \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1/2 & 1/2 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1 & 1/2 & 0 \end{bmatrix}$$

Note that each column vector of the matrix above sums up to 1, and therefore $\mathbf{M}$ is said to be *column-stochastic*.
Let us assume the initial PageRank vector $\mathbf{r}(0)$ is set up as follows:

$$\mathbf{r}(0) = \begin{bmatrix} r_w \\ r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Then, we apply the power method as described above. Table 1 below shows how the PageRank vector evolves after few iterations of the algorithm.

#iteration (t)	r_w	r_x	r_y	r_z
0	1	1	1	1
1	1	0.3333	0.8333	1.8333
2	1.3333	0.3333	1.25	1.0833
3	1.667	0.4444	0.9861	1.4028
4	1.1944	0.3889	1.0903	1.3264
5	1.2083	0.3981	1.0613	1.3322
6	1.1968	0.4028	1.0689	1.3316
7	1.2002	0.3989	1.0647	1.3361

Table 1: The evolution of PageRank vector computed with power method.

We may observe that the power method algorithm converges quickly in this example[2]. Within less than 10 iterations, page z has the highest rank. Indeed, page z has 3 incoming links, while the others have either 1 or 2 incoming links. This somehow conforms with the rationale of the intuition of PageRank, which assumes that a page with a larger number of incoming links has higher importance.

Question: *How do we know that such a solution* $\mathbf{r}^*$ *exists and is unique?*

From linear algebra theory, we know the following *Perron-Frobenius theorem*: If a matrix $\mathbf{A}$ is *column-stochastic* with *all positive* entries then:

- $\lambda = 1$ is an eigenvalue of $\mathbf{A}$ with multiplicity one;
- $\lambda = 1$ is the *largest* eigenvalue of $\mathbf{A}$
- there exists a unique (right) eigenvector $\mathbf{x}^*$ associated with the eigenvalue $\lambda = 1$, whose entries sum up to 1.

It turns out that, if we could apply the Perron-Frobenius theorem to our matrix $\mathbf{M}$ we would be guaranteed of the existence and uniqueness of the PageRank vector $\mathbf{r}^*$.

Question: *Can we apply the Perron-Frobenius theorem to our matrix* $\mathbf{M}$*?*

In order to verify whether we can apply the Perron-Frobenius theorem we must check that $\mathbf{M}$ is *column-stochastic* and *positive.*

[2] Note that the entries of the resulting PageRank vector do not sum up to 1: this can be easily achieved by just dividing each component by the sum of the entries.

To be column-stochastic it means that from every web page there exists *at least* one outgoing link. In general, we cannot make such assumption; in fact, the Web graph is far from being perfect and may contain so-called *dangling nodes*, i.e., nodes whose out-degree is 0.

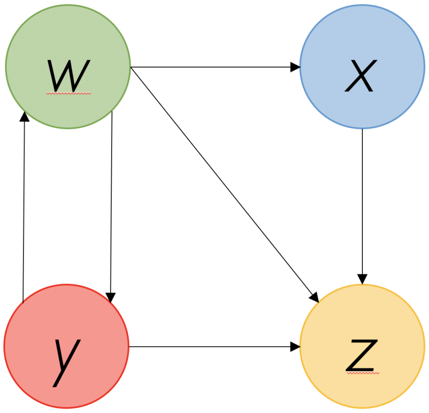
Moreover, even in the lucky case that no dangling node exists – therefore the matrix **M** is actually column-stochastic – still some of the matrix entries may be 0. This would ultimately cause that not every node is reachable from any other node (i.e., there is no path of hyperlinks connecting one web page to *any* other web page), which results in the presence of so-called *spider traps*. In other words, the fact that **M** is column-stochastic just implies that it is *non-negative*, not that it is positive as requested by the Perron-Frobenius theorem.

In the following, we will show how Brin and Page proposed to modify the matrix **M** so as to transform it into a column-stochastic and positive matrix.

2.1 Dealing with Dangling Nodes

Example 2

Consider a slightly modified version of graph shown above. This is still made of 4 web pages: w, x, y, and z, but here web page z has no outgoing links (i.e., it is a dangling node).



The matrix $\mathbf{M}$ corresponding to this graph is the following 4×4 matrix.

$$\mathbf{M} = \begin{bmatrix} | & | & | & | \\ \mathbf{m}_w & \mathbf{m}_x & \mathbf{m}_y & \mathbf{m}_z \\ | & | & | & | \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1/2 & 0 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 1 & 1/2 & 0 \end{bmatrix}$$

As it turns out, the entries in the last column ($\mathbf{m}_z$) are all zero, hence this matrix $\mathbf{M}$ is *not* (column) stochastic. Table 2 shows the evolution of the PageRank vector, if we apply the very same power method as above to this non-stochastic matrix, using the same initialization $\mathbf{r}(0) = (1, 1, 1, 1)^T$.

#iteration (t)	r_w	r_x	r_y	r_z
0	1	1	1	1
1	0.5	0.3333	0.8333	1.8333
2	0.1667	0.1667	1.25	0.6667
3	0.0833	0.0556	0.9861	0.3056
4	0.0278	0.0278	0.0278	0.1111
5	0.0139	0.0093	0.0098	0.0509

Table 2: The evolution of PageRank vector computed with power method on a *non*-stochastic matrix.

Notice that the PageRank vector will converge to zero ultimately, therefore it completely vanishes.

One possible remedy for that is to transform $\mathbf{M}$ into a column-stochastic matrix $\mathbf{M}'$ ($\mathbf{M} \rightsquigarrow \mathbf{M}'$), by replacing all the columns that are full of 0s (e.g., such as $\mathbf{m}_z$) with a constant column vector where each entry is equal to $1/N$. In other words, we define $\mathbf{M}'$ as follows:

$$m'_{v,w} = \begin{cases} \frac{1}{o_w} & \text{if } v \in O_w \\ \frac{1}{N} & \text{if } \sum_{v=1}^{N} m_{v,w} = 0 \\ 0 & \text{otherwise} \end{cases}$$

The change above captures the behavior of a random surfer navigating the Web graph who, once he/she landed on a web page with no outgoing links, simply picks uniformly at random (i.e., with equal probability $1/N$) one of the N possible pages available.

According to this modification, the matrix $\mathbf{M'}$ of the example above becomes:

$$\mathbf{M'} = \begin{bmatrix} 0 & 0 & 1/2 & 1/4 \\ 1/3 & 0 & 0 & 1/4 \\ 1/3 & 0 & 0 & 1/4 \\ 1/3 & 1 & 1/2 & 1/4 \end{bmatrix}$$

If we apply the power method to the problem $\mathbf{M'r} = \mathbf{r}$ – where we replaced the original $\mathbf{M}$ with the column-stochastic $\mathbf{M'}$ – we obtain the evolution of the PageRank vector described in the Table 3 below.

#iteration (t)	r_w	r_x	r_y	r_z
0	1	1	1	1
1	0.75	0.5833	0.58333	2.0833
2	0.8125	0.7708	0.7708	1.6458
3	0.7969	0.6823	0.6823	1.8385
4	0.8008	0.7253	0.7253	1.7487
5	0.7998	0.7041	0.7041	1.7920

Table 3: The evolution of PageRank vector computed with power method fixing the problem of dangling nodes.

We notice that PageRank does not vanish anymore; in fact it converges and z has the highest score as expected.

2.2 Dealing with Spider Traps

Example 3

Consider now another graph of web pages as the one shown below. This is made of 8 web pages: s, t, u, v, w, x, y, and z. Notice that there is no dangling node anymore (i.e., there exists an outgoing link from every web page), but once we get to w, x, y, or z, there is no way to get out from those four web pages. In other words, a random surfer who lands on one of those four web pages gets stuck there forever. This phenomenon is known as *spider trap*, and we will show why it is an issue when computing PageRank.
The 8 × 8 matrix $\mathbf{M}$ associated with the graph above is found below.

$$\mathbf{M}_{8\times 8} = \begin{bmatrix} | & | & | & | & | & | & | & | \\ \mathbf{m}_s & \mathbf{m}_t & \mathbf{m}_u & \mathbf{m}_v & \mathbf{m}_w & \mathbf{m}_x & \mathbf{m}_y & \mathbf{m}_z \\ | & | & | & | & | & | & | & | \end{bmatrix} =$$

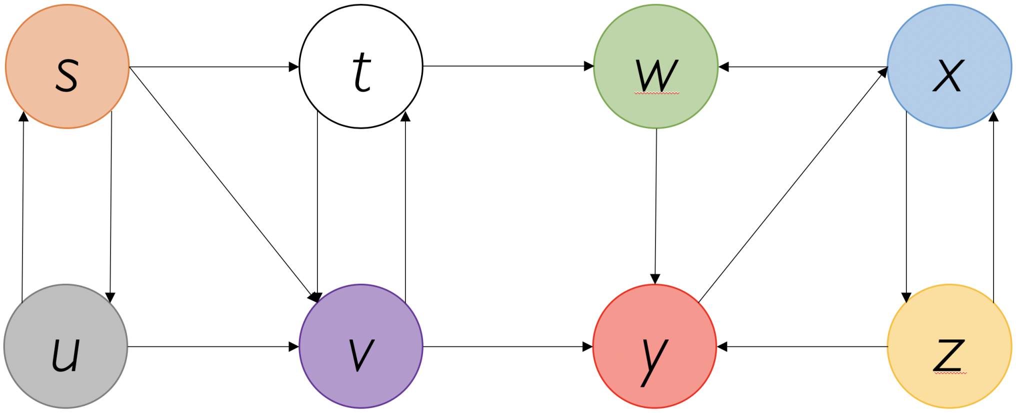
$$= \begin{bmatrix} 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 1/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1/2 &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

If we set $\mathbf{r}(0) = (1/8, \ldots, 1/8)^T$, the evolution of the PageRank vector is:

#iteration (t)	r_s	r_t	r_u	r_v	r_w	r_x	r_y	r_z
0	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
1	0.0625	0.1042	0.0417	0.1667	0.125	0.1875	0.25	0.0625
2	0.0208	0.1042	0.0208	0.0937	0.1458	0.2813	0.2396	0.0938
3	0.0104	0.0538	0.0069	0.0697	0.1927	0.2865	0.2396	0.1406
4	0.0035	0.0382	0.0035	0.0339	0.1701	0.3099	0.2977	0.1432
5	0.0017	0.0181	0.0012	0.0220	0.1740	0.3694	0.2587	0.1549
6	0.0006	0.0116	0.0006	0.0102	0.1937	0.3362	0.2625	0.1847
7	0.0003	0.0053	0.0002	0.0063	0.1739	0.3549	0.2912	0.1681
8	0.0001	0.0032	0.0001	0.0028	0.1801	0.3752	0.2610	0.1774

Table 4: The evolution of PageRank vector computed with power method on a graph containing a spider trap.

We can see that the ranking of pages s to v drop to zero, eventually. But page v has 3 incoming links and should have some non-zero importance. The



power method does not work here because of the particular structure of the graph in the example, i.e., because of the presence of a spider trap. Whenever a graph presents one or more spider traps it is called *reducible*. On the other hand, a graph is called *irreducible* if for any pair of distinct nodes, we can start from one of them, follow the links in the web graph and arrive at the other node, and vice versa. In Example 3, there is no path from w to t, and no path from y to v: the graph is therefore reducible.
To account for the fact that there might not be a path from every web page to any other web page, Brin and Page define the following matrix $\mathbf{G}$, starting from the column-stochastic matrix $\mathbf{M'}$:

$$g_{v,w} = dm'_{v,w} + (1-d)\frac{1}{N}$$

where $d \in (0,1)$ is called *damping factor*. Overall, we obtain:

$$\mathbf{G} = d\mathbf{M'} + \frac{1-d}{N}\mathbf{1}_{N\times N}$$

where $\mathbf{1}_{N\times N}$ is the $N \times N$ square matrix all filled with 1s, i.e.,:

$$\mathbf{1}_{N\times N} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}$$

This last transformation ($\mathbf{M'} \rightsquigarrow \mathbf{G}$) has the side-effect of making the resulting matrix strictly positive and the Web graph fully-connected with so-called *teleport links*. Here the intuition is as follows. Whenever a random surfer lands on a web page he/she will follow one of its outgoing links with probability d, or it will jump to another web page (originally not connected to that) with probability $(1-d)$. Typically, $d = 0.85$ gives good results in practice.
The resulting matrix $\mathbf{G}$ is column-stochastic and positive; as such, the Perron-Frobenius applies to it and we are guaranteed to find a unique solution $\mathbf{r}^*$ to the problem:

$$\mathbf{Gr} = \mathbf{r}$$

If we set the damping factor $d = 0.85$, the matrix $\mathbf{G}$ resulting from the Example 3 above is:

$$\mathbf{G} = \begin{bmatrix} 0.0187 & 0.0187 & 0.4437 & 0.0187 & 0.0187 & 0.0187 & 0.0187 & 0.0187 \\ 0.3021 & 0.0187 & 0.0187 & 0.4437 & 0.0187 & 0.0187 & 0.0187 & 0.0187 \\ 0.3021 & 0.0187 & 0.0187 & 0.4437 & 0.0187 & 0.0187 & 0.0187 & 0.0187 \\ 0.3021 & 0.4437 & 0.4437 & 0.0187 & 0.0187 & 0.0187 & 0.0187 & 0.0187 \\ 0.0187 & 0.4437 & 0.0187 & 0.0187 & 0.0187 & 0.4437 & 0.0187 & 0.0187 \\ 0.0187 & 0.0187 & 0.0187 & 0.0187 & 0.0187 & 0.0187 & 0.8688 & 0.4437 \\ 0.0187 & 0.0187 & 0.0187 & 0.4437 & 0.8688 & 0.0187 & 0.0187 & 0.4437 \\ 0.0187 & 0.0187 & 0.0187 & 0.0187 & 0.0187 & 0.4437 & 0.0187 & 0.0187 \end{bmatrix}$$

Using the matrix **G** above, the evolution of PageRank vector using the power method looks like the following.

#iteration (t)	r_s	r_t	r_u	r_v	r_w	r_x	r_y	r_z
0	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125
1	0.0719	0.1073	0.0542	0.1604	0.1250	0.1781	0.2313	0.0719
2	0.0418	0.1073	0.0391	0.1077	0.1401	0.2459	0.2237	0.0945
3	0.0354	0.0764	0.0306	0.0928	0.1688	0.2491	0.2237	0.1232
4	0.0317	0.0682	0.0288	0.0742	0.1571	0.2613	0.2541	0.1246
5	0.0310	0.0593	0.0277	0.0690	0.1588	0.2877	0.2368	0.1298
6	0.0305	0.0568	0.0275	0.0645	0.1662	0.2752	0.2382	0.1410
7	0.0304	0.0548	0.0274	0.0633	0.1598	0.2811	0.2474	0.1357
8	0.0304	0.0543	0.0274	0.0623	0.1615	0.2867	0.2392	0.1382

Table 5: The evolution of PageRank vector computed with power method fixing the problem of spider traps.

3 Existence and Uniqueness of PageRank

The matrix **G** is now column-stochastic and all of its entries are positive. We can therefore invoke the Perron-Frobenius theorem on it, which guarantees the existence and uniqueness of a (right) eigenvector ($\mathbf{r}^*$) associated with the largest eigenvalue $\lambda_{\text{max}} = 1$ of **G**. In other words, the theorem ensures the PageRank problem $\mathbf{Gr} = \mathbf{r}$ converges to a unique solution $\mathbf{r}^*$.
Before showing this, let us first prove the following lemma, which states that the largest eigenvalue λ_{max} of *any* stochastic matrix $\mathbf{A}$ is $\lambda_{\text{max}} = 1$.

Assume that $A_{N\times N}$ is a square column-stochastic matrix (if it is row-stochastic the proof is similar). If that is the case, the N-dimensional row vector $\mathbf{1} = (\underbrace{1,\dots,1}_{N})^T$ is a trivial *left* eigenvector of $\mathbf{A}$, associated with the eigenvalue $\lambda = 1$, that is:

$$\mathbf{1}^T\mathbf{A} = \mathbf{1}^T$$

Indeed:

$$\underbrace{(1,\dots,1)^T}_{\mathbf{1}^T} \times \underbrace{\begin{bmatrix} a_{1,1} & \dots & a_{1,N} \\ a_{2,1} & \dots & a_{2,N} \\ \vdots & \ddots & \vdots \\ a_{N,1} & \dots & a_{N,N} \end{bmatrix}}_{\mathbf{A}} = \left(\underbrace{\sum_{i=1}^{N} a_{i,1}}_{1}, \underbrace{\sum_{i=1}^{N} a_{i,2}}_{1}, \dots, \underbrace{\sum_{i=1}^{N} a_{i,N}}_{1} \right) = \mathbf{1}^T$$

We therefore convinced ourselves that $\lambda = 1$ is actually an eigenvalue of $\mathbf{A}$, but we still need to prove that this is also the *maximum* of all the possible eigenvalues. Suppose there exists another eigenvalue λ', such that $\lambda' > \lambda = 1$. In other words:

$$\mathbf{x}^T\mathbf{A} = \lambda'\mathbf{x}^T \quad (\mathbf{x}^T \neq \mathbf{0})$$

Let x_j be the largest element of such eigenvector $\mathbf{x}$ associated with λ':

$$\lambda'\mathbf{x}^T = \lambda' x_1, \lambda' x_2, \dots, \lambda' x_N$$

Notice that each column of $\mathbf{A}$ sums up to 1 and each entry of $\lambda'\mathbf{x}^T$ is a convex combination of the elements of $\mathbf{x}^T$, therefore:

$$\lambda' x_j \leq x_j \quad \forall j \in \{1,\dots,N\}$$

However, we previously assumed that there exists $\lambda' > \lambda = 1$, which means:

$$\lambda' x_j > x_j$$

We get to a contradiction, hence $\lambda = 1 = \lambda_{\max}$ is the largest eigenvalue of $\mathbf{A}$. To wrap things up:

- We have proved that any column-stochastic matrix $\mathbf{A}$ has its largest eigenvalue $\lambda_{\max} = 1$;
- We have also shown that the left eigenvector associated with that largest eigenvalue is the $1 \times N$ row vector $\mathbf{1}^T = (1,\dots,1)^T$;

- The above also applies to our column-stochastic matrix $\mathbf{G}$: $\mathbf{1}^T\mathbf{G} = \mathbf{1}^T$, namely $\mathbf{1}^T$ is a left eigenvector of the (modified) Web graph matrix $\mathbf{G}$.

Question: *How does that relate to our original PageRank formulation?*

$$\mathbf{Gr} = \mathbf{r}$$

In the equation above, $\mathbf{r}$ represents a *right* eigenvector associated with the largest eigenvalue $\lambda_{\max} = 1$ (as opposed to the *left* eigenvector $\mathbf{1}^T$). It is easy to show that for any square matrix both the left and right eigenvalues are the same, i.e., the maximum right eigenvalues is also $\lambda_{\max} = 1$.
To see why this is the case, take for example a $K \times K$ square matrix $\mathbf{B}_{K\times K}$. To find the *right* eigenvalues of $\mathbf{B}$ we have to solve for any $\mathbf{x} \neq \mathbf{0}$ the following homogeneous system of equations:

$$\mathbf{Bx} = \lambda\mathbf{x} \Rightarrow (\mathbf{B} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}$$

where $\mathbf{I}$ is the identity matrix.
Similarly, to find the *left* eigenvalues of $\mathbf{B}$ we have to solve for any $\mathbf{x} \neq \mathbf{0}$ the following homogeneous system of equations:

$$\mathbf{x}^T\mathbf{B} = \lambda\mathbf{x}^T \Rightarrow \mathbf{x}^T(\mathbf{B} - \lambda\mathbf{I}) = \mathbf{0}^T$$

In both cases, in order to find a non-trivial solution $\mathbf{x} \neq \mathbf{0}$, the matrix $(\mathbf{B} - \lambda\mathbf{I})$ must be *non-invertible* (*singular*). This happens if and only if the determinant of that matrix is equal to 0, namely:

$$det[(\mathbf{B} - \lambda\mathbf{I})] = 0$$

The determinant is a K-th degree polynomial in λ (as the matrix $\mathbf{B}$ is $K\times K$), which is also known as the *characteristic equation* or *characteristic polynomial* of $\mathbf{B}$. The solutions to this equation are the eigenvalues of $\mathbf{B}$. Since both right and left formulation lead to the same characteristic equation, right and left eigenvalues are also the same.

From the result above, we know that $\lambda_{\max} = 1$ is the largest eigenvalue of our column-stochastic matrix $\mathbf{G}$. We also know that $\mathbf{1}^T$ is the left eigenvector associated with $\lambda_{\max} = 1$ of $\mathbf{G}$, namely $\mathbf{1}^T\mathbf{G} = \mathbf{1}^T$. That means that $\mathbf{G}$ has also another eigenvector (i.e., a right eigenvector), associated with the same largest eigenvalue $\lambda_{\max} = 1$; such a right eigenvector is exactly the PageRank vector $\mathbf{r}$, namely: $\mathbf{Gr} = \mathbf{r}$.

Note that, unless the matrix is symmetric, left and right eigenvectors are generally different from each other.

Now that we are convinced ourselves of the fact that the largest eigenvalue of any column-stochastic matrix is $\lambda_{\max} = 1$, and there is only one of that, we can see how to prove the convergence of PageRank using the Perron-Frobenius theorem.
Our (adjusted) $N \times N$ Web matrix $\mathbf{G}$ is column-stochastic *and* strictly positive, therefore it has $\lambda_{\max} = 1$ as its largest eigenvalue. More precisely, let λ_i, for $i = \{1, 2, \ldots, N\}$ be the eigenvalues of the matrix $\mathbf{G}$, sorted in decreasing order of absolute values.[3] By Perron-Frobenius theorem, we have:

$$1 = \lambda_1 > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_N|$$

At this point, we make a simplifying assumption that all eigenvalues are *distinct*, i.e., we can find N distinct eigenvalues for the $N \times N$ matrix $\mathbf{G}$. We use a fact from linear algebra that if the eigenvalues of a matrices are distinct, we can find an eigenvector for each of the N eigenvalues, and such N eigenvectors are *linearly independent*. Therefore, let $\mathbf{e}_i$, for $i = \{1, 2, \ldots, N\}$ be the eigenvectors, corresponding to the eigenvalues λ_i, respectively. In addition, let $\mathbf{P}$ be an $N \times N$ matrix with the j-th column equal to $\mathbf{v}_j$, for $j = \{1, 2, \ldots, N\}$:

$$\mathbf{P}_{N\times N} = \begin{bmatrix} | & | & \cdots & | & \cdots & | \\ \mathbf{e}_1 & \mathbf{e}_2 & \vdots & \mathbf{e}_j & \vdots & \mathbf{e}_N \\ | & | & \cdots & | & \cdots & | \end{bmatrix}$$

Since the columns of $\mathbf{P}$ are linearly independent, its determinant is non-zero, i.e., $det(\mathbf{P}) \neq 0$, and hence the inverse of $\mathbf{P}$ ($\mathbf{P}^{-1}$) exists.
Now, let $\mathbf{D}$ be a diagonal matrix, whose diagonal entries are the eigenvalues λ_i:

$$\mathbf{D} = \begin{bmatrix} 1 & & & & & \\ & \lambda_2 & & & & \\ & & \ddots & & & \\ & & & \lambda_i & & \\ & & & & \ddots & \\ & & & & & \lambda_N \end{bmatrix}$$

[3] The eigenvalues may be complex, so we have to take the absolute value.

We can put the defining formulation of (right) eigenvectors, $\mathbf{G}\mathbf{e}_i = \lambda_i \mathbf{e}_i$, for $i = \{1, 2, \dots, N\}$, together in a single matrix equation:

$$\mathbf{GP} = \mathbf{PD}$$

Or, equivalently, multiplying both sides of the equation by $\mathbf{P}^{-1}$ (which we know exists because $\mathbf{P}$ is invertible):

$$\mathbf{G} = \mathbf{PDP}^{-1}$$

The factorization of $\mathbf{G}$ as $\mathbf{PDP}^{-1}$ is called the *diagonalization* of $\mathbf{G}$.

Let $\mathbf{r}(0)$ be the initial PageRank vector. The PageRank vector after k iterations is given by:

$$\mathbf{G}^k\mathbf{r}(0) = (\mathbf{PDP}^{-1})^k\mathbf{r}(0) =$$
$$= \underbrace{(\mathbf{PDP}^{-1})(\mathbf{PDP}^{-1})\dots(\mathbf{PDP}^{-1})}_{k \text{ times}}\mathbf{r}(0)$$

Note that:

$$\mathbf{G}^2 = \mathbf{GG} = (\mathbf{PDP}^{-1})(\mathbf{PDP}^{-1}) = \mathbf{PD}(\mathbf{P}^{-1}\mathbf{P})\mathbf{DP}^{-1} = \mathbf{PD}^2\mathbf{P}^{-1}$$

Moreover:

$$\mathbf{G}^3 = \mathbf{G}^2\mathbf{G} = \mathbf{PD}^2\mathbf{P}^{-1}(\mathbf{PDP}^{-1}) = \mathbf{PD}^2(\mathbf{P}^{-1}\mathbf{P})\mathbf{DP}^{-1} = \mathbf{PD}^3\mathbf{P}^{-1}$$

In general:

$$\mathbf{G}^k = \mathbf{PD}^k\mathbf{P}^{-1}$$

Therefore:

$$\mathbf{G}^k\mathbf{r}(0) = \mathbf{PD}^k\mathbf{P}^{-1}\mathbf{r}(0)$$

Notice that as $k \to \infty$, all the entries of the diagonal matrix $\mathbf{D}^k$ goes to 0, except the very first one corresponding to the largest eigenvalue $\lambda_1 = 1$. This is because, the other eigenvalues $\lambda_2, \dots, \lambda_N$ are all *strictly* less than 1.

$$\lim_{k\to\infty} \mathbf{D}^k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

After running the power method algorithm for a long time, the PageRank vector will converge to:

$$\mathbf{P}\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{D}^\infty}\mathbf{P}^{-1}\mathbf{r}(0) = \begin{bmatrix} | & | & \dots & | & \dots & | \\ \mathbf{e}_1 & \mathbf{0} & \vdots & \mathbf{0} & \vdots & \mathbf{0} \\ | & | & \dots & | & \dots & | \end{bmatrix}\mathbf{P}^{-1}\mathbf{r}(0)$$

The resulting PageRank vector is a scalar multiple of the eigenvector $\mathbf{e}_1$. Hence, we can conclude that the PageRank with damping factor d, under the assumption that all eigenvalues of $\mathbf{G}$ are distinct, converges to an eigenvector corresponding to the largest eigenvalue $\lambda_{\text{max}} = \lambda_1 = 1$, namely:

$$\mathbf{r}^* = \begin{bmatrix} | & | & \dots & | & \dots & | \\ \mathbf{e}_1 & \mathbf{0} & \vdots & \mathbf{0} & \vdots & \mathbf{0} \\ | & | & \dots & | & \dots & | \end{bmatrix}\mathbf{P}^{-1}\mathbf{r}(0)$$

4 The PageRank Algorithm

```
Algorithm: PageRank
Input  : A directed Web graph $G = (V, E)$, where $|V| = N$ and its
         associated matrix $\mathbf{M}_{N \times N}$ defined as follows: $\mathbf{M}_{v,w} = \frac{1}{o_w}$ if
         $w$ points to $v$, 0 otherwise ($o_w = |O_w|$ where
         $O_w = \{x \in V : (w, x) \in E\}$);
         A damping factor $d \in (0, 1)$;
         A tolerance $\epsilon > 0$.
Output: The PageRank vector $\mathbf{r}^*_{N \times 1}$
Init   : $t \leftarrow 0; \mathbf{r}(t) \leftarrow \left(\frac{1}{N}, \dots, \frac{1}{N}\right)$;
repeat
    $t \leftarrow t + 1$;
    /* Compute the temporary PageRank score of every page $v$ */
    for $i \leftarrow 1$ to $N$ do
        $r_v^{\text{tmp}}(t) \leftarrow \sum_{w \in I_v} \frac{r_w(t-1)}{o_w}$; /* $r_v^{\text{tmp}}(t) = 0$ if $v$ has no in-links */
    end
    /* Adjust the PageRank score of each page $v$ with teleporting */
    for $i \leftarrow 1$ to $N$ do
        $r_v(t) \leftarrow d \times r_v^{\text{tmp}}(t) + \frac{1-d}{N}$;
    end
until $|\mathbf{r}(t) - \mathbf{r}(t-1)| < \epsilon$
return $\mathbf{r}^* = \mathbf{r}(t)$;
```