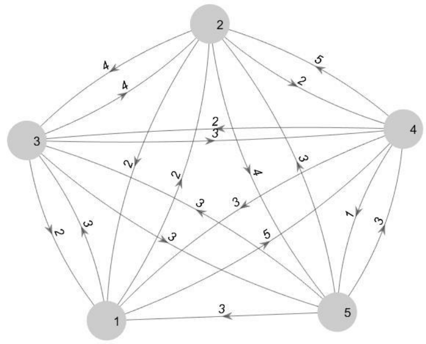
Djafarou Saley Abdou et Elio Pani

Projet SINF 1250 « Ranking de pages web : PageRank » (groupe 3)



Introduction : qu'est-ce que PageRank ?

PageRank est un algorithme de classement de page Internet qui fonctionne en comptant le nombre et la qualité des liens vers une page afin d'estimer l'importance du site en question. L'intuition est que les sites les plus importants sont plus susceptibles de recevoir plus de lien venant d'autres sites.

Matrice d'adjacence du graph orienté sur Moodle :

Pour rappel, dans le cas d'un graph orienté, au sein de la matrice d'adjacence, l'élément a_{ij} désigne le nombre d'arc d'origine i et d'extrémité j. Le terme à l'indice $_{ij}$ compte toujours le nombre de chemins allant de i vers j.

Dans le cadre de ce projet, en plus d'être orienté, le graph est pondéré ce qui nous amène à la matrice d'adjacence d'un graph pondéré. Il s'agit d'une matrice dans laquelle on note à la place du nombre de lien entre 2 nœud, la somme du poids des arcs entre 2 nœuds.

Matrice d'adjacence de notre graph :

$$\mathbf{A} = \begin{pmatrix} 0 & 2 & 3 & 5 & 0 \\ 1 & 0 & 4 & 2 & 4 \\ 2 & 4 & 0 & 3 & 3 \\ 3 & 5 & 2 & 0 & 1 \\ 3 & 3 & 3 & 3 & 0 \end{pmatrix}$$

Dans le cadre de notre programme python, nous avons donné en input la matrice d'adjacence sous forme d'un fichier muni de l'extension « .csv ». Chaque élément a_{ij} de la matrice est séparé par une virgule et les lignes sont représenté en faisant un retour à la ligne. Notre programme a ensuite lis ce fichier pour le convertir en matrice dans python. La lecture du fichier se fait au sein de la méthode main() du programme et utilise les méthodes de la librairie de python destinées à cet usage.

Comment calculer le vecteur de scores PageRank ?

Matrice de probabilité de transition :

Les conditions pour pouvoir déterminer un graph probabiliste est qu'il doit être orienté et pondéré.

La matrice de transition associé à un graph probabiliste d'ordre n est une matrice carrée d'ordre n. Pour trouver le facteur à la position a_{ij} de la matrice de transition, nous allons utiliser la formule du cours :

Formule du cours (slide 123) :

$$\mathrm{P}(page(k+1) = i | page(k) = j) = \frac{w_{ji}}{w_{j.}}$$

$$w_{j.} = \sum_{i=1}^{n} w_{ji}$$

Cette formule décrit : la probabilité de passer d'une page j, en étant à un page i, est la probabilité de passer de la page j à i qui correspond à la pondération de l'arc de j vers i divisé par la somme des probabilités pour passer de la page j à toutes les autres pages i du graph qui correspond à la somme des pondérations de j vers tous les autres pages i. La probabilité de passer de la page j à toutes les autres par i du graph correspond au degré sortant du nœud j.

On réalise cette suite d'opération pour chaque nœud et on obtient la matrice de probabilité de transition où les lignes correspondent aux nœuds de départ et les colonnes aux nœuds d'arrivée.

Matrice de probabilité de transition :

$$\mathbf{P} = \begin{pmatrix} \frac{0}{10} & \frac{2}{10} & \frac{3}{10} & \frac{5}{10} & \frac{0}{10} \\ \frac{1}{11} & \frac{0}{11} & \frac{4}{11} & \frac{2}{11} & \frac{4}{11} \\ \frac{2}{12} & \frac{4}{12} & \frac{0}{12} & \frac{3}{12} & \frac{3}{12} \\ \frac{3}{11} & \frac{5}{11} & \frac{2}{11} & \frac{0}{11} & \frac{1}{11} \\ \frac{3}{12} & \frac{3}{12} & \frac{3}{12} & \frac{3}{12} & \frac{0}{12} \end{pmatrix}$$

L’avant dernière étape consiste à remplir la matrice Google en utilisant la formule :

$$G_{ij} = alpha*P_{ij} + (1\text{-}alpha) *1/N$$

G_{ij} : l’élément à l’indice ij de la matrice Google

Alpha : le facteur de saut “dumping factor” qui nous est donné et qui vaut 0.9

P_{ij} : l’élément à l’indice ij de la matrice de transition

N : le nombre total de nœud de graph (qui correspond au nombre total de pages web)

On obtient :

$$\mathbf{G} = \begin{pmatrix} \frac{1}{50} & \frac{9}{50} & \frac{27}{100} & \frac{9}{20} & 0 \\ \frac{9}{110} & \frac{1}{50} & \frac{18}{55} & \frac{9}{55} & \frac{18}{55} \\ \frac{3}{20} & \frac{3}{10} & \frac{1}{50} & \frac{9}{40} & \frac{9}{40} \\ \frac{27}{110} & \frac{9}{22} & \frac{9}{55} & \frac{1}{50} & \frac{9}{110} \\ \frac{9}{40} & \frac{9}{40} & \frac{9}{40} & \frac{9}{40} & \frac{1}{50} \end{pmatrix}$$

$$G^T = \begin{pmatrix} \frac{1}{50} & \frac{9}{110} & \frac{3}{20} & \frac{27}{110} & \frac{9}{40} \\ \frac{9}{50} & \frac{1}{50} & \frac{3}{10} & \frac{9}{22} & \frac{9}{40} \\ \frac{27}{100} & \frac{18}{55} & \frac{1}{50} & \frac{9}{55} & \frac{9}{40} \\ \frac{9}{20} & \frac{9}{55} & \frac{9}{40} & \frac{1}{50} & \frac{9}{40} \\ 0 & \frac{18}{55} & \frac{9}{40} & \frac{9}{110} & \frac{1}{50} \end{pmatrix}$$

<u>Explications de la formule de la matrice de Google :</u>

La matrice G décrit ceci : La plupart du temps, quelqu'un qui surf sur le net va suivre un lien sur une page : de la page i la personne va suivre les autres liens qui mène au voisin de i. Un petit pourcent du temps, la personne va quitter la page pour en rejoindre une différente, il va s'y « téléporter ». Le facteur P reflète la probabilité que la personne quitte la page actuelle et se "téléporte" vers une nouvelle. Comme il/elle peut se téléporte vers n'importe quelle page, chaque page à 1/N probabilité d'être choisie.

La dernière étape consiste à utiliser la power méthode afin de déterminer le vecteur contenant les page rank de chaque page internet.

$$\boldsymbol{v0 * G^T = v1}$$

v0 : le vecteur initiale de l'itération

Transposée(G) : matrice de Google transposée

v1 : le nouveau vecteur qui servira pour l'itération suivante

Le vecteur v0 s'obtient en additionnant les éléments de la colonne de la matrice de probabilité et les diviser par la somme des éléments de la matrice de probabilité de transition (opération de normalisation). En sachant que la somme des éléments des colonnes doit être égale à zéro.

$$\begin{array}{c}
\begin{pmatrix}
\frac{0}{10} & \frac{2}{10} & \frac{3}{10} & \frac{5}{10} & \frac{0}{10} \\
\frac{1}{11} & \frac{0}{11} & \frac{4}{11} & \frac{2}{11} & \frac{4}{11} \\
\frac{2}{12} & \frac{4}{12} & \frac{0}{12} & \frac{3}{12} & \frac{3}{12} \\
\frac{3}{11} & \frac{5}{11} & \frac{2}{11} & \frac{0}{11} & \frac{1}{11} \\
\frac{3}{12} & \frac{3}{12} & \frac{3}{12} & \frac{3}{12} & \frac{0}{12}
\end{pmatrix} \\
\begin{array}{ccccc} \alpha & \beta & \gamma & \Phi & \lambda \end{array}
\end{array}$$

$\sum \alpha + \sum \beta + \sum \gamma + \sum \Phi + \sum \lambda = 1$ ➔ la somme des colonnes doit être égale à 1, raison pour laquelle on devra diviser cette somme par la somme des éléments de la matrice de probabilité de transition.

Ce qui nous donne comme vecteur initial :

$$v0 = \begin{pmatrix} \frac{1030}{6600} \\ \frac{1634}{6600} \\ \frac{1446}{6600} \\ \frac{1560}{6600} \\ \frac{930}{6600} \end{pmatrix}$$

Nous allons réaliser cette formule itérative jusqu'à convergence. Nous allons définir cette convergence comme étant :

$$|v1 - v0| < gamma$$

gamma : notre marge d'erreur initialisé à 10^{-6}.

Remarque : Nous prenons la valeur absolue qui est par convention utilisée dans les méthodes numériques pour approximer une solution

Pour ce qui est des itérations de la PowerMethod, elles sont réalisées au point 4 des feuilles scannées.

Conclusion

Nous pouvons donc en conclure qu'avec la combinaison de la PowerMethod et l'utilisation de la matrice/formule de Google nous avons su crée un code/algorithme qui classe l'ordre d'importance des pages web sur Internet.

Annexe : Résolution du système d'équation linéaire à la main

Université Catholique de Louvain Louvain-la-Neuve

Pani Elio, 61541400
Saley Abdou Djafarou, 24031600
Le 24 décembre 2017

LSINF1250 : PageRank

① Trouver la matrice d'adjacence de notre graph. L'indice "i" correspondra aux lignes de la matrice (noeud de départ) et l'indice "j" aux colonnes de la matrice (noeud d'arrivée).

$$A = \begin{array}{c} \\ i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{array} \begin{array}{c} \begin{array}{ccccc} j_1 & j_2 & j_3 & j_4 & j_5 \end{array} \\ \begin{bmatrix} 0 & 2 & 3 & 5 & 0 \\ 1 & 0 & 4 & 2 & 4 \\ 2 & 4 & 0 & 3 & 3 \\ 3 & 5 & 2 & 0 & 1 \\ 3 & 3 & 3 & 3 & 0 \end{bmatrix} \end{array}$$

la valeur à l'indice A_{ij} correspond au poid de l'arête.

② Trouver la matrice de probabilité de transition. Je vais utiliser la formule de probabilité conditionnelle du slide 123

$$P(\text{page}(k+1) = j \mid \text{page}(k) = i) = \frac{w_{ij}}{\ } \quad \text{avec } w_{i\cdot} = \sum^{n} w_{ij}$$

« la probabilité de passer d'une page i à une page j est égale à la probabilité de passer de la page i à la page j divisé par la somme probabilité de passer de la page i à n'importe que autre page du graph »

$$P = \begin{array}{c} \\ i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{array} \begin{array}{c} \begin{array}{ccccc} j_1 & j_2 & j_3 & j_4 & j_5 \end{array} \\ \begin{bmatrix} \frac{0}{10} & \frac{2}{10} & \frac{3}{10} & \frac{5}{10} & \frac{0}{10} \\ \frac{1}{11} & \frac{0}{11} & \frac{4}{11} & \frac{2}{11} & \frac{4}{11} \\ \frac{2}{12} & \frac{4}{12} & \frac{0}{12} & \frac{3}{12} & \frac{3}{12} \\ \frac{3}{11} & \frac{5}{11} & \frac{2}{11} & \frac{0}{11} & \frac{1}{11} \\ \frac{3}{12} & \frac{3}{12} & \frac{3}{12} & \frac{3}{12} & \frac{0}{12} \end{bmatrix} \end{array}$$

①

③ Remplissons la matrice de google avec la formule suivante

$$G_{ij} = \alpha^{*} P_{ij} + (1-\alpha)^{*} \frac{1}{N} \quad \text{avec } \alpha = 0{,}8$$

N : nombre de noeud dans le graph

on obtient :

$$G = \begin{bmatrix} \frac{1}{50} & \frac{9}{50} & \frac{27}{100} & \frac{9}{20} & 0 \\ \frac{9}{110} & \frac{1}{50} & \frac{18}{55} & \frac{9}{55} & \frac{18}{55} \\ \frac{3}{20} & \frac{3}{10} & \frac{1}{50} & \frac{9}{40} & \frac{9}{40} \\ \frac{27}{110} & \frac{9}{22} & \frac{9}{55} & \frac{1}{50} & \frac{9}{110} \\ \frac{9}{40} & \frac{9}{40} & \frac{9}{40} & \frac{9}{40} & \frac{1}{50} \end{bmatrix}$$

" j'ai utilisé le logiciel " Matrix calculator " en ligne en implémentant la formule " $0{,}9^{*} P + (1-0{,}9)^{*} \frac{1}{5}$ " ".

④ nous allons implémenter la formule de la powerMethod :

$$G^{*} v_0 = v_1$$

Comment trouver le vecteur v_0 ? Nous allons Normaliser la somme des colonnes de la matrice de probabilité de transition et égaliser l'expression à 1.

$$P = \begin{array}{c} \left[\begin{array}{ccccc} \frac{0}{10} & \frac{2}{10} & \frac{3}{10} & \frac{5}{10} & \frac{0}{10} \\ \frac{1}{11} & \frac{0}{11} & \frac{4}{11} & \frac{2}{11} & \frac{4}{11} \\ \frac{2}{12} & \frac{4}{12} & \frac{0}{12} & \frac{3}{12} & \frac{3}{12} \\ \frac{3}{11} & \frac{5}{11} & \frac{2}{11} & \frac{0}{11} & \frac{1}{11} \\ \frac{3}{12} & \frac{3}{12} & \frac{3}{12} & \frac{3}{12} & \frac{0}{12} \end{array}\right] \begin{array}{c} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{array} \\ \begin{array}{ccccc} j_1 & j_2 & j_3 & j_4 & j_5 \end{array} \end{array}$$

j fixe

une colonne ϕ — une ligne lambda

$$\alpha = \sum_{\phi=1}^{5} \sum_{\lambda=1}^{5} j_{\phi} + i_{\lambda}$$

α correspond à la somme des éléments de chacune des colonnes.

pour que $\alpha = 1$, il faut que l'on divise α par la somme des éléments de la matrice P.

$$\alpha \stackrel{?}{=} 1 \text{ si } \frac{\alpha}{\underbrace{\sum_{i=1}^{5} \sum_{j=1}^{5} P_{ij}}_{\text{la somme de tous les éléments de la matrice}}} = 1$$

Ceci nous amène à dire que chaque élément a_{ij} du vecteur initial v_0 sera un terme de la somme du membre de gauche de l'expression => $\frac{\alpha}{\sum_{i=1}^{5} \sum_{j=1}^{5} P_{ij}}$

Djafarou Saley Abdou et Elio Pani

la somme des éléments de la matrice de probabilité de transit° donne : $\frac{6600}{1320}$

- la somme de la colonne $j1 : \frac{0}{10} + \frac{1}{11} + \frac{2}{12} + \frac{3}{11} + \frac{3}{12} = \frac{1030}{1320}$
- " " " $j2 : \frac{2}{10} + 0 + \frac{4}{12} + \frac{5}{11} + \frac{3}{12} = \frac{1634}{1320}$
- " " " $j3 : \frac{3}{10} + \frac{4}{11} + 0 + \frac{2}{11} + \frac{3}{12} = \frac{1446}{1320}$
- " " " $j4 : \frac{5}{10} + \frac{2}{11} + \frac{3}{12} + 0 + \frac{3}{12} = \frac{1560}{1320}$
- " " " $j5 : 0 + \frac{4}{11} + \frac{3}{12} + \frac{1}{11} + 0 = \frac{930}{1320}$

le vecteur r_0 sera donc

$$r_0 = \begin{bmatrix} \frac{\frac{j1}{6600}}{1320} \\ \frac{\frac{j2}{6600}}{1320} \\ \frac{\frac{j3}{6600}}{1320} \\ \frac{\frac{j4}{6600}}{1320} \\ \frac{\frac{j5}{6600}}{1320} \end{bmatrix} \Rightarrow r_0 = \begin{bmatrix} \frac{1030}{6600} \\ \frac{1634}{6600} \\ \frac{1446}{6600} \\ \frac{1560}{6600} \\ \frac{930}{6600} \end{bmatrix}$$

nous allons maintenant faire 3 itérations avec la power method

A on prend la transposée $\Rightarrow$ $G^T * r_0 = r_1$ (j'utilise le logiciel en ligne "Matrix calculator" pour effectuer les multiplications).

$$\begin{bmatrix} \frac{1}{50} & \frac{9}{110} & \frac{3}{20} & \frac{27}{110} & \frac{9}{40} \\ \frac{9}{50} & \frac{1}{50} & \frac{3}{10} & \frac{9}{22} & \frac{9}{40} \\ \frac{27}{100} & \frac{18}{55} & \frac{1}{50} & \frac{9}{55} & \frac{9}{40} \\ \frac{9}{20} & \frac{9}{55} & \frac{9}{40} & \frac{1}{50} & \frac{9}{40} \\ 0 & \frac{18}{55} & \frac{9}{40} & \frac{9}{110} & \frac{1}{50} \end{bmatrix} * \begin{bmatrix} \frac{1030}{6600} \\ \frac{1634}{6600} \\ \frac{1446}{6600} \\ \frac{1560}{6600} \\ \frac{930}{6600} \end{bmatrix} = \begin{bmatrix} \frac{19267}{132000} \\ \frac{1649243}{7260000} \\ \frac{478979}{2420000} \\ \frac{9509}{48400} \\ \frac{6709}{44000} \end{bmatrix}$$ 1ère itérat°

$$\equiv \quad G^T \quad * \quad r_0 \quad = \quad r_1$$

(2)

$$\begin{bmatrix} \frac{1}{50} & \frac{9}{110} & \frac{3}{20} & \frac{27}{110} & \frac{9}{40} \\ \frac{9}{50} & \frac{1}{50} & \frac{3}{10} & \frac{9}{22} & \frac{9}{40} \\ \frac{27}{100} & \frac{18}{55} & \frac{1}{50} & \frac{9}{55} & \frac{9}{40} \\ \frac{9}{20} & \frac{9}{55} & \frac{9}{40} & \frac{1}{50} & \frac{9}{40} \\ 0 & \frac{18}{55} & \frac{9}{40} & \frac{9}{110} & \frac{1}{50} \end{bmatrix} * \begin{bmatrix} \frac{19267}{132000} \\ \frac{1649243}{7260000} \\ \frac{478979}{2420000} \\ \frac{9509}{48400} \\ \frac{6709}{44000} \end{bmatrix} = \begin{bmatrix} \frac{427173133}{319440000} \\ \frac{3272252947}{15972000000} \\ \frac{986525551}{5324000000} \\ \frac{494413607}{2662000000} \\ \frac{146945341}{10648000000} \end{bmatrix}$$ 2ème itérat°

$$\equiv \quad G^T \quad * \quad r_1 \quad = \quad r_2$$

Djafarou Saley Abdou et Elio Pani

$$\begin{bmatrix} \frac{1}{50} & \frac{9}{110} & \frac{3}{20} & \frac{27}{110} & \frac{9}{40} \\ \frac{9}{50} & \frac{1}{50} & \frac{3}{10} & \frac{9}{22} & \frac{9}{40} \\ \frac{27}{100} & \frac{18}{55} & \frac{1}{50} & \frac{9}{55} & \frac{9}{40} \\ \frac{9}{20} & \frac{9}{55} & \frac{9}{40} & \frac{1}{50} & \frac{9}{40} \\ 0 & \frac{18}{55} & \frac{9}{40} & \frac{9}{110} & \frac{1}{50} \end{bmatrix} * \begin{bmatrix} \frac{427173133}{3194400000} \\ \frac{3272252947}{15972000000} \\ \frac{380525551}{5324000000} \\ \frac{49413607}{266200000} \\ \frac{146945941}{1064800000} \end{bmatrix} = \begin{bmatrix} \frac{790141152617}{6388800000000} \\ \frac{669063913}{351384000000} \\ \frac{1970854407569}{117128000000} \\ \frac{199004134897}{117128000000} \\ \frac{269257780859}{212960000000} \end{bmatrix}$$

$$= \quad G^T \quad * \quad v_2 \quad = \quad v_3$$

Le vecteur v_3 est notre vecteur de score PageRank après 3 itérations

Code Python

Méthode pageRankScore

PageRank.py

```python
import numpy as np

def pageRankScore(mat1, alpha):
    #we initialise en empty matrix (full of zero) that will be the transition probability matrix
    transitionMatrix = np.zeros(shape=(len(mat1),len(mat1)))
    #the loop for the transtion matrix line's
    for line in range (0, len(mat1)):
      #the loop for the transition matrix column's
      for column in range (0,len(mat1)):
        #we apply the matrix probability method to get all the value in the transitionMatrix
        transitionMatrix [line][column] = mat1[line][column]/np.sum(mat1[line])

    #after checking that the matrix was stochaistic (the sum of every line =1), let's do the Google matrix
    #the formula of the Google Matrix : G[i][j]= alpha*transitionMatrix[i][j] + (1-alpha)* (1/N) where N = number of pages/nodes
    googleMatrix = np.zeros(shape=(len(transitionMatrix), len(transitionMatrix)))
    #We apply the formule above
    for line in range (0, len(googleMatrix)):
        for column in range (0, len(googleMatrix)):
            googleMatrix[line][column]= alpha*transitionMatrix[line][column] + (1-alpha)* (1.0/len(googleMatrix))
    #let's now do the powerMethod in order to find the eigen vector of the google matrix which is simply the pageRankScore of each page
    #first , we create the vector that we need to multiply with the google matrix, this vector will be the final pageRank solution
    vectorPageRank = np.zeros(shape=(len(mat1),1))
    #We make the addition of every column in the adjacency matrix and we store the result in a list
    listOfColumnSum = mat1.sum(axis = 0)
    #we fill the initial vector of Pgae rank with the correct value stored in the list listOfColumnSum
    for fillin in range (0, len(listOfColumnSum)):
        vectorPageRank[fillin][0] = listOfColumnSum[fillin]/ np.sum(mat1)
    #Before computing the power method, we initialise an error margin wich will indicates us that the method found a convergence point
    gamma = 0.000001
    #we create a boolean to go out of the loop as soon as we get a good result, I prefer use a boolean instead of a break statement
    finish = False
    while(not(finish)):
        #we store the value of the Page rank vector at the next iteration we take the transpose of the google
        nextVectorPageRank = np.dot(np.transpose(googleMatrix), vectorPageRank)
        #we compare the vector at the previous iteration and at the next iteration to see wheter the solution is stabilized, the method .all() compare every elements of each ve
        if (abs(nextVectorPageRank - vectorPageRank < gamma )).all():
            #we change the boolean of the while loop
            finish = True
            #we return the last iteration with the highest accuracy
            print (nextVectorPageRank)
        #if the error degree stills to big, we go for an other iteration
        vectorPageRank = nextVectorPageRank
```

Méthode main

```python
def main():
 # we open the file with it's location, in this case, it's in the same folder as the code , no need to print the path
 fichier = open("matrice-adjacence.csv")
 #we are going to read the first line of the document
 contenu = fichier.readline()
 # the dimension of the matrix according to the previous line (we divide it by 2 because the comas are counted)
 dimOfMatrix = int(len(contenu)/2)
 # we initialise a empty matrix (full of zeros)
 matrice = np.zeros(shape=(dimOfMatrix,dimOfMatrix))
 # we create a list for a later use which will count 2 by 2 until the end of the line which is read (to avoid the comas)
 advance = list(range(0,len(contenu),2))
 # this variable will help us to keep track of the matrix line that we are filling
 line = 0
 # that will be our stop condition, whend the variable "contenu" refers to an empty string, that mean that we've reached the end of the file
 while not(contenu ==""):
  # we set up the column to the matrix at zero at each iteration to be able to fill the matrix iteratively
  column = 0
  for j in advance:
       # we fill the matrix with the elements inside "contenu", j grows 2 by 2 because we don't need comas in our matrix!
       matrice[line][column]=float(contenu[j])
       #we increment the column to go to the next one
       column +=1
  line +=1
  # that's the statement that will continue or not the loop
  contenu = fichier.readline()
 # to avoid possible future problems, we have to close the file
 fichier.close()
 pageRankScore(matrice, alpha=0.9)

main()
```