UCLouvain
École polytechnique de Louvain

Université catholique de Louvain

- Rapport LINFO1114 - Mathématiques discrètes « Ranking de réseaux sociaux et pages web : PageRank »

Membres du groupe 30 :
Luka Chd
Noah Fraiture
Lyse Grolaux

Professeur du cours :
Marco Saerens

Assistants :
Sylvain Courtain
Pierre Leleux

15 mai 2022

Table des matières

Partie 1

Présentation du projet

Un algorithme PageRank est un algorithme inventé et utilisé par la société Google afin de classer les pages web entre elles. Il fonctionne sur le principe de classement par "scores". Plus une page sera pointée par d'autres (via des liens), plus son score sera élevé et plus elle sera importante et donc mieux classée.

Cet algorithme est utilisé afin de rendre les recherches Google plus pertinentes. Á cela, nous pouvons ajouter un vecteur de personnalisation qui affinera encore plus le classement des pages proposées à un utilisateur.

1.1 Énnoncé, consignes et données

Dans le cadre de ce projet, nous devions implémenter un algorithme de type "PageRank" avec vecteur de personnalisation et une téléportation non-uniforme qui serait capable de classer les noeuds d'un graphe dirigé selon le score d'importance de chaque noeud attribué par cet algorithme. Pour se faire, chaque groupe a reçu un vecteur de personalisation sur lequel se baser pour trouver le score de chaque noeud du graphe en utilisant deux méthodes différentes :

1. En résolvant un système d'équations linéaires
2. En implémentant la "power method" à l'aide de la bibliothèque Numpy

L'algorithme se compose de deux méthodes implémentées en Python nommées *pageRankLinear* et *pageRankPower* qui correspondent respecivement aux deux méthodes de résolution listées ci-dessus.

Premièrement voici le vecteur de personnalisation qui nous a été assigné :

$$v = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 0.0244 & 0.0065 & 0.0919 & 0.22 & 0.0473 & 0.0022 & 0.1847 & 0.0518 & 0.1408 & 0.2304 \end{pmatrix}$$

Nous nous sommes aussi vu attribuer un graphe dirigé et pondéré sur lequel se baser (cf. fig. 1.1).

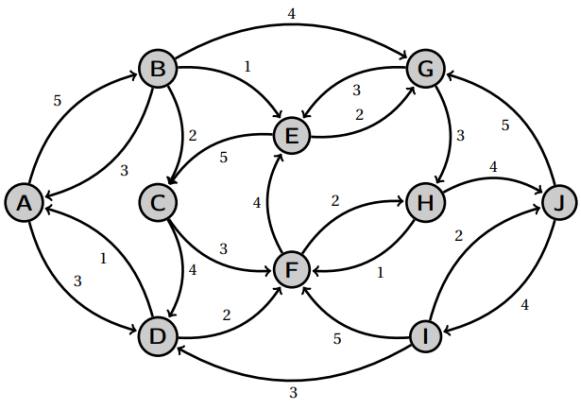

FIGURE 1.1 – Un graphe dirigé et pondéré représentant un réseau de 10 noeuds.

Partie 2

Rappel - comment calculer le vecteur de scores PageRank ?

2.1 En résolvant un système d'équations linéaires

Cette méthode correspond à la fonction *pageRankLinear* et possède comme entrées : A la matrice d'adjacence correspondante, α le facteur de téléportation et v le vecteur de personnalisation. Elle doit retourner un vecteur x contenant les scores d'importance de chacun des noeuds du graphe dans le même ordre que la matrice d'adjacence.

Pour se faire, il faut tout d'abord former une matrice P, aussi appelée la matrice des probabilités de transition, avant de simplement résoudre un système d'équations donné[1] :

$$\begin{cases} (I - \alpha P)^T x = (1 - \alpha)v, \\ x^T e = 1. \end{cases}$$

Où :
— $(I - \alpha P)$ est une matrice **M**, inversible et dont la somme de chacune de ses lignes est égale à $1 - \alpha$
— x est un vecteur-ligne de 1 x n
— α est le paramètre de téléportation ($0<\alpha<1$),
— P une matrice dont toutes les lignes sont normalisées,
— v est le vecteur de personnalisation.

Ici, l'équation de normalisation ($x^T e = 1$) sert simplement à s'assurer que v est bien un vecteur de probabilité.

2.2 En implémentant la « power method »

Cette méthode de calcul correspond quant à elle à la fonction *pageRankPower*. Celle-ci possède les mêmes entrées que la fonction précédente. À savoir : A la matrice d'adjacence correspondante, α le facteur de téléportation et v le vecteur de personnalisation. Elle retourne aussi un vecteur x qui possède les même spécificités. Il contient les scores d'importance de chacun des noeuds du graphe dans le même ordre que la matrice d'adjacence.

Cette fois-ci, nous devons d'abord former la matrice G (Google) des noeuds du graphe. Il s'agit aussi d'une formule à appliquer :

$$G = \alpha P + (1 - \alpha)ev^T$$

Ensuite, il suffit d'initialiser les scores avant d'itérer cette matrice G jusqu'à ce que le vecteur de scores x se stabilise (il y a convergence). On retourne alors le résultat trouvé.

1. Equation donnée à la slide 144 du cours sur les graphes (chapitre 10)

2.3 Importance du vecteur v et du paramètre α

Un peu plus haut, nous avons brièvement introduits ces deux variables sans pour autant les développer. Il parait évident que vu qu'elles interviennent toutes deux dans le calcul de score PageRank, elles ont un réel impact sur le résultat final.

2.3.1 Impact du paramètre de téléportation α

Le paramètre α est une constante qui contrôle l'influence de la structure première du graphe. Plus simplement, si la valeur de cette constante tend vers 0, le random walker va avoir tendance à suivre le chemin attendu (le nombre d'itérations nécessaires sera moindre) tandis que si α tend vers 1, le nombre d'itérations va considérablement augmenter et va engendrer le fait que l'algorithme deviendra fluctuant. Autrement dit, son efficacité va devenir aléatoire (Langville Meyer, 2006).

2.3.2 Impact du vecteur de personnalisation v

De son coté, l'apparition d'un vecteur de personnalisation a pour conséquence que les probabilités de téléportations ne sont plus uniformément distribuées et suivent à la place les probabilités données dans v (Langville Meyer, 2006). On peut en déduire que différents vecteurs de personnalisation produisent différents classements de pages (PageRankings) et nous auront différents résultats de recherche en fonction des différents vecteurs donnés.

Partie 3

Application des deux méthodes

Maintenant que toutes les notions nécessaires ont été introduites, nous pouvons passer à la pratique. Pour ce faire, nous avons toujours notre vecteur de personnalisation v et notre graphe dirigé et pondéré (cf. fig. 1.1).

3.1 Application de la méthode du système d'équations linéaires

Comme dit précédemment, ce système est l'un de ceux qui permet de calculer le score PageRank du graphe associé. En entrant toutes nos données, nous obtenons ce système :

$$(I-\alpha\begin{bmatrix} 0 & 0.625 & 0 & 0.375 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.3 & 0 & 0.2 & 0 & 0.1 & 0 & 0.4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.57142857 & 0 & 0.42857143 & & & & \\ 0.33333333 & 0 & 0 & 0 & 0 & 0.66666667 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.71428571 & 0 & 0 & 0 & 0.28571429 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.66666667 & 0 & 0 & 0.33333333 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0.8 \\ 0 & 0 & 0 & 0.3 & 0 & 0.5 & 0 & 0 & 0 & 0.2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.55555556 & 0 & 0.44444444 & 0 \end{bmatrix})^T x = (1-\alpha)\begin{pmatrix} 0.0244 \\ 0.0065 \\ 0.0919 \\ 0.22 \\ 0.0473 \\ 0.0022 \\ 0.1847 \\ 0.0518 \\ 0.1408 \\ 0.2304 \end{pmatrix}$$

3.2 Application de la « power method »

Dans le rappel théorique nous avons aussi vu que le calcul du vecteur de scores par la « power method » nécessite plusieurs étapes. Ci-dessous, sont présentes toutes les impressions des matrices qui sont primordiales au calcul du vecteur de scores que la fonction *pageRankPower* retourne.

3.2.1 Impression de la matrice d'adjacence "A"

```
Matrice d'adjacence A :

 [[0 5 0 3 0 0 0 0 0 0]
 [3 0 2 0 1 0 4 0 0 0]
 [0 0 0 4 0 3 0 0 0 0]
 [1 0 0 0 0 2 0 0 0 0]
 [0 0 5 0 0 0 2 0 0 0]
 [0 0 0 0 4 0 0 2 0 0]
 [0 0 0 0 3 0 0 3 0 0]
 [0 0 0 0 0 1 0 0 0 4]
 [0 0 0 3 0 5 0 0 0 2]
 [0 0 0 0 0 0 5 0 4 0]]
```

3.2.2 Impression de la matrice de probabilités de transition "P"

```
Matrice de probabilités de transition P (Normalisée) :

[[0.         0.625      0.         0.375      0.         0.
  0.         0.         0.         0.         0.        ]
 [0.3        0.         0.2        0.         0.1        0.
  0.4        0.         0.         0.         0.        ]
 [0.         0.         0.         0.57142857 0.         0.42857143
  0.         0.         0.         0.         0.        ]
 [0.33333333 0.         0.         0.         0.         0.66666667
  0.         0.         0.         0.         0.        ]
 [0.         0.         0.71428571 0.         0.         0.
  0.28571429 0.         0.         0.         0.        ]
 [0.         0.         0.         0.         0.66666667 0.
  0.         0.33333333 0.         0.         0.        ]
 [0.         0.         0.         0.         0.5        0.
  0.         0.5        0.         0.         0.        ]
 [0.         0.         0.         0.         0.         0.2
  0.         0.         0.         0.8        0.        ]
 [0.         0.         0.         0.3        0.         0.5
  0.         0.         0.         0.2        0.        ]
 [0.         0.         0.         0.         0.         0.
  0.55555556 0.         0.44444444 0.         0.        ]]
```

3.2.3 Impression de la matrice Google "G"

```
Matrice Google G :

[[0.1        0.6625     0.1        0.4375     0.1        0.1
  0.1        0.1        0.1        0.1        0.1       ]
 [0.37       0.1        0.28       0.1        0.19       0.1
  0.46       0.1        0.1        0.1        0.1       ]
 [0.1        0.1        0.1        0.61428571 0.1        0.48571429
  0.1        0.1        0.1        0.1        0.1       ]
 [0.4        0.1        0.1        0.1        0.1        0.7
  0.1        0.1        0.1        0.1        0.1       ]
 [0.1        0.1        0.74285714 0.1        0.1        0.1
  0.35714286 0.1        0.1        0.1        0.1       ]
 [0.1        0.1        0.1        0.1        0.7        0.1
  0.1        0.4        0.1        0.1        0.1       ]
 [0.1        0.1        0.1        0.1        0.55       0.1
  0.1        0.55       0.1        0.1        0.1       ]
 [0.1        0.1        0.1        0.1        0.1        0.28
  0.1        0.1        0.1        0.82       0.1       ]
 [0.1        0.1        0.1        0.37       0.1        0.55
  0.1        0.1        0.1        0.28       0.1       ]
 [0.1        0.1        0.1        0.1        0.1        0.1
  0.6        0.1        0.5        0.1        0.1       ]]
```

3.2.4 Impression des trois premières itérations et du résultat final

```
Iteration 1
[[0.08829211 0.05985526 0.06925113 0.1018494  0.09737895 0.17901639
  0.12089624 0.09672368 0.10113684 0.0856   ]]
Iteration 2
[[0.07721881 0.07877069 0.09124984 0.10143179 0.14063164 0.13196981
  0.099678   0.10953065 0.07065263 0.09886615]]
Iteration 3
[[0.07984085 0.07549241 0.1076762  0.10108739 0.12164544 0.13029718
  0.11260681 0.09707686 0.07344551 0.10083134]]
Resultat final :
[[0.08002792 0.07632406 0.10189765 0.10481852 0.12423788 0.13317042
  0.10951235 0.09959562 0.07311575 0.09729983]]
```

Bibliographie

[1] Amy N. Langville Carl D. Meyer. *Google's pagerank and beyond : the science of search engine rankings.* Princeton, 2006.