Projet LINFO1114 – Mathématiques discrètes
« Ranking de réseaux sociaux et pages web: PageRank »

Professeur	Marco Saerens marco.saerens@uclouvain.be
Bureau	b.102
Adresse	Université catholique de Louvain Place des Doyens 1 1348 Louvain-La-Neuve Belgique
Assistants	Sylvain Courtain sylvain.courtain@uclouvain.be Pierre Leleux p.leleux@uclouvain.be
Date	28 Mars 2022

Objectif : Le but de ce projet est d'implémenter et de tester un algorithme permettant de classer (to rank) les noeuds d'un graphe en assignant un score d'importance à chacun de ces noeuds. Cet algorithme est celui de PageRank avec téléportation et vecteur de personnalisation. Vous travaillerez par groupe de *trois* étudiantes et étudiants impérativement (merci de vous inscrire dans un groupe sur Moodle). Chaque groupe trouvera un vecteur de personnalisation différent sur Moodle, sur lequel il devra se baser pour calculer les scores PageRank. La résolution doit être détaillée dans le rapport en utilisant deux méthodes différentes (1) en résolvant un système d'équations linéaires en Python (voir slides du cours) et (2) en implémentant la "power method" en Python à l'aide de Numpy (bibliothèque Python pour le calcul scientifique). Il faudra suivre rigoureusement ces algorithmes, comme détaillé au cours. Vous trouverez toutes les informations utiles dans les slides du cours mais aussi dans des ouvrages de "link analysis" (par exemple l'ouvrage *Google's PageRank and Beyond* de Amy Langville et Carl Meyer).

Méthode (Python 3) :
Dans le cadre de l'implémentation de la résolution du système d'équations linéaires, la signature de la méthode est :

```
def pageRankLinear(A : np.matrix, alpha : float, v : np.array) -> np.array
```

- **Input :** Une matrice d'adjacence [1] **A** d'un graphe dirigé, pondéré et régulier G, un vecteur de personnalisation **v**, ainsi qu'un paramètre de téléportation α compris entre 0 et 1 (0.9 par défaut et pour les résultats à présenter). Toutes ces valeurs sont non-négatives.
- **Output :** Un vecteur **x** contenant les scores d'importance des noeuds ordonnés dans le même ordre que la matrice d'adjacence.

Dans le cadre de l'implémentation de la power method, la signature de la méthode est :

```
def pageRankPower(A : np.matrix, alpha : float, v : np.array) -> np.array
```

- **Input :** Une matrice d'adjacence **A** d'un graphe dirigé, pondéré et régulier G, un vecteur de personnalisation **v**, ainsi qu'un paramètre de téléportation α compris entre 0 et 1, $\alpha \in]0,1[$ (0.9 par défaut et pour les résultats à présenter).
- **Output :** Un vecteur **x** contenant les scores d'importance des noeuds ordonnés dans le même ordre que la matrice d'adjacence.

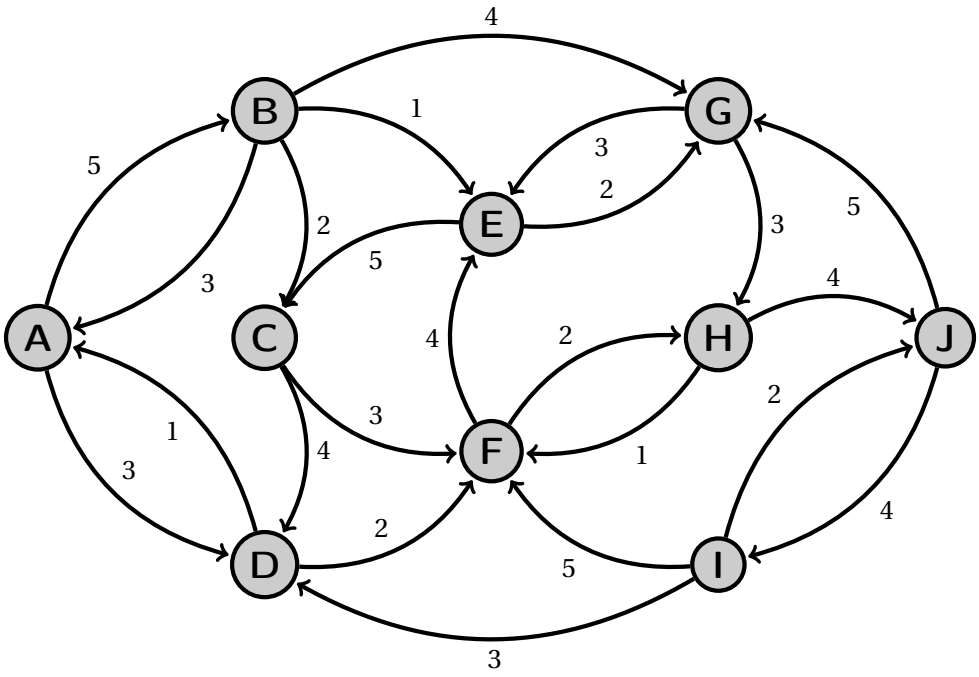
Vous devez donc calculer les scores PageRank de deux façons différentes :
- En Python, en résolvant un système d'équations linéaires. Pour cette technique, vous pouvez utiliser les fonctions numpy permettant de résoudre un système d'équations linéaires.
- En Python, en calculant le vecteur propre dominant de gauche de la matrice Google, en utilisant la "power method". Pour cette technique, vous devez implémenter vous-même la power method, et donc la boucle permettant de calculer le vecteur propre dominant de gauche de la matrice.

Comme nous considérons une téléportation *non-uniforme*, vous devrez calculer la matrice Google à partir de la matrice de probabilités de transitions du graphe et un vecteur de personnalisation non-uniforme. Notez que, dans l'algorithme basé sur la "power method", il faudra initialiser les scores par le vecteur de personnalisation.
Ainsi que déjà mentionné, évitez d'utiliser les fonctions permettant de calculer directement des vecteurs propres car le but du projet est justement de comprendre et d'implémenter la power method. Donc, dans ce cadre, vous ne pouvez utiliser cette librairie que pour effectuer des opérations matricielles simples telles que la transposée, la multiplication matricielle, etc. Par contre, pour la résolution à l'aide d'un système d'équations, nous vous autorisons à utiliser des fonctions spécialisées.

1. Aussi parfois notée **W** dans les slides du cours.

De plus, il vous est demandé d'ajouter à votre code une méthode "main" qui lit un fichier csv contenant la matrice d'adjacence **A** du graphe présenté ci-dessous (séparez vos valeurs par une virgule et chaque ligne de la matrice par un passage à la ligne), qui exécute le calcul de PageRank (via vos méthodes pageRankPower et pageRankLinear) et imprime les résultats.

Données : Un graphe dirigé et pondéré représentant un réseau de 10 noeuds.



Ainsi qu'un vecteur de personnalisation propre à chaque group disponible sur Moodle dans le dossier "Vecteur".

Rapport et code : Le rapport est un fichier PDF (8 pages maximum ; écrit en LaTeX) qui se compose de :

- Un rappel théorique expliquant brièvement comment calculer le vecteur de scores PageRank (en résolvant un système d'équations linéaires et en implémentant la power method). Discuter également l'impact du paramètre de téléportation α et du vecteur de personnalisation **v** sur le score.
- La présentation numérique du système d'équations linéaires permettant de calculer le score PageRank du graphe ci-dessus avec le vecteur de personnalisation qui vous a été assigné.
- Dans le cadre de l'implémentation en Python de la power method, l'impression :
  — de la matrice d'adjacence **A**,
  — de la matrice de probabilités de transition **P**,
  — de la matrice Google **G**,
  — des trois premières itérations de la power method (vecteur de scores) et

— du résultat final, c'est à dire le score PageRank final après convergence.
- Le code complet en annexe – n'oubliez pas de bien commenter ce code!

Attention, respectez bien ces consignes car nous nous baserons sur celles-ci pour calculer la note du projet de manière semi-automatique.

Langage de programmation : L'implémentation devra impérativement être codée en Python 3 en respectant les consignes et les signatures énoncées ci-dessus.
Comme déjà mentionné, vous devez utiliser une librairie Python externe de calcul et de manipulation matricielle/vectorielle nommée numpy. Cette librairie vous évitera d'implémenter la multiplication matricielle, ou la transposition, de manière à vous concentrer sur l'algorithme proprement dit.

Evaluation et consignes : Le projet est **obligatoire** et à réaliser par groupes de trois étudiantes et étudiants. L'évaluation portera sur le contenu du rapport (maximum 8 pages) et le code (lisibilité, structure, **commentaires**,...) et comptera pour 4 points sur 20 dans la note finale (le reste des points étant donné par l'examen écrit). Le rapport doit être très professionnel[2], du type article scientifique ou technique, et doit contenir les références sur lesquelles vous vous êtes basées.
Les différents fichiers, c'est-à-dire les fichiers de code source, le fichier .csv et le rapport en pdf, tous compressés ensemble (nom du fichier compressé : "groupe" suivi du numéro du groupe (deux digits), et suivi par les noms de famille des membres du groupe séparés par des underscores et par ordre alphabétique ; par exemple "groupe05_Courtain_Leleux_Saerens"[3]), sont à remettre sur Moodle au plus tard le dernier jour avant le début du blocus de la session de juin (dimanche 15 mai 2022), avant 23h55. Si vous rendez le projet en retard, nous retirons 1 point sur 20 (note du projet) plus 1 point par jour de retard. Par exemple, si vous le rendez à 23h58 le jour de la deadline, vous aurez $-1/20$. Si vous le rendez le lendemain, ce sera $-2/20$. La note sera la même pour tous les membres du groupe.

Bon travail!

---

2. Par exemple pas de copier/coller d'images de formule mathématique ou algorithme.
3. Nous ne corrigeons pas les projets qui ne respectent pas cette consigne : vous devrez re-soumettre le projet avec pénalités.