TP2 – Le pagerank

M2 informatique – Université Paris Diderot

Année 2020-2021

Matrices initiales

Soit G un graphe orienté dont les sommets sont numérotés de 0 à $n-1$. La matrice d'adjacence de G est la matrice A^0 carrée de taille n telle que :

— $A^0_{i,j} = 0$ si (i,j) n'est pas un arc de G ;

— et si (i,j) est un arc alors $A^0_{i,j} = 1/d_i$ (où d_i désigne le degré sortant du sommet i).

Rappel : cette matrice creuse est représentée sous forme « CLI ».

On ne travaillera pas directement avec A^0, car elle peut contenir des lignes de zéros correspondant aux sommets sans arc sortant. Nous considérerons donc la matrice A stochastique équiprobable associée à G, carrée de taille n, telle que :

— si le sommet i n'a pas de voisins sortants, alors $A_{i,j} = 1/n$ pour tout j ;

— sinon :

· $A_{i,j} = 0$ si (i,j) n'est pas un arc de G,

· et si (i,j) est un arc alors $A_{i,j} = 1/d_i$ (où d_i désigne le degré sortant du sommet i).

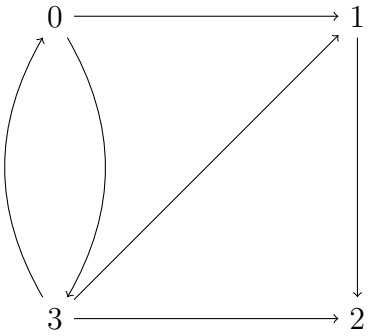
On trouvera un exemple à la figure 1. Par rapport à A^0, on a remplacé les lignes de coefficients nuls par des lignes de coefficients $1/n$. Pour représenter A, on utilisera simplement la représentation CLI de A^0 (la ligne i est vide ssi $L[i] = L[i+1]$).

Produit matrice-vecteur

Exercice 1 *Transposée*

Si M est une matrice creuse (donnée sous forme CLI) et V un vecteur, nous voulons faire le calcul non pas de $P = MV$ (*cf.* TP 1) mais de $P = ({}^tM)V$. Or transposer une matrice est coûteux. Nous allons évaluer $({}^tM)V$ **sans calculer explicitement** la transposée de M :

$$P[i] = \sum_{j=0}^{j=n-1} M_{ji} V[j].$$



$$\begin{pmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \end{pmatrix}$$

FIGURE 1 – Un exemple de graphe orienté avec sa matrice A associée.

Calculer $P[i]$ revient à parcourir une colonne de M, peu efficace en représentation CLI. Mais si on parcourt la **ligne** i on peut faire $P[j]$ += $M_{ij}V[i]$ en ayant initialisé P au vecteur nul. On fait m fois de suite ces incréments.

1. Programmer cela, en parcourant donc la matrice M ligne par ligne. Le calcul doit **impérativement** se faire en $O(n+m)$ et en une seule passe des tableaux L, C et I.
2. Modifier votre code pour prendre en compte la liste des lignes « vides » remplacées par des coefficients $1/n$ (matrice A plutôt que A^0).

Pagerank

Soit J la matrice carrée de taille n dont tous les coefficients sont 1. Pour le *pagerank*, nous allons utiliser la matrice

$$A_G = (1-\varepsilon)A + (\varepsilon/n)J,$$

où $\varepsilon \in]0,1[$ est un réel bien choisi (de l'ordre de 1/7). Cela permet d'obtenir une matrice stochastique dont les coefficients sont tous strictement positifs, et garantit ainsi l'existence et l'unicité du vecteur que l'on cherche (théorème de Perron-Frobenius).

Exercice 2
Donner la matrice A_G pour le graphe de la figure 1.
Attention, la matrice A_G n'est plus creuse. Pour représenter A_G, on se servira de la matrice creuse A^0 (plus précisément de la matrice A dont la représentation est la même, cf. ci-dessus) et du réel ε.

Rappelons maintenant l'algorithme de pagerank.

Données : une matrice A_G, une distribution de probabilités Π_0 et un entier k.
Résultat : le vecteur Π de pagerank avec une certaine précision.
début
  $\Pi \leftarrow \Pi_0$;
  pour *i de 1 à k* **faire**
    $\Pi \leftarrow ({}^tA_G)\Pi$
  fin pour
  renvoyer Π
fin

Exercice 3 *Pagerank*

1. En vous servant de l'exercice 1, écrire un programme qui demande un sommet de départ et un nombre de pas et affiche, étape par étape, la probabilité d'être sur les différents sommets selon la matrice A_G. Vérifier pour le graphe de la figure 1 que vous obtenez les bonnes valeurs.

   Attention à la représentation de la matrice A_G. La ligne $\Pi \leftarrow ({}^tA_G)\Pi$ de l'algorithme doit être calculée grâce à

   $$\Pi \leftarrow (1-\varepsilon)({}^tA)\Pi + \varepsilon\mathbb{1}$$

   où $\mathbb{1}$ est le vecteur dont tous les coefficients sont égaux à 1.
2. Programmer l'algorithme du pagerank en partant de la distribution Π_0 uniforme sur les n sommets.
3. Tester votre algorithme sur des graphes bien choisis.

Application

Exercice 4 *Poids des pages*

1. Choisir (en expliquant votre choix) la valeur de ε et le nombre d'itérations k, et faire tourner l'algorithme de pagerank (exercice 3) sur le graphe des pages collectées au TP 1.
2. Vérifier sur quelques pages arbitraires que le résultat semble cohérent.
3. On a ainsi obtenu le pagerank de chaque page. Enregistrer le résultat sur le disque.