PageRank for Artworks

11630A

Tue Nguyen

Table of contents

Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work, and including any code produced using generative AI systems. I understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

1 Introduction

This project implements a ranking system for artworks using the PageRank algorithm (Page et al. 1999) and experiments it on the Prado Museum Pictures dataset (Mario 2023). The dataset contains 13,487 paintings from the Prado Museum in Madrid. Each painting is on a separate row and and has several attributes such as url, image url, title, author, tags, etc. The goal is to develop a ranking system that assigns a rank score to each piece and how to correctly interpret the results.

First, we start with a discussion of the PageRank algorithm, including its mathematical foundation, the issues that arise, and how they can be fixed using teleportation. The subsequent experiment section details the data processing pipeline, graph construction method, score interpretation, and result analysis.

The project is implemented using Python 3.10 and Spark 3.5.3, with all the code available on my GitHub repository[1].

[1] https://github.com/tuedsci/amd

2 PageRank

This section explores the fundamental concepts and mathematical foundations of PageRank, and the content is based on Chapter 5 of *Mining of Massive Datasets* (Leskovec, Rajaraman, and Ullman 2014)

2.1 Naive PageRank

Let $G = (V, E)$ be a directed graph, where V is the set of nodes and E is the set of edges. Let $n = |V|$ be the number of nodes in the graph, and let each node in V be indexed from 1 to n. Our objective is to assign to each node in V a score representing its importance in the graph.

We introduce the concept of a **random surfer** who starts at a random node and keeps moving between nodes by following the edges of G. In the long run, the surfer will visit each node with a certain probability, with higher probabilities assigned to more important nodes. These probabilities are called the **PageRank scores** of the nodes.

Let $p_t(i)$ denote the probability that the surfer is at node i at time t, and let $p(j \to i|j)$ represent the probability of moving from node j to node i in the next step, given that the surfer is at node j at time t. We assume the transition probabilities $p(j \to i|j)$ are independent of time and depend only on the structure of graph G. Thus, we can organize these probabilities into a fixed transition matrix $M_{n\times n}$, where columns represent the source nodes, rows represent the destination nodes, and entry M_{ij} equals $p(j \to i|j)$. The transition probabilities are assigned by assuming the surfer follows an out-edge of node j uniformly at random, i.e., each neighbor of node j is equally likely to be chosen with the probability $\frac{1}{d_j}$, where d_j is the out-degree of node j. Thus:

$$M_{ij} = \begin{cases} \frac{1}{d_j} & \text{if there is an edge from node } j \text{ to node } i \\ 0 & \text{otherwise} \end{cases}$$

The state of the surfer at time t is summarized by the probability distribution $p_t \in \mathbb{R}^n$ over all nodes in V. Using the law of total probability, we can model the evolution of p_t over time as follows:

$$p_{t+1}(i) = \sum_{j=1}^{n} p(j \to i|j) \cdot p_t(j) = \sum_{j=1}^{n} M_{ij} \cdot p_t(j)$$

Or in matrix notation:

$$p_{t+1} = Mp_t$$

We hope that, as $t \to \infty$, the sequence $\{p_t\}$ converges to a unique stationary distribution π, where entry $\pi(i)$ gives the PageRank score of node i. In fact, when M is a column-stochastic matrix (i.e., each column contains non-negative entries summing to 1) and G is strongly connected and aperiodic, π is guaranteed to exist and be unique.

2.2 Issues

The naive PageRank algorithm may fail when G has **dead ends** (nodes with no out-edges) or **spider traps** (groups of nodes with no out-edges except circling among themselves). In such cases, the surfer gets stuck, causing all probability mass to accumulate at these traps, leaving zero probability mass for other nodes. Another issue aries when G is not strongly connected, i.e., it has **disconnected components**. In this case, once the surfer enters a component, it cannot move to others. In any of these scenarios, we have inaccurate PageRank scores.

2.3 Teleportation

To address these issues, we introduce a modification called **teleportation**. The idea is to allow the surfer to have a small degree of freedom to occasionally jump to a random node at any point in time, instead of strictly following the graph structure. This way, the surfer can always escape from dead ends and spider traps or jump between components in a finite time.

Teleportation is implemented using a **damping factor** $\beta \in (0, 1)$. At each step, the surfer has probability β of following an outgoing edge from the current node and probability $1 - \beta$ of teleporting to a random node within G. During teleportation, each node has an equal probability $\frac{1}{n}$ of being selected. The modified process is described by the following equation:

$$p_{t+1} = \beta M p_t + (1 - \beta)\frac{1}{n}\mathbf{1}$$

where $\mathbf{1}$ is a vector of ones.

3 Experiment

3.1 Graph construction

The dataset consists of 13,487 artworks, with each row representing a unique piece. While each artwork has a distinct `work_url`, this field is too cumbersome to serve as identifiers. Therefore, I created a new column `node` by hashing the `work_url`. Below is an example of the newly created `node` column.

```
+-----------+
|       node|
+-----------+
| 2092045149|
|  146135401|
|-1256791983|
+-----------+
```

A quick examination of the `work_tags` column reveals that each artwork can have multiple tags separated by semicolons. However, there is a redundant `;+` sequence at the end of each tag string that needs to be removed, as shown in the below example.

```
+--------------------------------------------------------------------+
|work_tags                                                           |
+--------------------------------------------------------------------+
|Óleo;Lienzo;Pescador/es;Peces;1785;González Velázquez, Zacarías;+ |
|Aguada;Albayalde;Pluma;Papel;Guerra / Combate;1501;+                |
|Pluma;Tinta parda;Papel amarillento;Desnudo femenino;Santos;1853;+|
|Óleo;Lienzo;Historia Moderna;Armas de fuego;Espada;Lanza;+         |
|Aguada parda;Lápiz negro;Pluma;Papel azulado;Religión;1651;+       |
+--------------------------------------------------------------------+
```

To prepare clean data for further analysis, I performed several transformation steps. First, the trailing `;+` was removed from each tag string. Then, these strings are split into individual tags by semicolons and exploded into separate rows under a new column called `tag`. During this process, I also standardized the tags by trimming whitespaces and converting them to lowercase. Finally, I kept only the `id` and `tag` columns that are relevant for graph construction later. Below are some sample rows from the resulting node-tag data frame.

```
+----------+-------------------+
|node      |tag                |
+----------+-------------------+
|549207445 |albayalde          |
|1839129989|desnudo masculino  |
|-371632852|manzana            |
|1645402546|felipe ii          |
|758883171 |león (panthera leo)|
+----------+-------------------+
```

Further investigation reveals that there are many tags that are popular and shared by many pieces. For example, `óleo` (oil) are shared by 4,177 pieces (31%) or `lienzo` (canvas)

are shared by 3,225 pieces (24%). This suggests that the resulting graph might be highly dense.

From this node-tag dataset, I created an edge list representation of the artwork graph, where two pieces are connected if they share at least one tag. This is done by self-joining the above id-tag data frame on the `tag` column and then removing self-loops and duplicates. While I use the names `src` (source) and `dst` (destination) for the columns, it's important to note that the underlying graph is undirected. For every pair (i, j), there exists a corresponding pair (j, i).

```
+----------+----------+
|       src|       dst|
+----------+----------+
|  549207445|-1766413314|
| 1839129989|-1472967913|
|-1757744297|-1916161485|
|-1757744297| -702265327|
|-1757744297|-1591386974|
+----------+----------+
```

The result is an edge list of 36 million rows bi-directional (i.e., 18 million undirected edges). This means the corresponding graph has a density of 0.2 (i.e., 20% of all possible edges are present), which is unusually high. Most real-world networks are very sparse with a density below 0.001. This suggests that we might need a more proper cleaning of the tags to create a more meaningful relationship. However, this is beyond the scope of this project due to my lack of Spanish skills and domain knowledge in art.

This edge list, together with the node list to account for isolated or dangling nodes (if any), will be used as input for the PageRank algorithm. The resulting PageRank is then converted to a Pandas data frame for further analysis.

3.2 Interpretation

As the graph is formed by connecting artworks that share common tags, the PageRank score can be interpreted as a measure of cross-tag navigability rather than the traditional notion of prestige like in the case of web pages. In our context of artwork network, the PageRank score measures how likely an random surfer would discover an artwork when moving between pieces that share tags.

A high score indicates an artwork serves as a key connection point between different tag-based communities in the collection. For example, if we have a large community of artworks tagged with "oil" and another with "canvas", an artwork tagged with both "oil" and "canvas" would become an important bridge through which a viewer can discover artworks across these communities. This suggest that the PageRank score can be useful for recommending artworks to users based on their interests.

From the above analysis, we can extend to a topic-specific PageRank (Haveliwala 2002) to give a more personalized recommendation. For example, if a viewer is interested in "oil" paintings, we can bias the teleportation to only jump to artworks tagged with "oil". This would give a more focused recommendation based on the viewer's interest.

To construct such PageRank, we define a set S called the **teleportation set** containing

only nodes associated with a specific tag (or topic). The teleportation vector v is modified such that only entries corresponding to nodes in S receive a positive probability $\frac{1}{|S|}$. This means when teleporting, the surfer will only jump to nodes in S, and we have the topic-specific PageRank equation:

$$p_{t+1} = \beta M p_t + (1 - \beta)v$$

where v is defined as:

$$v_i = \begin{cases} \frac{1}{|S|} & \text{if node } i \text{ is in } S \\ 0 & \text{otherwise} \end{cases}$$

3.3 Results

As discussed earlier, the artwork network constructed from shared tags is highly dense (density = 0.2), containing 36M edges despite having only 13,487 nodes. While PageRank computation is still feasible at this scale, creating an equivalent NetworkX graph with 36M edges for verification purposes exceeds memory limits of both my local machine and the free tier of Google Colab. Even a moderate subset of 5,000 nodes generates approximately 5M edges. Therefore, for feasible verification, I selected a smaller subset of 1,000 nodes, generating a more manageable graph with 205,388 edges.

The PageRank scores are then computed using both the Spark implementation and the NetworkX built-in algorithm with a damping factor $\beta = 0.85$. As can be seen in Figure 1, both scores are almost identical. The numerical difference between the two implementations is negligible, with the L_2 norm of their difference being just 4.13×10^{-8}.

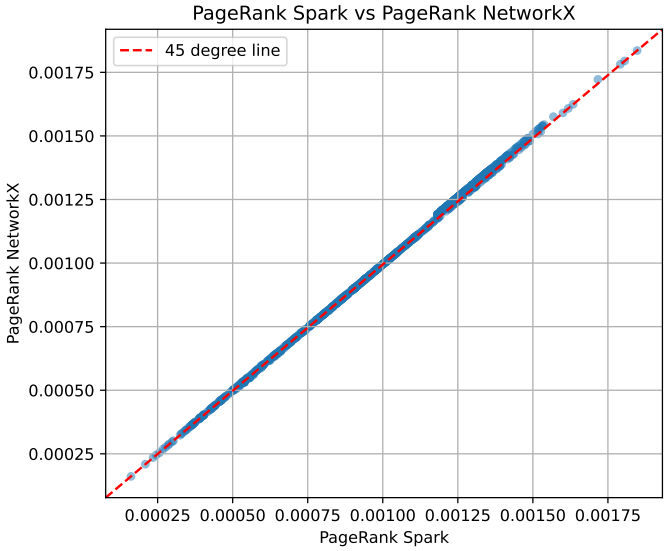

Figure 1: PageRank Spark vs. PageRank NetworkX

Besides the general PageRank, I also computed the topic-specific PageRank for 2 sample tags among most popular ones: `óleo` (oil, 31%) and `pluma` (canvas, 13%). The histograms and pairwise scatter plots for these three PageRank scores are shown in Figure 2

and their Pearson correlations are summarized in Table 1. As expected, these scores are highly correlated, with all pairwise correlations being above 0.9999.

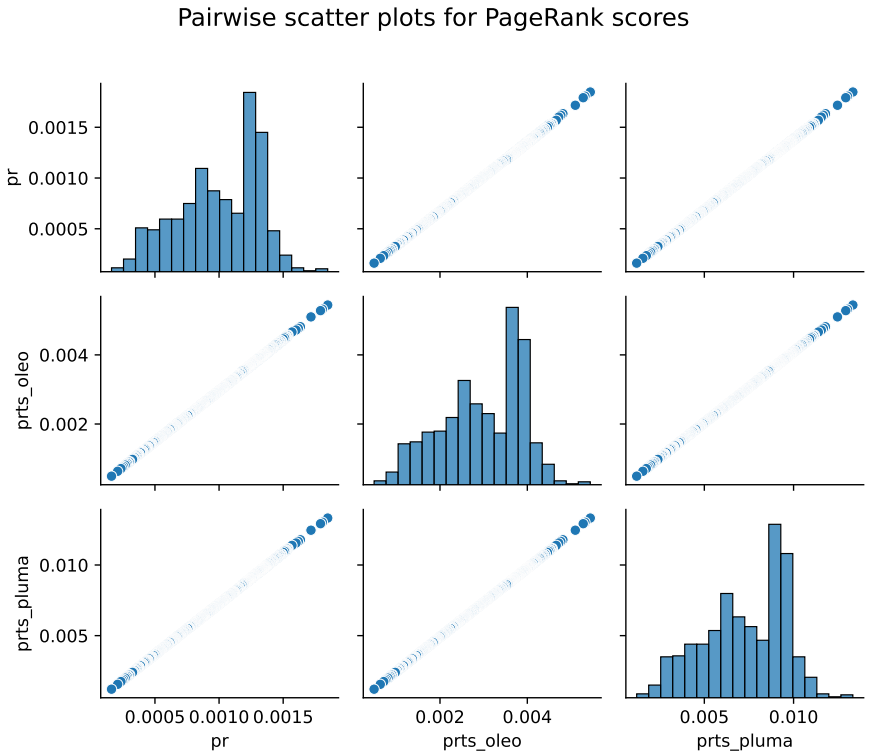

Figure 2: PageRank correlation

PageRank type	pr	prts_oleo	prts_pluma
pr	1.0	0.99991193	0.99995006
prts_oleo	0.99991193	1.0	0.99999209
prts_pluma	0.99995006	0.99999209	1.0

Table 1: Correlation matrix between price and price time series variables

The convergence of the PageRank scores is shown in Figure 3. We can see a rapid initial drop followed by more gradual convergence, which is the typical behavior of the power iteration method. However, the general PageRank converges much faster than the topic-specific PageRank, stabilizing after only 10 iterations. We can also see a clear relationship that larger teleportation sets require less iterations to converge. This is because a larger teleportation set allows the surfer to jump to more different parts of the graph, which helps spread the probability mass more quickly and evenly, resulting in faster stabilization.

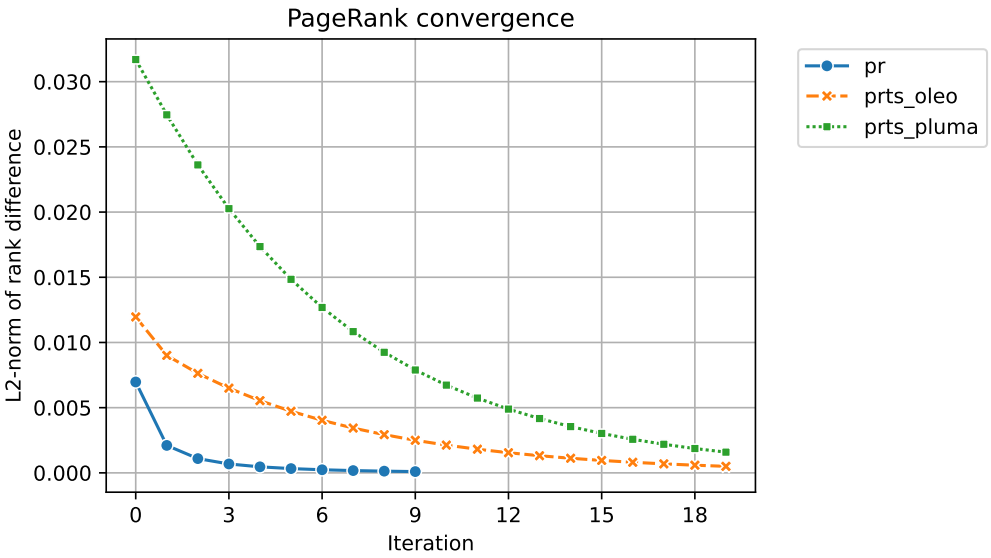

Figure 3: Convergence progress

References

Haveliwala, Taher H. 2002. "Topic-Sensitive PageRank." *Proceedings of the 11th International Conference on World Wide Web*. https://api.semanticscholar.org/CorpusID:129431.

Leskovec, Jure, Anand Rajaraman, and Jeffrey David Ullman. 2014. *Mining of Massive Datasets*. 2nd ed. USA: Cambridge University Press.

Mario, Parreño Lara. 2023. "Prado Museum Pictures." https://www.kaggle.com/datasets/maparla/prado-museum-pictures.

Page, Lawrence, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. "The PageRank Citation Ranking : Bringing Order to the Web." In *The Web Conference*. https://api.semanticscholar.org/CorpusID:1508503.