Matrix Methods for PageRank: Theoretical Foundations and Computational Applications

Elena Rossi
Politecnico di Torino
Turin, Italy
s349342@studenti.polito.it

Giorgio Zoccatelli
Politecnico di Torino
Turin, Italy
s349395@studenti.polito.it

***Abstract*—This paper examines the computation and analysis of PageRank values for web structures built by adding new pages and links to a base example, and then extends the methodology to a realistic web graph. It studies how topological changes affect page rankings through the leading eigenvector of the damped transition matrix and analyzes the convergence rate of the power method via error norms and spectral properties. The framework is finally applied to the Hollins University website, where PageRank is computed on a large, sparse graph and its empirical distribution is investigated.**

INTRODUCTION

The PageRank algorithm developed by Google is a fundamental method for ranking web pages by interpreting the web as a directed graph and computing the principal eigenvector of a column-stochastic matrix representation. This work builds upon the theoretical foundation outlined in *PageRank.pdf*[1], which details the construction of the web matrix $\mathbf{M}$ incorporating a damping factor m to model random surfing behavior.

In the first part of the paper, we extend a small example web graph by gradually enriching its link structure and observing how the corresponding PageRank vector changes. We then investigate the numerical behavior of the power method used to compute PageRank, with particular attention to its convergence properties. Finally, we apply the same ideas to a realistic web graph built from the Hollins University domain, and discuss the resulting PageRank distribution on a larger, sparsely connected network.

The *repository* linked here contains the full implementation supporting the analyses discussed in this paper, including the small illustrative examples, the convergence experiments, and the Hollins dataset application.

1. EXERCISE 11

Consider again the web in Figure 2.1, with the addition of a page 5 that links to page 3, where page 3 also links to page 5. Calculate the new ranking by finding the eigenvector of $\mathbf{M}$ *(corresponding to* $\lambda = 1$*) that has positive components summing to one. Use* $m = 0.15$.

To solve this exercise, we start by defining the new matrix $\mathbf{A}$ representing our web after the addition of a fifth page that is reciprocally linked with page 3

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} & 1 \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 \end{bmatrix}. \tag{1.1}$$

This matrix illustrates the impact of the new page on the original web: the addition of a fifth column in equation (1.1) reflects the vote cast by page 5 for page 3, while the alteration of the third column accounts for the new vote that page 3 directs towards page 5. It is important to note that, in the original web, page 3 had a unique backlink toward page 1, so the new vote is split accordingly.

Since the web contains no dangling nodes, the matrix $\mathbf{A}$ is column-stochastic: all its entries are non-negative, and each column sums to one. Moreover, the web is assumed to be strongly connected, meaning that every page can be reached from any other page through a finite sequence of links (i.e., $\mathbf{A}$ is irreducible). These properties ensure that 1 is an eigenvalue of $\mathbf{A}$ and that there exists an associated eigenvector with non-negative components, which can be normalized so that $\sum_i x_i = 1$ and interpreted as a PageRank vector. However, without further assumptions such as primitivity, the eigenvalue 1 need not be simple and the eigenspace $V_1(\mathbf{A})$ may have dimension greater than one.

To compute a PageRank vector reliably via the power method, an additional condition is therefore required: the matrix must be primitive, as stated by the Perron–Frobenius theorem. While $\mathbf{A}$ is column-stochastic and irreducible, it may still fail to be primitive if the graph is periodic. This is problematic because the power method converges only when the dominant eigenvalue is simple and strictly larger in modulus than all others. Primitivity guarantees exactly this spectral property, whereas periodicity would prevent convergence. For this reason, we introduce a modified matrix $\mathbf{M}$, defined as a combination of $\mathbf{A}$ and a strictly positive matrix $\mathbf{S}$, which is primitive (i.e., there exists some integer $k > 0$ such that $\mathbf{M}^k > 0$) and thus admits a unique positive eigenvector corresponding to the eigenvalue 1, suitable for a well-defined and numerically stable PageRank computation.

[1]K. Bryan, T. L. Leise, *The $25,000,000,000 Eigenvector: The Linear Algebra behind Google*, SIAM Review, 2006

We know from the theory that our new matrix will have the following shape

$$\mathbf{M} = (1-m)\mathbf{A} + m\mathbf{S} \tag{1.2}$$

where $0 \leq m \leq 1$ is the so-called *damping factor*. Moreover, for a n page web, the matrix $\mathbf{S}$ is a new column-stochastic matrix of dimension $n \times n$ with all entries $1/n$.

In our new web with 5 pages we have

$$\mathbf{S} = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix} \tag{1.3}$$

and if we fix $m = 0.15$ as required we obtain

$$\mathbf{M} = \begin{bmatrix} 0.03 & 0.03 & 0.455 & 0.455 & 0.03 \\ 0.31\bar{3} & 0.03 & 0.03 & 0.03 & 0.03 \\ 0.31\bar{3} & 0.455 & 0.03 & 0.455 & 0.88 \\ 0.31\bar{3} & 0.455 & 0.03 & 0.03 & 0.03 \\ 0.03 & 0.03 & 0.455 & 0.03 & 0.03 \end{bmatrix} \tag{1.4}$$

Since $\mathbf{M}$ is a combination of $\mathbf{A}$ and the strictly positive matrix $\mathbf{S}$, it follows that $\mathbf{M}$ is itself strictly positive and column-stochastic. As a consequence, $\mathbf{M}$ is primitive and satisfies the last condition needed to complete the Perron–Frobenius theorem assumptions. In particular, the dominant eigenvalue of $\mathbf{M}$ is 1, it is simple, and its eigenspace $V_1(\mathbf{M})$ is one-dimensional. Moreover, the corresponding eigenvector (the so-called *Perron eigenvector*) has strictly positive entries.

We are now sure that we are able to compute our PageRank vector $\mathbf{q}$ with positive components such that $\mathbf{Mq} = \mathbf{q}$ and $\sum_i q_i = 1$ by exploiting an iterative method such as the power method. Notice that, to effectively take advantage of the properties of our matrix $\mathbf{M}$ described in equation (1.2), we have to impose $m \in (0, 1]$ in order to avoid falling back onto the original matrix $\mathbf{A}$. The opposite extreme, i.e., $m = 1$, represents the most egalitarian case, where only the matrix $\mathbf{S}$ is considered, yielding the same rank to all pages in the web.

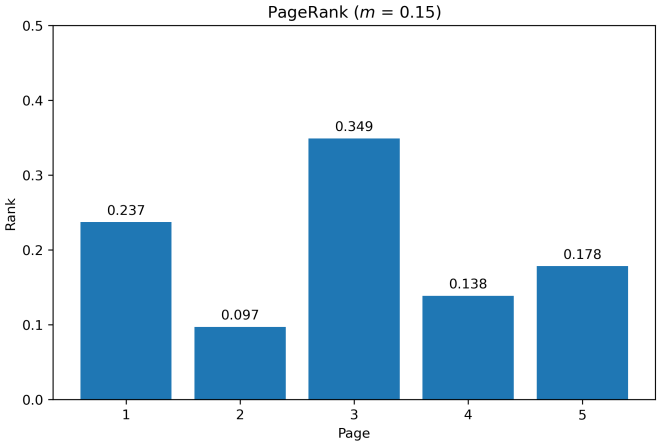
The algorithm adopted for this purpose is the classic power method described as it follows

Algorithm 1 Power method algorithm

Require: $\mathbf{M}$, $\mathbf{v}^{(0)}$, *maxIter*, *relTol*

$\mathbf{v}^{(0)} \leftarrow \frac{\mathbf{v}^{(0)}}{\|\mathbf{v}^{(0)}\|_2}$
$\lambda^{(0)} \leftarrow \infty$
$k \leftarrow 0$
repeat
  $\tilde{\mathbf{v}}^{(k+1)} \leftarrow \mathbf{M}\mathbf{v}^{(k)}$
  $\lambda^{(k+1)} \leftarrow (\mathbf{v}^{(k)})^\top \tilde{\mathbf{v}}^{(k+1)}$
  $\mathbf{v}^{(k+1)} \leftarrow \frac{\tilde{\mathbf{v}}^{(k+1)}}{\|\tilde{\mathbf{v}}^{(k+1)}\|_2}$
  $k \leftarrow k + 1$
until $k \leq$ *maxIter* **or** $\left|\lambda^{(k+1)} - \lambda^{(k)}\right| \geq$ *relTol*$\left|\lambda^{(k+1)}\right|$

This is the exact version implemented in the `pagerank_power_method_classic` function, with some minor modifications to ensure consistency with Python programming language. The plot of the Perron eigenvector for $m = 0.15$ will be the following.



Now that we have successfully computed the rank of the new web, we can proceed to visualize some relevant analyses.

First, we want to verify whether the iterative method we adopted actually converges to the desired result. From the description of the power method, we know that the quantity updating our approximate eigenvalue is the Rayleigh quotient, defined as

$$r_A(\mathbf{v}^k) = \frac{(\mathbf{v}^k)^T \mathbf{M}\mathbf{v}^k}{\|\mathbf{v}^k\|_2^2} \tag{1.5}$$

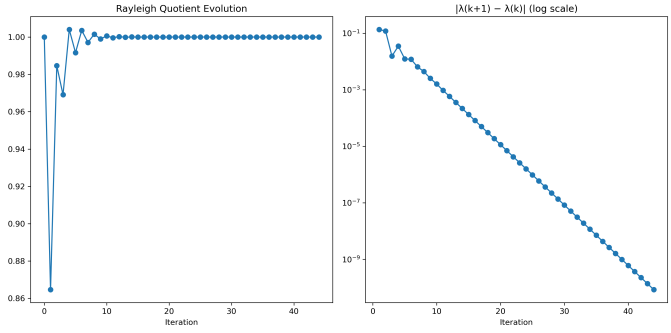
In our specific case, since $\|\mathbf{v}^k\|_2^2 = 1$ and $\mathbf{M}\mathbf{v}^k = \tilde{\mathbf{v}}^{(k+1)}$, the Rayleigh quotient simplifies to

$$r_A(\mathbf{v}^k) = (\mathbf{v}^{(k)})^\top \tilde{\mathbf{v}}^{(k+1)} \tag{1.6}$$

From theory, we know that the Rayleigh quotient satisfies the following property:

$$\lim_{m \to \infty} r_A(\mathbf{v}^k) = \lambda_1 \tag{1.7}$$

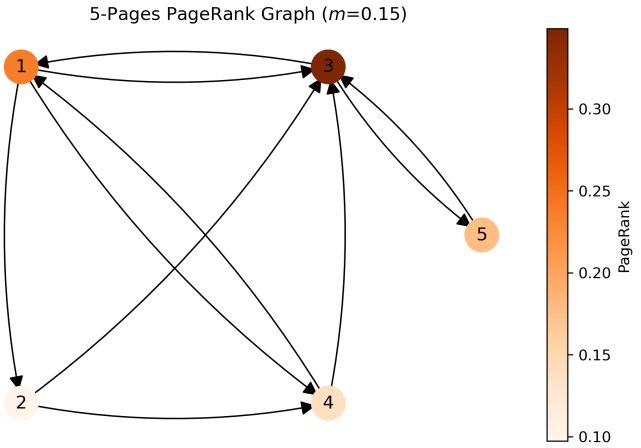
where $\lambda_1 = 1$, as established in our initial analysis. We therefore plot the evolution of $r_A(\mathbf{v}^k)$ together with the evolution of the error $\left|\lambda^{(k+1)} - \lambda^{(k)}\right|$, which we expect to converge to zero (up to the chosen tolerance).



As expected, the plot confirms that the Rayleigh quotient $r_A(\mathbf{v}^k)$ converges to the dominant eigenvalue $\lambda_1 = 1$, while

the error $\left|\lambda^{(k+1)} - \lambda^{(k)}\right|$ decreases towards zero up to the prescribed tolerance. These curves were obtained by storing, at each iteration of the power method, the current Rayleigh quotient and the corresponding error in two separate lists, which were then used to generate the final visualization.

To conclude our analysis of the new web, we can also visualize how each page (node) is weighted according to its rank.



The final part of our analysis aims to highlight the impact of introducing the fifth page into the original web structure. By examining how this additional node interacts with the rest of the network, we can observe how PageRank redistributes importance when the underlying topology changes.

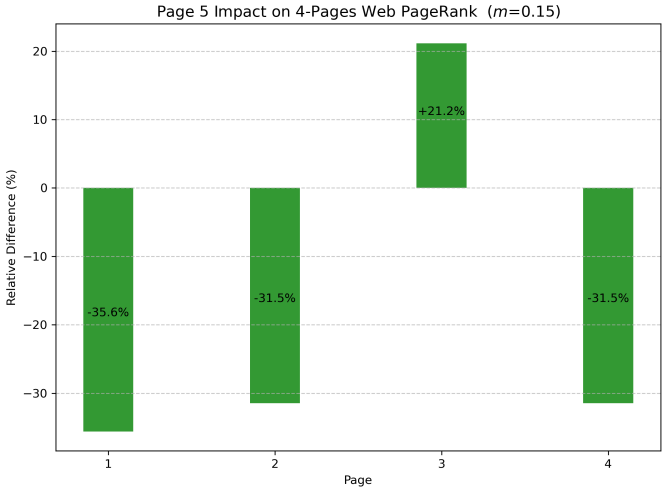
It is worth noting that the PageRank of the original four-page web is computed using the same `pagerank_power_method_classic` function employed for the five-page configuration, with the appropriate modifications to the matrix $\mathbf{M}$ to reflect the different graph structures. Specifically, a dedicated `compute_M_matrix` function was created to make the exercise more modular: it takes as input the link matrix $\mathbf{A}$ and the damping factor m, and returns the corresponding matrix $\mathbf{M}$ along with the initial vector $\mathbf{v}_0$ to be used as input for the PageRank computation. This approach ensures a consistent methodological framework and allows the same PageRank function to be applied to different web configurations without rewriting the core computation.

Since page 3 was the second highest-ranked page in the original configuration, we expect that establishing a mutual link between page 3 and the newly added page 5 will influence the ranking dynamics in two main ways:

- Page 3 is likely to see its rank increase. As the second most authoritative node in the network, page 3 transfers part of its significant importance to page 5, but simultaneously receives the only outgoing link from page 5. This reciprocal connection reinforces page 3's central role and effectively amplifies its influence within the web.
- All other pages are expected to experience a decrease in rank. None of the remaining nodes receives an additional inbound link. With the introduction of page 5, the total rank must now be distributed among a larger set of pages. As a result, pages that do not gain new backlinks will see their relative importance reduced, since the network must account the presence of the new node.

Overall, the addition of page 5 illustrates an essential characteristic of PageRank: even small modifications to the link structure can alter the distribution of importance, particularly when new connections reinforce already authoritative pages.

These properties are empirically verified by the following chart which shows, for each page present in both webs, the relative percentage change in its PageRank with respect to its original value after the introduction of the fifth page.



2. Exercise 12

Add a sixth page that links to every page of the web in the previous exercise, but to which no other page links. Rank the pages using $\mathbf{A}$*, then using* $\mathbf{M}$ *with* $m = 0.15$*, and compare the results.*

The theoretical framework and the numerical implementation for this exercise strictly follow the methodology developed in Exercise 11. The primary difference lies in the topology of the web graph, which now includes a sixth node. This new page is unique because it acts as a pure source: it points to every other page in the network but receives no incoming links itself.

The link matrix $\mathbf{A}$ is expanded to include the outgoing links from Page 6 (distributed uniformly as $1/5$ to the other five pages) and a row of zeros representing its lack of inbound links. Similarly, the matrix $\mathbf{S}$ and the matrix $\mathbf{M}$ are adjusted according their structure.

The new link matrix $\mathbf{A}$ is given by:

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & \frac{1}{5} \\ \frac{1}{3} & 0 & 0 & 0 & 0 & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} & 1 & \frac{1}{5} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{5} \\ 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{5} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.1}$$

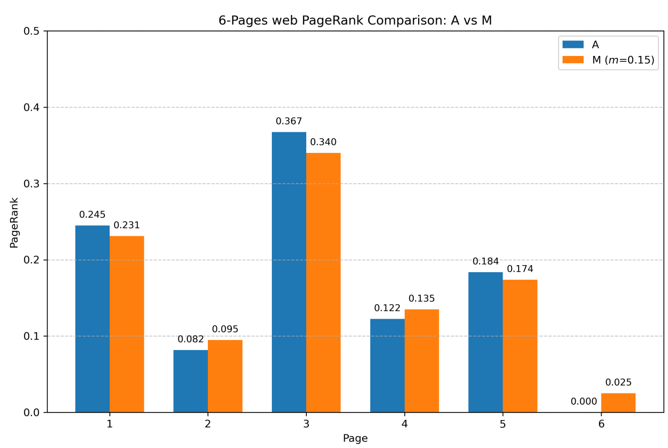
The new matrix $\mathbf{S}$, representing a uniform probability of visiting any page, is given by:

$$\mathbf{S} = \begin{bmatrix} \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} \end{bmatrix} \tag{2.2}$$

Finally, the matrix $\mathbf{M}$ is computed using the standard damping formula $\mathbf{M} = (1-m)\mathbf{A} + m\mathbf{S}$, with $m = 0.15$.

Adding a sixth page that links to all others, but is not itself linked to by any page, produces a distinctive shift in the PageRank distribution with respect to the previous modification.

The figure below compares, on the same six-page web, the PageRank values computed using the two matrices $\mathbf{A}$ and $\mathbf{M}$ (with $m = 0.15$) for each page.



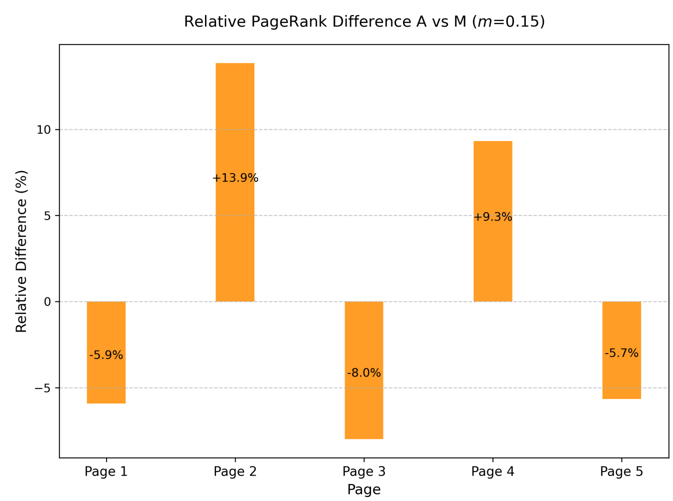
We want to investigate the behavior of the newly added Page 6. In the calculation using $\mathbf{A}$, its PageRank is strictly zero, as no random surfer ever arrives at Page 6 following backlinks: its row in $\mathbf{A}$ consists entirely of zeros, and thus it is unreachable via ordinary link navigation. As a consequence, all the rank is redistributed among the original five pages, and the existing hierarchy is further polarized: already well-connected nodes tend to accumulate more importance, while weaker nodes become less relevant.

When employing the matrix $\mathbf{M}$, which incorporates a uniform teleportation component, Page 6 receives a strictly positive PageRank. This contribution reflects the possibility that a user may visit any page at random rather than navigating solely through links. In this way, $\mathbf{M}$ ensures that even nodes with no inbound links remain discoverable and receive a non-zero share of global importance.

The different behavior of the ranking under $\mathbf{A}$ and under $\mathbf{M}$ on the same six-page graph can be summarized as follows:

- Under $\mathbf{A}$: Page 6 has rank zero and behaves as a pure source. Its outbound links are active, but its lack of incoming links prevents it from accumulating PageRank. The total rank is therefore concentrated on the original five pages, with highly connected nodes (such as Page 3, Page 1, and Page 5) reinforcing their central role, while structurally weaker pages (Pages 2 and 4) receive smaller scores.
- Under $\mathbf{M}$: The teleportation mechanism redistributes a small share of probability to every page at each iteration. Page 6 now acquires a strictly positive score, and the random visit partially compensates nodes that are disadvantaged by the link structure. As a result, when comparing $\mathbf{M}$ to $\mathbf{A}$ on the six-page web, the most central pages lose a small fraction of their rank, whereas less central ones gain relatively more, leading to a more homogeneous stationary distribution.

The impact of replacing $\mathbf{A}$ with $\mathbf{M}$ on the six-page web is quantified more precisely in the figure below, which reports, for each page, the relative percentage change in its PageRank when moving from $\mathbf{A}$ to $\mathbf{M}$.



Eventually, we can compare the rankings obtained with $\mathbf{M}$ before and after the introduction of Page 6. When moving from the five-page web to the six-page web (both ranked with the same damping factor), Page 6 receives a positive share of the total probability, while all original pages experience a slight decrease in their PageRank values. This effect is a direct consequence of the fact that only a new outgoing node has been added: Page 6 distributes its rank to all existing pages, but no new backlinks are created for them. As a result, the total rank is now split between a larger number of pages, so each of the original nodes must give up a small portion of its score to take into account the presence of Page 6.

3. EXERCISE 14

For the Excercise 11, compute the values of $\|\mathbf{M}^k\mathbf{x_0} - \mathbf{q}\|_1$ *and* $\frac{\|\mathbf{M}^k\mathbf{x_0}-\mathbf{q}\|_1}{\|\mathbf{M}^{k-1}\mathbf{x_0}-\mathbf{q}\|_1}$ *for* $k = 1, 5, 10, 50$*, using an initial guess* $\mathbf{x_0}$ *not too close to the actual eigenvector* $\mathbf{q}$*. Determine* $c = \max_{1\le j\le n} |1 - 2\min_{1\le i\le n} \mathbf{M}_{ij}|$ *and the absolute value of the second largest eigenvalue of* $\mathbf{M}$*.*

This exercise shifts the focus from the determination of ranking vectors to the computational dynamics of the PageRank algorithm. Having analyzed the impact of topological changes in previous exercises, we now investigate the convergence rate of the power iteration method applied to the 5-page web structure from Exercise 11.

The primary objective is to quantify how fast the iterative sequence approaches the unique stationary distribution vector $\mathbf{q}$. We perform this quantitative analysis by tracking two key metrics:

- The absolute approximation error in the 1-norm: $\|\mathbf{M}^k\mathbf{x_0} - \mathbf{q}\|_1$.
- The empirical convergence ratio between successive steps: $\rho_k = \frac{\|\mathbf{M}^k\mathbf{x_0}-\mathbf{q}\|_1}{\|\mathbf{M}^{k-1}\mathbf{x_0}-\mathbf{q}\|_1}$.

A fundamental prerequisite for analyzing the approximation error is the definition of the "exact" solution $\mathbf{q}$ (the ground truth) against which the iterations can be measured. While spectral decomposition methods (such as those available in standard linear algebra libraries) are theoretically applicable for small matrices, they scale cubically with the system size ($O(n^3)$), making them unsuitable for realistic web-scale graphs.

To maintain methodological consistency with the scalable nature of the PageRank algorithm, we determine the reference vector $\mathbf{q}_{ref}$ using a high-precision simulation of the power method itself. By executing the algorithm with a tolerance close to machine epsilon ($\epsilon \approx 10^{-16}$) and a high iteration count (> 2000), we obtain a numerical solution that acts as the exact eigenvector for the purpose of this analysis.

The convergence of the power method is not heuristic but is rigorously governed by the linear algebraic properties of the transition matrix $\mathbf{M}$. We evaluate the algorithm's performance against two specific theoretical bounds: the contraction bound derived from the matrix norm and the asymptotic rate derived from the spectral gap.

1. The Contraction Coefficient Proposition: The matrix $\mathbf{M}$ acts as a contraction mapping on the probability simplex. Specifically, we refer to the following property of positive stochastic matrices:

Proposition. Let $\mathbf{M}$ be a positive, column-stochastic $n \times n$ matrix, and let V denote the subspace of $\mathbb{R}^n$ consisting of vectors $\mathbf{v}$ such that $\sum_j \mathbf{v}_j = 0$. Since the error vector $e_k = \mathbf{M}^k\mathbf{x}_0 - \mathbf{q}$ is the difference between two probability distributions, it always lies within this subspace V. It holds that $\mathbf{M}$ maps V into itself, and for any $\mathbf{v} \in V$:

$$\|\mathbf{M}\mathbf{v}\|_1 \le c\|\mathbf{v}\|_1 \tag{3.1}$$

where the contraction coefficient c is defined as:

$$c = \max_{1\le j\le n} \left|1 - 2\min_{1\le i\le n} \mathbf{M}_{ij}\right| \tag{3.2}$$

Since the damping factor guarantees that every entry $\mathbf{M}_{ij}$ is strictly positive (at least m/n), it follows that $c < 1$. This inequality provides a rigorous, worst-case upper bound for the error reduction at every single iteration step.

2. The Asymptotic Spectral Rate (λ_2): While c provides a bound for a single step, the long-term behavior of the error is governed by the subdominant eigenvalue. The error vector can be decomposed into a linear combination of the non-dominant eigenvectors. As $k \to \infty$, the components associated with smaller eigenvalues vanish rapidly, leaving the error dominated by the second largest eigenvalue in modulus, $|\lambda_2|$. Consequently, the ratio of errors is expected to converge asymptotically to this value:

$$\lim_{k\to\infty} \frac{\|\mathbf{M}^k\mathbf{x}_0 - \mathbf{q}\|_1}{\|\mathbf{M}^{k-1}\mathbf{x_0} - \mathbf{q}\|_1} \approx |\lambda_2| \tag{3.3}$$

For the Google matrix, spectral theory suggests that $|\lambda_2| \le 1 - m$. With $m = 0.15$, we expect an upper limit for the asymptotic rate of roughly 0.85.

To verify these theoretical bounds empirically, we set up a simulation using the 5-page web graph defined in Exercise 11.

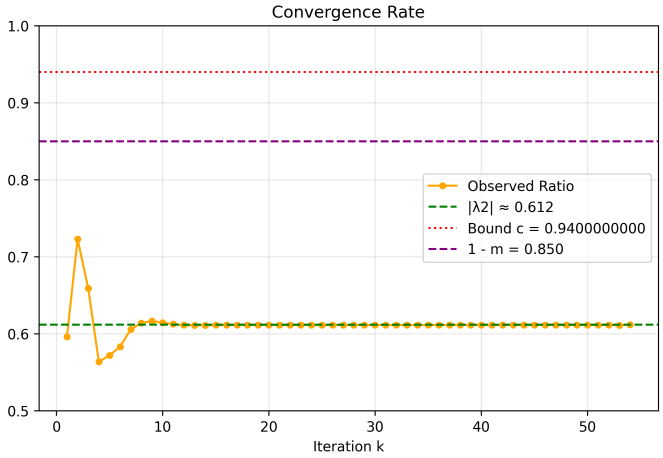
- Matrix Construction.
  The transition matrix $\mathbf{M}$ is constructed using the `compute_M_matrix` utility function defined in previous exercises. Based on the topology of the 5-page web, the matrix $\mathbf{M}$ is dense and strictly positive. The minimum entry in the matrix is $\min \mathbf{M}_{ij}) = 0.03$, which corresponds to the damping probability distributed to nodes with no direct link.
- Initialization of $\mathbf{x}_0$.
  The choice of the initial vector $\mathbf{x}_0$ is critical for observing the transient convergence behavior. While the results presented below utilize a uniform distribution for reproducibility, the interactive Jupyter notebook provided with this paper allows readers to vary the initial vector $\mathbf{x}_0$ (e.g., concentrating probability on a single node or using random values) to empirically observe that the asymptotic convergence rate remains invariant.
  The uniform initialization is defined as:
  $$\mathbf{x}_0 = \left[\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\right]^T = [0.2, 0.2, 0.2, 0.2, 0.2]^T \tag{3.4}$$
  This starting point represents a "blind" guess, assuming no prior knowledge of the web structure. Geometrically, this vector is distant from the final stationary distribution $\mathbf{q}$ (which we know is heavily skewed towards Page 3 due to the link structure), ensuring that the algorithm must perform significant work to converge.
- Measurement Steps.
  The simulation logs the error metrics at iteration steps $k \in \{1, 5, 10, 50\}$. These milestones allow us to observe

the initial rapid decay (transient phase) and the eventual stabilization of the convergence rate (asymptotic phase). The numerical results of the simulation are summarized in Table I. We report the absolute error in the L_1 norm and the empirical convergence ratio ρ_k relative to the previous step.

Step (k)	**Error Norm** ($\|\mathbf{M}^k\mathbf{x}_0 - \mathbf{q}\|_1$)	**Ratio** (ρ_k)
0	0.3719 (approx)	-
1	0.221887	0.596361
5	0.034081	0.571927
10	0.002799	0.614363
50	1.23×10^{-10}	0.611268

TABLE I
EVOLUTION OF THE APPROXIMATION ERROR AND THE EMPIRICAL CONVERGENCE RATIO OVER 50 ITERATIONS.

These data correspond to the 'Uniform' initialization scenario visualized in the computational notebook. The error decays exponentially, as evidenced by the linear trend in the semi-logarithmic plots. By iteration 50, the approximation error is in the order of 10^{-10}, which exceeds the precision requirements for most practical ranking applications. This behavior is visually summarized figure below , which plots the empirical convergence ratio ρ_k against the theoretical bounds. The graph demonstrates that the ratio rapidly stabilizes at the value of the subdominant eigenvalue $|\lambda_2| \approx 0.611$, strictly satisfying the contraction bound $c = 0.94$.



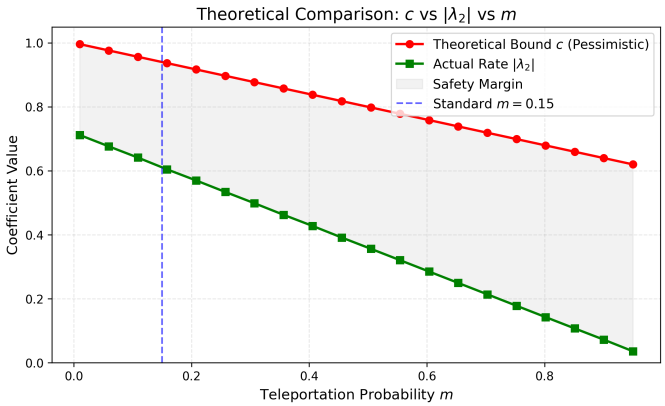
Based on the numerical data and the matrix definition, we compute the theoretical bounds to validate the proposition:

- Theoretical Contraction c.
  Using the minimum entry $\min(\mathbf{M}_{ij}) = 0.03$, the formula yields:
  $$c = |1 - 2(0.03)| = 0.94$$
- Empirical Asymptotic Rate ρ_k.
  As shown in Table I, the ratio ρ_k stabilizes at approximately 0.611 as k increases.
- Spectral Gap $|\lambda_2|$.
  We verified the second largest eigenvalue of $\mathbf{M}$, which is indeed $|\lambda_2| \approx 0.611$.

So, the experimental analysis confirms the validity of the theoretical framework. The inequality $\|\mathbf{Mv}\|_1 \leq c\|\mathbf{v}\|_1$ implies that the convergence rate should be bounded by c. Our results show that:

$$c = 0.94 \geq |\lambda_2| \approx 0.611$$

This confirms that while c provides a guaranteed "worst-case" upper bound, the actual convergence on this specific web graph is significantly faster, driven by the subdominant eigenvalue λ_2. The value of 0.611 is well below the damping factor threshold ($1 - m = 0.85$), indicating that the specific link structure of the 5-page web facilitates rapid mixing of the probability distribution. To demonstrate the robustness of this result, the figure below performs a sensitivity analysis across the entire range of the damping factor m. The plot compares the theoretical bound c with the actual spectral rate $|\lambda_2|$; the shaded area between the two curves represents the safety margin guaranteed by the bound, confirming that the inequality $c \geq |\lambda_2|$ remains valid for any choice of m



4. HOLLINS DATASET

The exercises presented so far have focused on small, fully-specified web graphs with a limited number of pages. While these simple examples effectively illustrate the theoretical foundations and computational dynamics of PageRank, real-world web graphs present a fundamentally different challenge in terms of scale and structure. This section extends the methodological framework to handle realistic datasets, exemplified by the Hollins University website and its complete link structure, which includes thousands of pages and links.

The key transition from the previous setting to a realistic environment lies in the computational representation of the transition matrix $\mathbf{M}$. In Exercises 11-12-14, we worked with dense matrices of dimension 5×5 or 6×6, where every entry could be stored explicitly in memory. This approach becomes computationally prohibitive when scaling to real-world web graphs. A dense matrix representation would require $O(n^2)$ memory, where n is the number of pages. For $n \approx 10^3$,

this translates to storing roughly 10^6 floating-point values, consuming several megabytes of memory.

In practice, real web graphs are extremely sparse: the average node has only a handful of outgoing links, and the number of edges is $O(n)$ rather than $O(n^2)$. Consequently, the link matrix $\mathbf{A}$ and the transition matrix $\mathbf{M}$ are predominantly zeros. To exploit this sparsity, we adopt the Compressed Sparse Row (CSR) format from the SciPy library, which stores only the non-zero entries along with their row and column indices. This reduces memory consumption to $O(\text{nnz})$, where nnz denotes the number of non-zero entries.

However, the challenge of scaling extends beyond simple storage. The standard PageRank formulation involves the web matrix $\mathbf{M} = (1-m)\mathbf{A} + m\mathbf{S}$, where $\mathbf{S}$ is a matrix with all entries equal to $1/n$. Even if the matrix $\mathbf{A}$ is stored in a sparse format, the addition of the dense matrix $\mathbf{S}$ results in a matrix $\mathbf{M}$ that is completely dense. Computing this explicitly, as done in the previous function `compute_M_matrix`, would vanish all memory benefits of the sparse representation.

To address this, the implementation `pagerank_sparse` adopts a matrix-free approach. Instead of constructing $\mathbf{M}$ explicitly, we exploit the structure of the PageRank iteration equation. The power iteration $\tilde{\mathbf{v}}^{(k+1)} = \mathbf{M}\mathbf{v}^{(k)}$ can be rewritten by decomposing the operation:

$$\tilde{\mathbf{v}}^{(k+1)} = [(1-m)\mathbf{A} + m\mathbf{S}]\,\mathbf{v}^{(k)} = (1-m)\mathbf{A}\mathbf{v}^{(k)} + m\mathbf{S}\mathbf{v}^{(k)} \tag{4.1}$$

Since $\mathbf{S} = \frac{1}{n}\mathbf{e}\mathbf{e}^T$ (where $\mathbf{e}$ is a vector of ones) and $\mathbf{v}^{(k)}$ is a probability vector ($\sum v_i = 1$), the term $\mathbf{S}\mathbf{v}^{(k)}$ simplifies to a constant vector where every element is $1/n$. Consequently, the iteration implemented in `pagerank_sparse` becomes:

$$\tilde{\mathbf{v}}^{(k+1)} = (1-m)\mathbf{A}\mathbf{v}^{(k)} + \frac{m}{n}\mathbf{e} \tag{4.2}$$

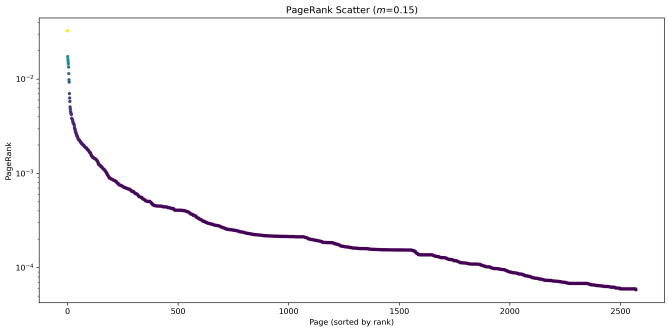
This algebraic manipulation allows us to perform matrix-vector multiplication solely using the sparse matrix $\mathbf{A}$ in CSR format, adding the damping probability mass as a scalar addition post-multiplication. This reduces the computational complexity of each iteration from $O(n^2)$ to $O(\text{nnz} + n)$.

Furthermore, real-world datasets often contain dangling nodes, which are pages with no outgoing links. These nodes act as probability sinks, causing the total rank mass to leak out of the system, violating the column-stochastic property required for the Power Method to converge to a valid distribution. The function `preprocessing` addresses this via an iterative pruning strategy. It does not merely remove nodes with zero out-degree once; it performs a cascade removal. Since removing a dangling node may cause the node pointing to it to become dangling, the function iterates until the core structure is free of dangling nodes. This ensures that the matrix $\mathbf{A}$ passed to the solver is strictly column-stochastic.

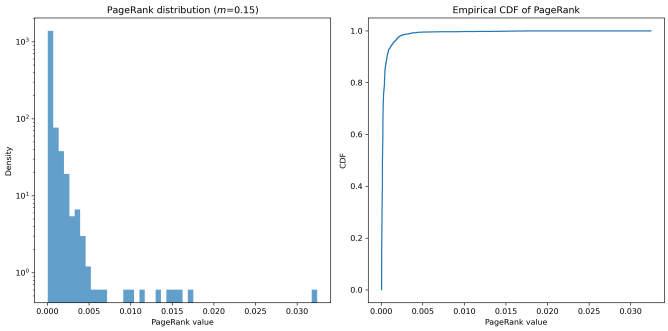
Finally, the convergence criterion has been refined. In the initial exercises, we monitored the stabilization of the eigenvalue λ to approximate λ_1. However, for a stochastic, irreducible, and aperiodic matrix, the Perron-Frobenius theorem assures us that the dominant eigenvalue is exactly $\lambda_1 = 1$. Therefore, estimating λ is redundant. The new implementation focuses directly on the stability of the eigenvector itself. We terminate the loop when the L_1 norm of the difference between consecutive vectors falls below a tolerance threshold:

$$||\tilde{\mathbf{v}}^{(k+1)} - \mathbf{v}^{(k)}||_1 < \tau \tag{4.3}$$

This ensures that the distribution of ranks has stabilized, providing a more direct and computationally efficient stopping condition for large-scale graphs. The Hollins University website graph provides a concrete example of applying PageRank to a real institutional domain. The dataset was assembled by performing a web crawl of all pages reachable from the root domain, extracting their URLs and hyperlink relationships. The resulting network contains thousands of nodes (individual pages) and hyperlinks forming a directed acyclic graph characteristic of hierarchical websites.



The plot above reports the PageRank values of all pages in the Hollins domain, sorted in decreasing order along the horizontal axis. Each point in the scatter corresponds to a single page, while the vertical axis is shown on a logarithmic scale to accommodate the large dynamic range of the scores. The curve displays a smooth, monotonically decreasing trend with a pronounced head and a long tail: a handful of pages with very high PageRank values appear on the left, followed by a rapid decay and a slowly flattening profile as we move towards lower-ranked pages. This shape is consistent with the heavy-tailed behavior theoretically expected for PageRank on real web graphs: the scatter visualization provides an immediate picture of how importance is distributed across the site, confirming that only a small subset of pages plays a dominant structural role in the navigation flow.



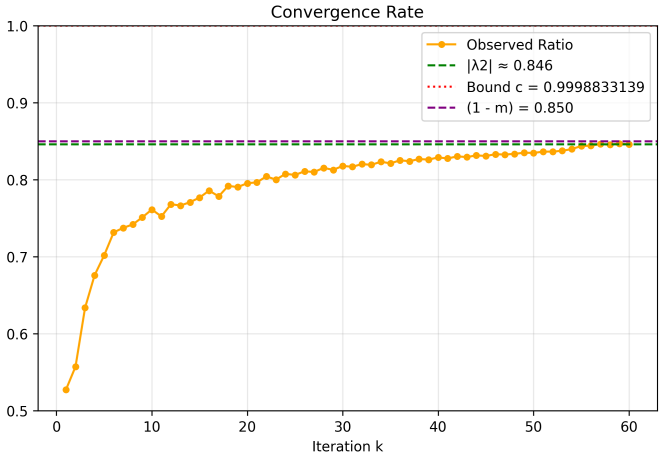
The second figure offers a complementary statistical view of the same PageRank scores. The left panel displays a histogram of the PageRank values on a logarithmic density axis. The bins near zero contain the vast majority of pages, with counts that span several orders of magnitude, while the bins corresponding to larger PageRank values become progressively sparser and eventually contain only a few elements. This reinforces the intuition that most pages in the Hollins website are peripheral from the perspective of link-based importance: they are reachable and contribute to the overall structure, but they accumulate only a negligible fraction of the rank.

The right panel shows the empirical cumulative distribution function (CDF) of the PageRank values. The curve rises very steeply in the neighborhood of the origin, reaching a large fraction of its maximum value for extremely small thresholds, and then approaches 1 only slowly as the threshold increases towards the highest ranks. Interpreted in probabilistic terms, this means that a randomly chosen page in the site is overwhelmingly likely to have a very small PageRank, while a relatively small fraction of pages, located in the upper tail of the distribution, concentrates and explains most of the total rank to be distributed.

The combined analysis of the scatter plot, the histogram bins, and the empirical CDF therefore describes most of the qualitative features of the PageRank vector on the Hollins dataset: strong inequality in the distribution of importance and the presence of a small core of highly central pages.

To complete the study of the Hollins dataset, we shift our focus from the static ranking distribution to the dynamic properties of the algorithm. Specifically, we analyze the convergence behavior on this large-scale graph by applying the same error metrics and spectral bounds established in Exercise 14.

Using the high-precision ground truth vector $\mathbf{v}_{target}$, we monitor the decay of the approximation error e_k and track the empirical convergence ratio ρ_k across iterations. Comparing these empirical results with the theoretical contraction coefficient c reveals a fundamental dichotomy between the behavior of small, dense examples and real-world web graphs. stark contrast emerges when comparing the convergence speed of the Hollins dataset to the 5-page example analyzed previously. In the small synthetic graph, the specific link topology facilitated a rapid mixing of probability, resulting in a convergence rate significantly faster than the worst-case guarantee. Conversely, on the Hollins graph, the convergence is notably slower. The empirical rate $|\lambda_2|$ increases drastically, essentially saturating the upper limit imposed by the damping factor ($1-m = 0.85$). This phenomenon has a clear structural interpretation: the real-world graph is sparse and hierarchically structured, lacking the high connectivity required to mix the probability distribution organically. Consequently, the random walker's exploration is driven almost entirely by the artificial teleportation probability m, rather than by the graph's intrinsic link structure. The algorithm effectively relies on the damping factor to ensure convergence, as the "natural" mixing time of the graph would be far too slow. This structural inefficiency is



further evidenced by analyzing the "safety margin," previously defined as the gap between the pessimistic theoretical bound c and the actual convergence rate $|\lambda_2|$. In the dense toy examples, we observed a wide margin, implying that the theoretical bounds were overly conservative.

However, in the Hollins dataset, this margin collapses. Across the entire range of the damping factor m, the theoretical prediction c and the observed rate $|\lambda_2|$ are nearly coincident. This confirms an important computational property: for large-scale, sparse, and loosely connected matrices, the theoretical worst-case bounds provide a tight and realistic estimate of the algorithm's actual performance. The "pessimistic" scenario described by the mathematics becomes the realistic scenario in large-scale web analysis.