

# Projeto 5: Como o Google Ordena as Páginas

Herberth Luan Vieira Oliveira  
12559110

Juliana de Abreu Faria  
10336275

Victor Viana de Oliveira Matos  
11810821

Vinícius da Costa Collaço  
11811012

Vitor Sillos Alonso  
9506935

4 de junho de 2022

## 1 Exercício 1

Seja a matriz  $M$  da seguinte forma:

$$M = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1n} \\ m_{21} & m_{22} & \dots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \dots & m_{nn} \end{pmatrix}$$

Se  $M$  satisfaz as hipóteses do teorema 1 então temos as seguintes relações:

$$\begin{cases} m_{11} + m_{21} + \dots + m_{n1} = 1 \\ m_{12} + m_{22} + \dots + m_{n2} = 1 \\ \vdots \\ m_{1n} + m_{2n} + \dots + m_{nn} = 1 \end{cases}$$

Seja o vetor  $y$  da seguinte forma:

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Sendo o vetor  $y$  normalizado, temos que:

$$y_1 + y_2 + \cdots + y_n = 1$$

Fazendo  $z = My$ , temos que:

$$z = \begin{pmatrix} m_{11} & m_{12} & \cdots & m_{1n} \\ m_{21} & m_{22} & \cdots & m_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nn} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

$$z = \begin{pmatrix} m_{11} \cdot y_1 + m_{12} \cdot y_2 + \cdots + m_{1n} \cdot y_n \\ m_{21} \cdot y_1 + m_{22} \cdot y_2 + \cdots + m_{2n} \cdot y_n \\ \vdots \\ m_{n1} \cdot y_1 + m_{n2} \cdot y_2 + \cdots + m_{nn} \cdot y_n \end{pmatrix}$$

Representando a soma de todos os elementos do vetor  $z$  por  $S$ , temos que:

$$S = (m_{11}y_1 + m_{12}y_2 + \cdots + m_{1n}y_n) + (m_{21}y_1 + m_{22}y_2 + \cdots + m_{2n}y_n) + \cdots + (m_{n1}y_1 + m_{n2}y_2 + \cdots + m_{nn}y_n)$$

$$S = (m_{11} + m_{21} + \cdots + m_{n1}) y_1 + (m_{12} + m_{22} + \cdots + m_{n2}) y_2 + \cdots + (m_{1n} + m_{2n} + \cdots + m_{nn}) y_n$$

Sabemos que  $m_{11} + m_{21} + \cdots + m_{n1} = 1$ ,  $m_{12} + m_{22} + \cdots + m_{n2} = 1$  e  $m_{1n} + m_{2n} + \cdots + m_{nn} = 1$ , então:

$$S = (1) y_1 + (1) y_2 + \cdots + (1) y_n$$

$$S = y_1 + y_2 + \cdots + y_n$$

Como visto acima  $y_1 + y_2 + \cdots + y_n = 1$ , então:

$$\boxed{S = 1}$$

Portando está provado que a soma das entradas de  $z$  é igual a 1, então  $z$  é um vetor **normalizado**.

Também, é fácil verificar que todas as entradas de  $z$  são **positivas**, já que provém de somas e multiplicações de números positivos, pois todas as entradas de  $M$  e  $y$  são positivas.

## 2 Exercício 2

### 2.1 Demonstração 1

Sendo a matriz S da seguinte forma:

$$S = \begin{pmatrix} \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \end{pmatrix}$$

Então pela definição da matriz M, temos que:

$$M = (1 - m) A + m \begin{pmatrix} \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \end{pmatrix}$$

Multiplicando por y (pela direita) os dois lados da equação matricial acima temos:

$$My = (1 - m) Ay + m \begin{pmatrix} \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \end{pmatrix} \begin{pmatrix} \frac{1}{n} \\ \frac{1}{n} \\ \vdots \\ \frac{1}{n} \end{pmatrix}$$

Desenvolvendo a multiplicação matricial temos:

$$My = (1 - m) Ay + m \begin{pmatrix} \frac{1}{n^2} + \frac{1}{n^2} + \cdots + \frac{1}{n^2} \\ \frac{1}{n^2} + \frac{1}{n^2} + \cdots + \frac{1}{n^2} \\ \vdots \\ \frac{1}{n^2} + \frac{1}{n^2} + \cdots + \frac{1}{n^2} \end{pmatrix}$$

$$My = (1 - m) Ay + m \begin{pmatrix} \frac{n}{n^2} \\ \frac{n}{n^2} \\ \vdots \\ \frac{n}{n^2} \end{pmatrix}$$

Por fim, obtemos a igualdade pedida:

$$\boxed{\text{My} = (1 - m) \text{Ay} + m \begin{pmatrix} \frac{1}{n} \\ \frac{1}{n} \\ \frac{1}{n} \\ \vdots \\ \frac{1}{n} \end{pmatrix}}$$

## 2.2 Demonstração 2

Como visto anteriormente, por definição a matriz M é dada por:

$$\text{M} = (1 - m) \text{A} + m \begin{pmatrix} \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \end{pmatrix}$$

Multiplicando pelo vetor y (pela direita) em ambos os lados da equação matricial acima tem-se:

$$\text{My} = (1 - m) \text{Ay} + m \begin{pmatrix} \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n} & \cdots & \frac{1}{n} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

Desenvolvendo a multiplicação de matrizes:

$$\text{My} = (1 - m) \text{Ay} + m \begin{pmatrix} \frac{y_1}{n} + \frac{y_2}{n} + \cdots + \frac{y_n}{n} \\ \frac{y_1}{n} + \frac{y_2}{n} + \cdots + \frac{y_n}{n} \\ \vdots \\ \frac{y_1}{n} + \frac{y_2}{n} + \cdots + \frac{y_n}{n} \end{pmatrix}$$

Colocando os valores  $1/n$  em evidência:

$$\text{My} = (1 - m) \text{Ay} + m \begin{pmatrix} \frac{1}{n} (y_1 + y_2 + \cdots + y_n) \\ \frac{1}{n} (y_1 + y_2 + \cdots + y_n) \\ \vdots \\ \frac{1}{n} (y_1 + y_2 + \cdots + y_n) \end{pmatrix}$$

Como todas as entradas de y somam 1, isto é,  $y_1 + y_2 + \cdots + y_n = 1$ , então por fim

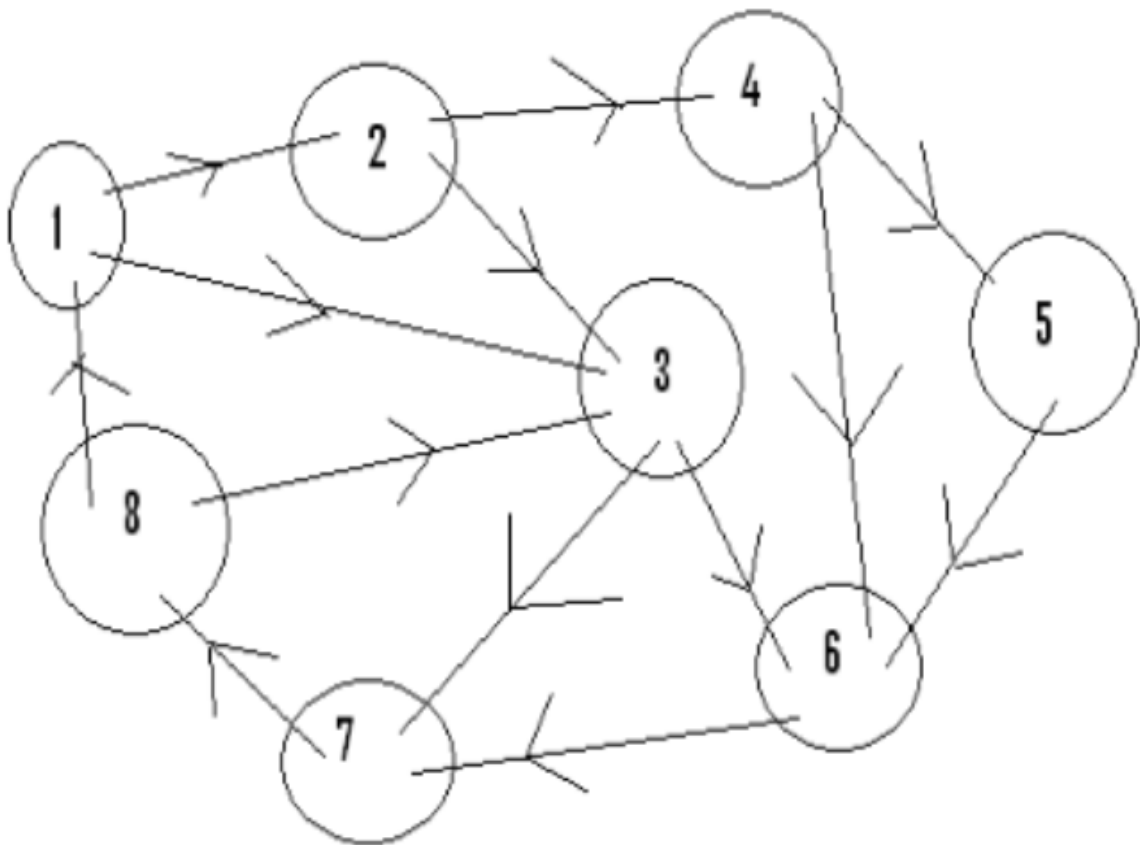
chegamos ao mesmo resultado encontrado anteriormente:

$$My = (1 - m) Ay + m \begin{pmatrix} \frac{1}{n} \\ \frac{1}{n} \\ \frac{1}{n} \\ \vdots \\ \frac{1}{n} \end{pmatrix}$$

### 3 Exercício 3

#### 3.1 Criação da Matriz de Ligação

Para a criação da matriz A partiremos do seguinte grafo:



Representação esquemática para o teste do Exercício 3

A partir da figura, podemos escrever as equações para cada site, com pesos baseados para quantos sites o site referencia, e quantos sites referencial um determinado site.

$$\begin{cases} x_1 = x_8/2 \\ x_2 = x_1/2 \\ x_3 = x_1/2 + x_2/2 + x_8/2 \\ x_4 = x_2/2 \\ x_5 = x_4/2 \\ x_6 = x_3/2 + x_4/2 + x_5/1 \\ x_7 = x_3/2 + x_6/1 \\ x_8 = x_7/1 \end{cases}$$

O sistema linear pode então ser escrita em forma matricial da seguinte maneira:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix}$$

Temos então a nossa matriz de ligação (A), podemos escreve-la em um array para o uso do nosso algoritmo. Com auxílio da biblioteca numpy do python.

```

1 import numpy as np
2
3 A = np.array([[0, 0, 0, 0, 0, 0, 0, 1/2],
4               [1/2, 0, 0, 0, 0, 0, 0, 0],
5               [1/2, 1/2, 0, 0, 0, 0, 0, 1/2],
6               [0, 1/2, 0, 0, 0, 0, 0, 0],
7               [0, 0, 0, 1/2, 0, 0, 0, 0],
8               [0, 0, 1/2, 1/2, 1, 0, 0, 0],
9               [0, 0, 1/2, 0, 0, 1, 0, 0],
10              [0, 0, 0, 0, 0, 0, 1, 0]]
11 )

```

Listing 1: Matriz de Ligação A

## 3.2 Matriz perturbada

A matriz de ligação criada a partir da figura, possui a soma das linhas igual à 1, garantindo que  $Ax = x$ , tenha solução, porém podemos perturbar essa matriz com um valor pequeno para termos uma matriz perturbada com as mesmas características da

matriz de ligação, sem ter valores zerados nela. Pois tendo valores nulos, não são satisfeitas as hipóteses do teorema de Perron-Frobenius.

Substituímos então  $A$  pela matriz perturbada

$$M = (1 - m)A + mS$$

onde  $0 < m < 1$  e  $S$  é a matriz  $n \times n$  com todas as entradas iguais a  $1/n$ : podemos então criar a matriz  $M$  da seguinte maneira:

```
1 n = A.shape[1] #matriz de ligacao NxN
2 S = np.full((n,n),(1/n))
3 M = (1-m)*A + m*S
```

Listing 2: Matriz  $M$

Essa correção feita na matriz  $A$  torna todas as entradas de  $M$  estritamente positivas e a soma dos elementos em cada coluna continua sendo igual a 1. Para todo  $m$  entre 0 e 1.

### 3.3 Método das Potências

Podemos então resolver o sistema linear pelo método das potências (ANTON and RORRES, 2012), dado por:

$$x^{(k)} = Mx^{(k-1)}$$

Se a matriz  $M$  satisfaz as hipóteses do Teorema de Perron-Frobenius,  $x^{(x)}$  converge para o autovetor normalizado associado ao autovalor 1 com taxa  $c$ .

```
1 def MetodoPotencias(M,x0,k=50):
2     x1 = M @ x0 #multiplicacao de matrizes, @ equivale ao np.dot
3     for i in range(k-1):
4         x1 = M @ x1
5     return x1
```

Listing 3: Método das Potências

O valor converge rápido, quanto mais iterações mais próximo do valor real. No programa criado foi adotado arbitrariamente um número padrão de 50 iterações.

### 3.3.1 Exemplo do Algoritmo do Método das Potências

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 1/2 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 1/2 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0.387 \\ 0.129 \\ 0.290 \\ 0.194 \end{pmatrix}$$

Exemplo 1

utilizando o algoritmo criado para resolver o problema temos:

```
1 #Resolucao Intro
2 A_intro = np.array([[0, 0, 1, 1/2],
3                     [1/3, 0, 0, 0],
4                     [1/3, 1/2, 0, 1/2],
5                     [1/3, 1/2, 0, 0]])
6 n = A_intro.shape[1]
7 x0 = np.full((n,1),(1/n))
8 MetodoPotencias(A_intro, x0).round(3)
```

Listing 4: Resolução do Exemplo 1 pelo algoritmo

```
array([ [0.387],
        [0.129],
        [0.29 ],
        [0.194] ])
```

Saída do Exemplo 1

A saída obtida corresponde ao exemplo, a matriz ligação do exemplo não é uma matriz perturbada, porém converge à uma resolução.

## 3.4 Taxa c e cálculo do erro

O método das potências converge o resultado com uma taxa c dada por:

$$c = \max_{1 \leq j \leq n} |1 - 2 * (\min_{1 \leq i \leq n} M_{ij})|$$

assim podemos calcular o c com a seguinte função:



```

1 def calculo_c(M):
2     #minimo de cada coluna
3     min_col = np.amin(M, axis=1)
4     c = max(abs(1-2*min_col))
5     return c

```

Listing 5: Cálculo da taxa c

tendo a taxa c calculada podemos calcular a estimativa do erro a posteriori por:

$$\|x - x^{(k)}\| \leq \frac{c}{1-c} \|x^{(k)} - x^{(k-1)}\|$$

onde x é o autovetor normalizado. Em python podemos normalizar um vetor com auxílio da biblioteca numpy com a função "np.linalg.norm"

### 3.5 Cálculo do Vetor dos Pesos Normalizado

Tendo a matriz de ligação, a matriz perturbada, c e a estimativa do erro. Podemos então criar o algoritmo para ordenar as "páginas".

```

1 def OrdenaPaginas(A, x0, m=0.15, erro=1e-5):
2     n = A.shape[1] #matriz de ligacao NxN
3     S = np.full((n,n), (1/n))
4     M = (1-m)*A + m*S
5     e = erro + 1 #erro inicial para entrar no loop
6     cnt = 0 #contador inicial
7     c = calculo_c(M)
8     xk_1 = x0
9
10    while e > erro:
11        xk = M @ xk_1
12        e = c/(1-c)*np.linalg.norm(xk - xk_1)
13        xk_1 = xk
14        cnt += 1
15
16    print(f'c = {c}')
17    print(f'Vetor dos pesos normalizado:\n{xk}\nApos {cnt} iteracoes ',
18          end='')
19    print(f'com erro menor que {erro}')

```

Listing 6: Algoritmo para a ordenação de páginas até um certo erro

```
[ ]    n = A.shape[1]
        x0 = np.full((n,1),(1/n))
        OrdenaPaginas(A, x0, m=0.15, erro=1e-5)

c = 0.9625
Vetor dos pesos normalizado:
[[0.10564275]
 [0.06364814]
 [0.17759131]
 [0.04580041]
 [0.03821521]
 [0.14617444]
 [0.21847447]
 [0.20445326]]
Após 26 iterações com erro menor que 1e-05
```

Saída do programa para o problema do exercício 3

Após 26 iterações conseguimos a saída com erro menor que  $10^{-5}$  e  $c = 0.9625$ . Podemos observar pelo resultado que a "página" mais importante, no exemplo, é a página 6, seguida pela 7 e em terceiro lugar a página 3. Interessante observar que a página 3 é referenciada mais vezes que a página 7, porém a página 7 acaba tendo mais relevância por ser referenciada por páginas mais relevantes, no caso a 6 e a 3.

## 4 Código Fonte

[https://colab.research.google.com/drive/1nWHHrDlD-4zl05NxqaqwESCP\\_GQQOw\\_S?](https://colab.research.google.com/drive/1nWHHrDlD-4zl05NxqaqwESCP_GQQOw_S?)  
Código criado em Python, com o Jupyter Notebook no Google Colaboratory.

## 5 Considerações Finais

O projeto teve o intuito de observar a maneira com que diferentes paginas da web são relacionadas entre si através de um algoritmo de um sistema de procura. Para tanto, primeiro observou-se o Teorema de Perron-Frobenius, na qual considerando-se uma matriz  $M$   $n \times n$  com todos os seus elementos positivos e cujas colunas somam 1, temos que:

- 1 será o autovalor de maior módulo de  $M$ ;
- seu auto-espaço é unidimensional;

- o autovetor normalizado só tem entradas estritamente positivas.

Isto posto, foi desenvolvido um programa em python que buscou encontrar as relações entre as paginas através do método das potências. Primeiramente, utilizou-se um valor para  $m$  igual a 0,15. Em seguida, calculou-se uma matriz perturbada para que não houvessem valores nulos e assim fossem respeitadas as hipóteses do teorema de Perron-Frobenius. Então, o sistema linear foi resolvido utilizando-se um numero arbitrário de 50 iterações.

No próximo passo, foram obtidas a taxa  $c$  e o cálculo do erro. A taxa  $c$  corresponde à taxa com o qual o método das potências converge ao resultado, e com ele foi possível obter a estimativa do erro. Por fim, foi aplicado o algoritmo e após 26 iterações encontrou-se um valor  $c = 0.9625$  e erro menor que  $10^{(-5)}$ .

Como observações, no exercício desenvolvido o grafo usado possuía poucas ligações, logo a matriz  $A$  possuía valores nulos em todas as colunas, fazendo com que o valor  $c$  fosse para um valor máximo. Em um caso com exemplos reais, talvez houvessem mais conexões e um valor menor para  $c$ , aumentando a complexidade do processamento.

## Referências

ANTON, H. and RORRES, C. (2012). *Álgebra Linear com Aplicações*. Bookman, Porto Alegre RS.