TP PageRank

GUYOT Corentin
MAUGET Clément
ROSE Manon

23 avril 2023

Table des matières

1 Introduction

L'objectif de ce TP est de nous familiariser avec les notions vues en cours pour la recherche de valeurs propres et de mettre en relation cette notion avec l'algorithme PageRank. Il s'agit d'un algorithme de classification des pages web.
Nous considérons que le web est une collection de N pages avec N $\in \mathbb{N}$ très grand.

Le principe de cet algorithme est de considérer qu'une page web est importante si elle est pointée (les autres pages incluent des liens vers cette page) par de nombreuses autres pages web importantes. Ainsi, une page web qui est liée par de nombreux liens provenant d'autres pages importantes aura un PageRank plus élevé qu'une page qui est peu liée ou liée par des pages moins importantes.

Le but final de l'algorithme PageRank est donc de fournir des résultats de recherche pertinents et utiles aux utilisateurs en classant les pages web en fonction de leur pertinence. Nous allons donc attribuer un score de pertinence à chaque page.

Pour modéliser notre problème, nous nous donnons un ordre arbitraire des pages numérotées de i=1 à N. On peut alors représenter le web sous la forme d'un graphe orienté. Par exemple, la page web i, représentée par un point i est reliée par une flèche vers le point j (page web j), signifie que la page web i pointe vers la page web j.
Nous définissons ainsi la matrice d'adjacence $C \in \mathbb{M}_N(\mathbb{R})$ de notre graphe telle que :

$$C_{ij} = \begin{cases} 1 \text{ si la page j pointe vers la page i} \\ 0 \text{ sinon} \end{cases} \tag{1}$$

Avec $C_{ii} = 0$ car une page ne peut pas pointer sur elle-même.

2 Exemples d'illustration

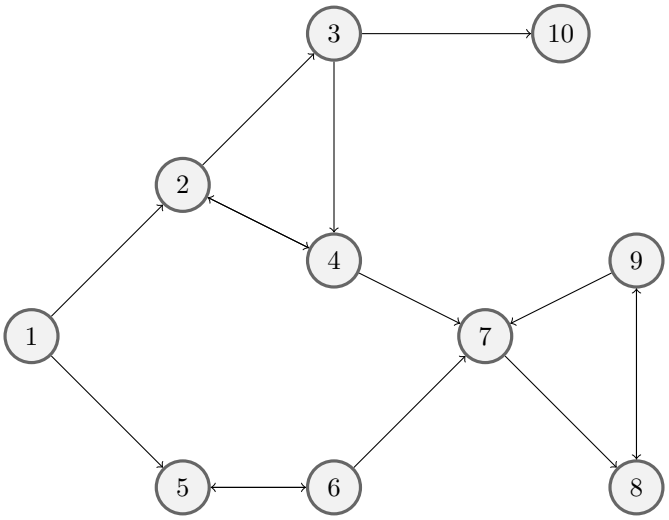
Dans un premier temps, nous illustrons cette notion avec 3 exemples de graphes et nous allons y associer leur matrice d'adjacence et les rentrer sous scilab. Ensuite, nous allons construire la matrice $Q \in \mathbb{M}_N(\mathbb{R})$ définie telle que :

$$Q_{ij} = \begin{cases} \frac{C_{ij}}{N_j} \text{ si } N_j > 0 \\ 0 \text{ sinon} \end{cases} \tag{2}$$

Où $N_j = \sum_{k=1}^{N} C_{kj}$ représente le nombre total de liens sortants de la page j.
Nous prenons ici N=10.

2.1 Exemple 1

Nous avons le premier graphe suivant :



Nous avons la matrice d'incidence correspondante :

$$\mathbb{C}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

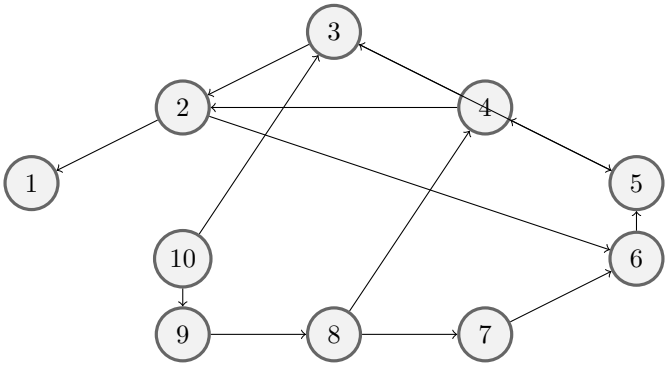
De plus, nous calculons N_j qui correspond à la somme des éléments de chaque colonne : $N_1 = 2$, $N_2 = 2$, $N_3 = 2$, $N_4 = 2$ $N_5 = 1$, $N_6 = 2$, $N_7 = 1$, $N_8 = 1$, $N_9 = 2$, $N_{10} = 0$. Nous obtenons alors :

$$\mathbb{Q}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Nous pouvons vérifier que la somme de chaque colonne vaut 1.

2.2 Exemple 2

Ensuite, nous avons le deuxième graphe suivant :



Nous avons la matrice d'incidence correspondante :

$$\mathbb{C}_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

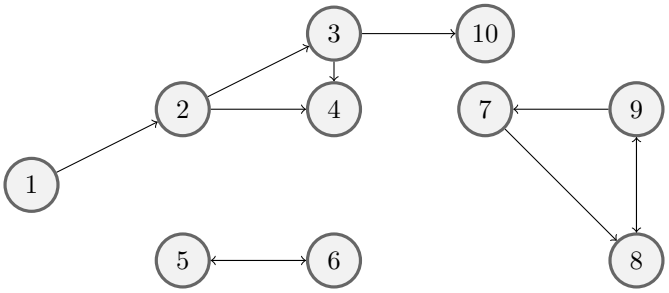
De plus, nous calculons N_j :
$N_1 = 0$, $N_2 = 2$, $N_3 = 2$, $N_4 = 2$ $N_5 = 1$, $N_6 = 1$, $N_7 = 1$, $N_8 = 2$, $N_9 = 1$, $N_{10} = 2$. Nous obtenons alors :

$$\mathbb{Q}_2 = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Nous pouvons, de même, vérifier que la somme de chaque colonne vaut 1.

2.3 Exemple 3

Dans ce dernier exemple, nous avons le graphe suivant :



Nous avons la matrice d'incidence correspondante :

$$\mathbb{C}_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

De plus, nous calculons N_j :
$N_1 = 1$, $N_2 = 2$, $N_3 = 2$, $N_4 = 0$ $N_5 = 1$, $N_6 = 1$, $N_7 = 1$, $N_8 = 1$, $N_9 = 2$, $N_{10} = 0$. Nous obtenons alors :

$$\mathbb{Q}_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

De même que pour les cas précédents, nous pouvons vérifier que la somme de chaque colonne vaut 1.

2.4 Vérification

En faisant tourner le code ci-joint, nous obtenons :

```
Somme des colonnes non nulles de Q de l'exemple 1 : [1. 1. 1. 1. 1. 1. 1. 1. 1.]
Somme des colonnes non nulles de Q de l'exemple 2 : [1. 1. 1. 1. 1. 1. 1. 1. 1.]
Somme des colonnes non nulles de Q de l'exemple 3 : [1. 1. 1. 1. 1. 1. 1. 1. 1.]
```

Nous avons donc bien que la somme des colonnes non nulles est égale à 1.

3 Recherche d'un vecteur propre associé à la valeur propre 1

En introduisant le vecteur $r \in \mathbb{R}^N$ tel que :
$r_i = \sum_{j=1}^{N} Q_{ij} r_j$.

Nous pouvons ainsi ramener le problème du classement des pages web à la recherche d'un vecteur propre associé à sa valeur propre 1.

Grâce à cette commande python :

```python
vp1, _ = np.linalg.eig(Q1)
vp2, _ = np.linalg.eig(Q2)
vp3, _ = np.linalg.eig(Q3)
print("1 appartient-il aux valeurs propres de Q de l'exemple 1 ?", True in np.isclose(vp1, 1))
print("1 appartient-il aux valeurs propres de Q de l'exemple 2 ?", True in np.isclose(vp2, 1))
print("1 appartient-il aux valeurs propres de Q de l'exemple 3 ?", True in np.isclose(vp3, 1))
```

Nous obtenons que les matrices Q des exemples 1 et 3 admettent comme valeur propre 1 tandis que la matrice Q associée à l'exemple 2 n'admet pas 1 comme valeur propre.

```
1 appartient-il aux valeurs propres de Q de l'exemple 1 ? True
1 appartient-il aux valeurs propres de Q de l'exemple 2 ? False
1 appartient-il aux valeurs propres de Q de l'exemple 3 ? True
```

Nous remarquons donc que, parfois, la matrice Q n'admet pas 1 comme valeur propre. Pour pallier ce problème, nous allons introduire une matrice $P \in \mathbb{M}_N(\mathbb{R})$ qui va intervenir notamment sur les colonnes nulles de Q :
$P = Q + \frac{1}{N} e d^t$ avec $e \in \mathbb{R}^N$ vecteur unitaire, et $d \in \mathbb{R}^N$ le vecteur tel que :

$$d_j = \begin{cases} 1 \text{ si } N_j = 0 \\ 0 \text{ sinon} \end{cases} \tag{3}$$

En revenant à nos exemples, nous avons les matrices P suivantes (nous prenons N=10) :

Exemple 1 : $P_1 =$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \frac{1}{N} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{N} \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{N} \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{N} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{N} \\ \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{N} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \frac{1}{N} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & \frac{1}{N} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{1}{2} & \frac{1}{N} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{1}{N} \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{N} \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{10} \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{10} \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{10} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{10} \\ \frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{10} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \frac{1}{10} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & \frac{1}{10} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{1}{2} & \frac{1}{10} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{1}{10} \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{10} \end{pmatrix}$$

En exécutant le code suivant :

```python
#calcul des matrices P
P1 = np.copy(Q1)
P1[:, Nj1 == 0] += 1./N

P2 = np.copy(Q2)
P2[:, Nj2 == 0] += 1./N

P3 = np.copy(Q3)
P3[:, Nj3 == 0] += 1./N

#calcul des vp
vp1, _ = np.linalg.eig(P1)
vp2, _ = np.linalg.eig(P2)
vp3, _ = np.linalg.eig(P3)

print("Multiplicité de 1 en valeur propre de P pour l'exemple 1 :", np.isclose(1, vp1).sum())
print("Multiplicité de 1 en valeur propre de P pour l'exemple 2 :", np.isclose(1, vp2).sum())
print("Multiplicité de 1 en valeur propre de P pour l'exemple 3 :", np.isclose(1, vp3).sum())
```

Nous obtenons comme résultat :

```
Multiplicité de 1 en valeur propre de P pour l'exemple 1 : 1
Multiplicité de 1 en valeur propre de P pour l'exemple 2 : 1
Multiplicité de 1 en valeur propre de P pour l'exemple 3 : 2
```

Nous avons bien une valeur propre de 1 pour cette matrice P_1 et une multiplicité de 1.

$$\textbf{Exemple 2} : P_2 = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \frac{1}{N} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{N} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{N} & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{N} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{N} & 0 & 0 & 0 & 1 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{N} & 0 & \frac{1}{2} & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{1}{N} & \frac{1}{2} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{1}{N} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{N} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{1}{N} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{N} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{10} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{10} & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{10} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{10} & 0 & 0 & 0 & 1 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{10} & 0 & \frac{1}{2} & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{1}{10} & \frac{1}{2} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{1}{10} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ \frac{1}{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{1}{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{10} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

En exécutant le code, nous avons bien une valeur propre de 1 pour cette matrice P_2 et une multiplicité de 1.

Exemple 3 : $P_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \frac{1}{N} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$

$$= \begin{pmatrix} 0 & 0 & 0 & \frac{1}{N} & 0 & 0 & 0 & 0 & 0 & \frac{1}{N} \\ 1 & 0 & 0 & \frac{1}{N} & 0 & 0 & 0 & 0 & 0 & \frac{1}{N} \\ 0 & \frac{1}{2} & 0 & \frac{1}{N} & 0 & 0 & 0 & 0 & 0 & \frac{1}{N} \\ 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{N} & 0 & 0 & 0 & 0 & 0 & \frac{1}{N} \\ 0 & 0 & 0 & \frac{1}{N} & 0 & 1 & 0 & 0 & 0 & \frac{1}{N} \\ 0 & 0 & 0 & \frac{1}{N} & 1 & 0 & 0 & 0 & 0 & \frac{1}{N} \\ 0 & 0 & 0 & \frac{1}{N} & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{N} \\ 0 & 0 & 0 & \frac{1}{N} & 0 & 0 & 1 & 0 & \frac{1}{2} & \frac{1}{N} \\ 0 & 0 & 0 & \frac{1}{N} & 0 & 0 & 0 & 1 & 0 & \frac{1}{N} \\ 0 & 0 & \frac{1}{2} & \frac{1}{N} & 0 & 0 & 0 & 0 & 0 & \frac{1}{N} \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 0 & \frac{1}{10} & 0 & 0 & 0 & 0 & 0 & \frac{1}{10} \\ 1 & 0 & 0 & \frac{1}{10} & 0 & 0 & 0 & 0 & 0 & \frac{1}{10} \\ 0 & \frac{1}{2} & 0 & \frac{1}{10} & 0 & 0 & 0 & 0 & 0 & \frac{1}{10} \\ 0 & \frac{1}{2} & \frac{1}{2} & \frac{1}{10} & 0 & 0 & 0 & 0 & 0 & \frac{1}{10} \\ 0 & 0 & 0 & \frac{1}{10} & 0 & 1 & 0 & 0 & 0 & \frac{1}{10} \\ 0 & 0 & 0 & \frac{1}{10} & 1 & 0 & 0 & 0 & 0 & \frac{1}{10} \\ 0 & 0 & 0 & \frac{1}{10} & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{10} \\ 0 & 0 & 0 & \frac{1}{10} & 0 & 0 & 1 & 0 & \frac{1}{2} & \frac{1}{10} \\ 0 & 0 & 0 & \frac{1}{10} & 0 & 0 & 0 & 1 & 0 & \frac{1}{10} \\ 0 & 0 & \frac{1}{2} & \frac{1}{10} & 0 & 0 & 0 & 0 & 0 & \frac{1}{10} \end{pmatrix}$$

En exécutant le code, nous avons bien une valeur propre de 1 pour cette matrice P_3 et une multiplicité de 2.

Or, nous souhaitons que la valeur propre 1 soit valeur propre simple pour que le problème soit bien posé. Nous introduisons alors la matrice $A_\alpha = \alpha P + (1-\alpha)\frac{1}{N}ee^t$ avec $0 < \alpha < 1$. Nous reprenons ainsi nos trois exemples et pour chacun nous allons calculer cette matrice A_α. Nous allons seulement donner un exemple de matrice A_α car les calculs sont trop longs et peu intéressants puisque nous pouvons directement rentrer la formule sous python.

Pour $\underline{\alpha = 0.1}$:

Exemple 1 : :

$$A_{\alpha 1} = 0.1P + 0.9\frac{1}{N}\begin{pmatrix}1\\1\\1\\1\\1\\1\\1\\1\\1\\1\end{pmatrix}\begin{pmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{pmatrix}$$

$$= \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{10N}\\
\frac{1}{20} & 0 & 0 & \frac{1}{20} & 0 & 0 & 0 & 0 & 0 & \frac{1}{10N}\\
0 & \frac{1}{20} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{10N}\\
0 & \frac{1}{20} & \frac{1}{20} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{10N}\\
\frac{1}{20} & 0 & 0 & 0 & 0 & \frac{1}{20} & 0 & 0 & 0 & \frac{1}{10N}\\
0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & \frac{1}{10N}\\
0 & 0 & 0 & \frac{1}{20} & 0 & \frac{1}{20} & 0 & 0 & \frac{1}{20} & \frac{1}{10N}\\
0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & \frac{1}{20} & \frac{1}{10N}\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & \frac{1}{10N}\\
0 & 0 & \frac{1}{20} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{10N}
\end{pmatrix} + \frac{9}{10N}\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}$$

$$= \begin{pmatrix}
\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{1}{N}\\
\frac{1}{20}+\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{1}{20}+\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{1}{N}\\
\frac{9}{10N} & \frac{1}{20}+\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{1}{N}\\
\frac{9}{10N} & \frac{1}{20}+\frac{9}{10N} & \frac{1}{20}+\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{1}{N}\\
\frac{1}{20}+\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{1}{20}+\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{1}{N}\\
\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & 0.1+\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{1}{N}\\
\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{1}{20}+\frac{9}{10N} & \frac{9}{10N} & \frac{1}{20}+\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{1}{20}+\frac{9}{10N} & \frac{1}{N}\\
\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & 0.1+\frac{9}{10N} & \frac{9}{10N} & \frac{1}{20}+\frac{9}{10N} & \frac{1}{N}\\
\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & 0.1+\frac{9}{10N} & \frac{9}{10N} & \frac{1}{N}\\
\frac{9}{10N} & \frac{9}{10N} & \frac{1}{20}+\frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{9}{10N} & \frac{1}{N}
\end{pmatrix}$$

$$= \begin{pmatrix} \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{1}{10} \\ \frac{14}{100} & \frac{9}{100} & \frac{9}{100} & \frac{14}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{1}{10} \\ \frac{9}{10N} & \frac{14}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{1}{10} \\ \frac{9}{100} & \frac{14}{100} & \frac{14}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{1}{10} \\ \frac{14}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{14}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{1}{10} \\ \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{19}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{1}{10} \\ \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{14}{100} & \frac{9}{100} & \frac{14}{100} & \frac{9}{100} & \frac{9}{100} & \frac{14}{100} & \frac{1}{10} \\ \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{19}{100} & \frac{9}{100} & \frac{14}{100} & \frac{1}{10} \\ \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{19}{100} & \frac{9}{100} & \frac{1}{10} \\ \frac{9}{100} & \frac{9}{100} & \frac{14}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{9}{100} & \frac{1}{10} \end{pmatrix}$$

Nous rentrons donc les formules suivantes pour $\alpha = 0.1$ pour obtenir la valeur de la matrice A_α et avoir le module des valeurs propres de cette matrice pour chaque exemple :

```
alpha1 = 0.1
A_alpha11 = alpha1 * P1 + (1 - alpha1) / N
A_alpha12 = alpha1 * P2 + (1 - alpha1) / N
A_alpha13 = alpha1 * P3 + (1 - alpha1) / N
```

```
np.array2string(np.sort(np.absolute(vp1))[::-1], precision = 2))
np.array2string(np.sort(np.absolute(vp2))[::-1], precision = 2))
np.array2string(np.sort(np.absolute(vp3))[::-1], precision = 2))
```

Les formules sont les mêmes pour $\alpha = 0.5$ en modifiant juste la valeur de α.Les calculs des autres matrices A_α sont donc identiques à ceux de A_{α_1}.

Nous obtenons alors les modules des valeurs propres des matrices A dans l'ordre décroissant suivants :

```
Modules des valeurs propres de A de l'exemple 1 pour alpha = 0.1 :
 [1.00e+00 7.37e-02 7.07e-02 7.07e-02 7.07e-02 7.07e-02 4.12e-02 4.12e-02
 1.02e-09 1.02e-09]
Modules des valeurs propres de A de l'exemple 2 pour alpha = 0.1 :
 [1.   0.07 0.07 0.05 0.05 0.05 0.05 0.05 0.03]
Modules des valeurs propres de A de l'exemple 3 pour alpha = 0.1 :
 [1.00e+00 1.00e-01 1.00e-01 7.26e-02 7.07e-02 7.07e-02 4.22e-02 4.22e-02
 3.86e-02 1.00e-17]
```

Finalement, nous remarquons que cet algorithme (PageRank) revient à trouver le vecteur $r_\alpha \in \mathbb{R}^N$ vérifiant : $r_\alpha = A_\alpha r_\alpha$ avec $0 < \alpha < 1$ et r_α correspondant au vecteur propre associé à la valeur propre 1 de la matrice A_α.

4 Partie Algorithmique

Pour résoudre le problème précédent, on utilise la méthode de la puissance itérée. Voici le code de cette méthode en python :

```python
def puissance_iteree(A_alpha, tol):
    n = A_alpha.shape[0]
    r_1 = np.zeros(n)
    r_1[0] = 1  #avec un 1 et des 0, ||r||=1
    r_0 = r_1 + tol #nous permet de rentrer dans la boucle
    iter = 0
    while np.linalg.norm(r_1 - r_0, ord=1) > tol :
        r_0 = r_1
        q_k = np.dot(A_alpha, r_1)
        r_1 = q_k / np.linalg.norm(q_k, ord=1)
        iter = iter + 1
    return [r_1, iter]
```

Pour tester cette méthode, on l'applique aux trois exemples précédents pour la valeur $\alpha = 0.5$:

```
alpha = 0.5
Exemple 1, vecteur propre :
 [0.09095849 0.10054839 0.09598944 0.10078689 0.10055339 0.10103289
 0.10611759 0.10665959 0.10159739 0.09575594]
4 itérations
Temps d'exécution : 5.1975250244140625e-05

Exemple 2, vecteur propre :
 [0.083548   0.11685519 0.10409581 0.14672878 0.14129128 0.12158894
 0.07611441 0.08795441 0.06767534 0.05414784]
6 itérations
Temps d'exécution : 4.506111145019531e-05

Exemple 3, vecteur propre :
 [0.05912963 0.08875137 0.0813714  0.10184068 0.1178616  0.1178616
 0.09075109 0.13601973 0.12681398 0.07959892]
7 itérations
Temps d'exécution : 4.506111145019531e-05
```

Les matrices A_α sont pleines alors que les matrices Q sont creuses, donc on utilise la formule suivante pour éviter de calculer cette matrice et donc d'alléger les calculs :

$$A_\alpha z = \alpha Q z + \frac{1}{N}(\alpha < d, z > +(1-\alpha) < e, z >)e, \ \forall z \in \mathbb{R}^N$$

On modifie alors la fonction de la méthode de la puissance itérée à l'aide de cette relation et on obtient le code suivant (on utilise la méthode *csr_matrix()* en python pour avoir le caractère creux de Q) :

```python
def puissance_iteree_adaptee(Q, alpha, tol):
    n = Q.shape[0]
    n_j = np.sum(Q, axis=0)
    colonnes_nulles = n_j == 0
    Q_sparse = csr_matrix(Q) #version creuse de Q
    r_1 = np.zeros(n)
    r_1[0] = 1
    r_0 = r_1 + tol
    nb_iterations = 0
    while np.linalg.norm(r_1 - r_0, ord=1) > tol :
        r_0 = r_1
        q_k = alpha * Q_sparse.dot(r_1) + (alpha * np.sum(r_1[colonnes_nulles]) + (1-alpha)*np.sum(r_1))/n
        r_1 = q_k / np.linalg.norm(q_k, ord=1)
        nb_iterations = nb_iterations + 1
    return [r_1, nb_iterations]
```

Nous allons maintenant comparer la vitesse d'exécution des deux algorithmes précédents, celui utilisant la puissance itérée classique et celui prenant en compte le caractère creux de

Q. Nous allons, pour cela, utiliser les trois exemples précédents en utilisant $\alpha = 0.5$. Nous avons déjà les résultats sans utiliser la formule. Utilisons-la maintenant et comparons les temps d'exécution :

```
avec l'utilisation de Q
Exemple 1, vecteur propre :
 [0.05364685 0.09102661 0.07647615 0.09562165 0.09240659 0.09955057
 0.13459979 0.15319153 0.13068791 0.07279234]
7 itérations
Temps d'exécution : 0.001116037368774414

Exemple 2, vecteur propre :
 [0.083548   0.11685519 0.10409581 0.14672878 0.14129128 0.12158894
 0.07611441 0.08795441 0.06767534 0.05414784]
6 itérations
Temps d'exécution : 0.000522613525390625

Exemple 3, vecteur propre :
 [0.05912963 0.08875137 0.0813714  0.10184068 0.1178616  0.1178616
 0.09075109 0.13601973 0.12681398 0.07959892]
7 itérations
Temps d'exécution : 0.0005469322204589844
```

On trouve que l'algorithme de puissance itérée classique est environ 5 fois plus rapide que l'algorithme où l'on utilise le caractère creux de la matrice pour les exemples 1, 2 et 3 respectivement.

Cela s'explique par le fais que les calculs étant relativement petits, le cout de stocker la matrice Q sous forme creuse est trop important par rapport au temps gagné. Cependant, sur des matrices bien plus grandes (tel que de taille 500), la deuxième méthode serait plus rapide.

5 Conclusion

En conclusion de ce TP, nous pouvons dire que nous avons atteint notre objectif qui était de nous familiariser avec les notions de recherche de valeurs propres et d'implémenter l'algorithme PageRank en utilisant le langage de programmation Python.

Nous avons pu comprendre comment l'algorithme PageRank fonctionne pour classer les pages web en fonction de leur importance et de leur pertinence, en utilisant des concepts tels que la matrice d'adjacence, la matrice de transition, la distribution de probabilité stochastique et la recherche de valeurs propres.

Grâce à l'implémentation en Python, nous avons pu appliquer ces concepts pour résoudre des problèmes de classement de pages web et obtenir des résultats précis. Nous avons également pu voir comment l'algorithme PageRank est utilisé dans la réalité pour améliorer les résultats de recherche sur le web.

En somme, ce TP nous a permis d'acquérir des compétences importantes en matière de

recherche de valeurs propres et de mise en pratique de l'algorithme PageRank, ce qui nous sera utile pour résoudre des problèmes similaires dans le futur.