TP2 – Le pagerank

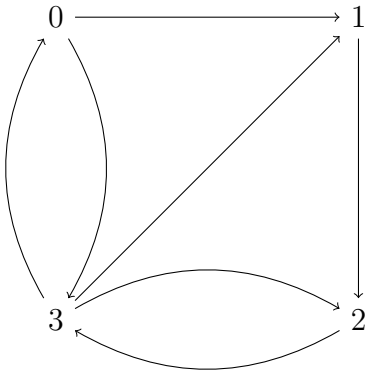
M2 informatique – Université Paris Diderot

Année 2018-2019

Soit G un graphe orienté fortement connexe et apériodique (le pgcd de la taille de ses circuits vaut 1). La matrice stochastique équiprobable associée à G est une matrice M telle que :

— $M_{i,j} = 0$ si (i, j) n'est pas un arc de G ;
— et si (i, j) est un arc alors $M_{i,j} = 1/d^+(i)$ (où $d^+(i)$ désigne le degré sortant du sommet i).

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



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

FIGURE 1 – Un exemple de graphe orienté fortement connexe et apériodique, avec sa matrice associée.

Produit matrice-vecteur

Exercice 1 *Transposée*
Nous voulons faire le calcul non pas de $P = MV$ (*cf.* TP 1) mais de $P = {}^tMV$. Or transposer une matrice est coûteux. Nous allons évaluer ${}^tMV$ **sans calculer explicitement** la transposée de M :

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

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.

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.

Pagerank-zéro

Rappelons l'algorithme standard de pagerank (pagerank-zéro).

Données : une matrice M, une distribution de probabilités Z et un réel $\epsilon > 0$.
Résultat : le vecteur P de pagerank-zéro avec une précision ϵ.
début
  $P_0 = Z$;
  répéter
    $P_{k+1} = {}^tMP_k$;
    $\delta = \|P_{k+1} - P_k\|_1$;
  jusqu'à $\delta < \epsilon$;
fin

Exercice 2 *Pagerank-zéro*

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. Vérifier pour le graphe de la figure 1 que vous obtenez les bonnes valeurs.
2. En déduire un algorithme qui calcule le pagerank-zéro, en partant du sommet 0 (on a donc $Z = (1, 0, ...0)$) et à la précision ϵ.
3. Tester votre algorithme sur des graphes bien choisis.

Pagerank avec facteur zap

Dans le modèle du surfeur (avec facteur zap $d \in [0, 1]$), celui-ci peut choisir :
— soit de suivre un lien sortant avec un probabilité $(1 - d)$;
— soit « zapper » sur une page aléatoire avec probabilité d.
On propose de prendre le facteur zap d compris entre 0,1 et 0,2.

Exercice 3 *Pagerank avec zap*

1. Programmer la variante du pagerank incluant le zap :

$$P_{k+1}[i] = \frac{d}{n} + (1-d)\sum_{j=0}^{n-1} M_{j,i} P_k(j).$$

2. Tester votre algorithme sur des graphes bien choisis.

Application

Exercice 4 *Poids des pages*

1. Choisir (en expliquant votre choix) le facteur zap d et la précision ϵ, et faire tourner l'algorithme de pagerank avec facteur zap (exercice 3) sur le graphe des pages collectées au TP 1.
2. Vérifier sur quelques pages arbitraires que le résultat semble cohérent.
3. Enregistrer le résultat sur le disque.

Exercice 5 *Tri*

1. Pour chaque mot, éliminer les pages pour lesquelles la fréquence de ce mot est trop faible (déterminer le seuil approprié).
2. Dans la relation mots-pages, trier par ordre de pagerank décroissant la liste des pages associées à chaque mot.
3. Enregistrer le résultat sur le disque.