Projet SINF 1250
« Une application de PageRank dans un contexte de recommendation: ItemRank »

Professeur	Marco Saerens marco.saerens@uclouvain.be
Téléphone	010 47 92 46
Bureau	b.108
Adresse	Université catholique de Louvain Place des Doyens 1 1348 Louvain-La-Neuve Belgique
Assistants	Pierre Leleux p.leleux@uclouvain.be Sylvain Courtain sylvain.courtain@uclouvain.be
Date	03 Novembre 2018

Objectif :
Dans le cadre de ce projet, l'algorithme qu'il vous est demandé d'implémenter est l'ItemRank (aussi appelé "Random Walk With Restart"), un algorithme de marche aléatoire ayant pour objectif de classer des produits en leur associant un score d'importance. Pour ce faire, vous travaillerez à partir d'un graphe dans lequel chaque noeud représente un produit (item en anglais) et où un arc liant deux noeuds représente un lien entre ces deux produits. Le score obtenu pour chaque produit vous permettra de déterminer les produits à recommander à un utilisateur précis.

Vous travaillerez par groupe de deux étudiants (merci de vous inscrire dans un groupe sur Moodle). Chaque groupe trouvera un graphe différent sur Moodle, ainsi que la liste des produits consommés par un utilisateur précis. Il vous est demandé de calculer le score associé à chaque produit pour l'utilisateur qui vous a été fourni.

Principe de l'algorithme :
L'ItemRank se base sur le principe de marche aléatoire avec téléportation. Il consiste à démarrer la marche aléatoire sur un produit choisi au hasard parmi ceux consommés par l'utilisateur, puis d'entammer une marche aléatoire sur le graphe entre produits. A chaque moment t, le marcheur a une probabilité α de continuer la marche aléatoire et une probabilité $(1-\alpha)$ d'être retéléporté au hasard sur l'un des produits déjà consommés par l'utilisateur. Les produits recommandés seront ceux associés aux noeuds qui sont visités le plus souvent par le marcheur.

Le calcul de $\vec{x}_u$, le vecteur de score d'importance des produits pour un utilisateur u s'obtient en partant de $\mathbf{P}$, la matrice de probabilité de transition du graphe des produits, et $\vec{v}_u$, le vecteur de personnalisation de l'utilisateur, défini comme $[\vec{v}_u]_i = \frac{1}{n}$ si l'utilisateur a consommé

le produit i et 0 sinon. Le terme n représente le nombre de produits différents achetés par cet utilisateur.

Calcul du vecteur de score :
Plusieurs méthodes existent afin de calculer le vecteur de score. La méthode qu'il vous est demandé d'utiliser dans le cadre de ce rapport est le calcul par récurrence. Son principe est le suivant :

Initialiser $\vec{x}_u(0) = \vec{v}_u$.

Répeter :

$$\vec{x}_u(t+1) = \alpha \mathbf{P}^{\mathrm{T}} \vec{x}_u(t) + (1-\alpha)\vec{v}_u$$

jusqu'à convergence du vecteur $\vec{x}_u$.

Missions :
Ecrivez une méthode calculant les scores par récurrence comme expliqué ci-dessus, puis utilisez-la pour calculer le score de chaque produit pour l'utilisateur en cours de traitement. Dans le cadre de ce projet, considérez une forte probabilité de retéléportation vers les produits consommés (85%).

Le score pour $t \to \infty$ peut être obtenu directement en utilisant une inversion matricielle comme suit :

$$\vec{x}_u(\infty) = (1-\alpha)(\mathbf{I} - \alpha \mathbf{P}^{\mathrm{T}})^{-1} \vec{v}_u$$

où **I** est la matrice identité de même dimension que **P**.

Calculez $\vec{x}_u(\infty)$ en Python puis vérifiez empiriquement (en imprimant les deux valeurs à l'écran et en les comparant) la convergence du vecteur de score vers cette valeur à l'aide de votre méthode de récurrence.

Méthode (Python 3) :
Il vous est demandé de fournir un code avec l'implémentation des deux approches de calcul du vecteur de score en une seule méthode, la signature de la méthode étant :

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

- **Input :** Une matrice d'adjacence **A** d'un graphe simple, un paramètre de téléportation α compris entre 0 et 1, le vecteur de personnalisation d'un utilisateur "v" (fourni sur moodle) et une variable booléenne "m" contenant `true` si le score doit être obtenu par récurrence et `false` s'il doit être obtenu par inversion matricielle.
- **Output :** Un vecteur **x** contenant les scores d'importance des noeuds ordonnés dans le même ordre que la matrice d'adjacence.

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** de l'image de graphe fournie sur Moodle ainsi que le fichier csv contenant le vecteur ligne de personnalisation de l'utilisateur (ce fichier est fourni sur Moodle – il vous suffit donc de le reprendre), qui lance le calcul d'ItemRank (en utilisant l'approche de votre choix) et qui imprime les résultats à l'écran.

Données :
Les données qui sont attribuées à votre groupe sont disponibles dans le dossier "Documents Projet" sur Moodle. **Veillez à bien télécharger les données spécifiquement associées à votre groupe.**

— Une image de réseau simple représentant un graphe entre 10 produits ;
— Un fichier texte contenant un vecteur ligne de taille 10 contenant des chiffres binaires (0 et 1) indiquant les produits achetés par l'utilisateur (1 si l'utilisateur a acheté le produit i, 0 sinon). Il s'agit d'un fichier texte csv d'une seule ligne.

Rapport :
Le rapport est un fichier PDF (6 pages maximum ; écrit en LaTeX). Dans celui-ci :

- Présentez rapidement l'algorithme et expliquez sa logique sous-jacente. En vous basant sur la formule, discutez l'impact du paramètre α sur les scores.
- Listez les inputs et outputs de votre méthode permettant de calculer les scores par récurrence, en présentant :
  — la matrice d'adjacence **A**,
  — le vecteur de personnalisation $\overrightarrow{v}_u$,
  — la matrice de probabilités de transition **P**,
  — les premières itérations de l'approche par récurrence jusqu'à $\overrightarrow{x}_u(3)$ inclu,
  — votre vecteur de score final (après convergence).
- Vérifiez l'exactitude de votre convergence en comparant votre résultat avec celui obtenu via inversion matricielle.
- Finalement, sur base de vos résultats, quel produit recommanderiez-vous à l'utilisateur. Justifiez votre choix.

N'oubliez pas de placer **le code complet et commenté** dans un fichier annexe Python (.py). Par ailleurs, n'oubliez pas non plus de citer vos références bibliographiques en utilisant les normes adéquates.

Langage de programmation :
L'implémentation devra impérativement être codée en Python 3 en respectant les consignes et la signature décrite 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 les opérations matricielle (par exemple la multiplication et l'inversion) de manière à vous concentrer sur l'algorithme ItemRank. Vous pouvez utiliser d'autres librairies pour certains traitements annexes (dessiner le graphe par exemple), mais pas pour implémenter l'algorithme bien sûr.

Evaluation et consignes :
Le projet est à réaliser par groupes de deux étudiants. L'évaluation portera sur le contenu du rapport (maximum 6 pages) et le code (lisibilité, structure, **commentaires**,...) et comptera pour 2 points sur 20 dans la note finale (le reste des points étant donné par l'examen écrit). Les fichiers, c'est-à-dire le rapport pdf, l'unique fichier de code source, le csv de votre matrice d'adjacence ainsi que le csv du vecteur de personnalisation fourni sur Moodle, compressés ensemble (nom du fichier compressé : numéro du groupe suivi par les noms de famille des membres du groupe séparés par des underscore ; par exemple "groupe05_Leleux_Courtain" [1]), sont à remettre sur Moodle au plus tard le mardi 18 décembre 2018, avant 23h55. Si vous rendez le projet en retard, nous retirons 1 point sur 20 plus 1 point par jour de retard. Par exemple, si vous le rendez à 23h58 le jour de la deadline, vous aurez une pénialité de $-1/20$. Si vous le rendez le lendemain, ce sera $-2/20$. La note sera la même pour tous les membres du groupe.
Respectez scrupuleusement ces consignes car nous utiliserons des programmes automatisés pour vérifier vos résultats.

Modalités liées au travail :
Le présent projet est **obligatoire**. Un échec absorbant est présent sur la partie travail à partir du seuil de 6/20. Cela signifie que votre note globale pour ce cours sera calculée comme suit (Examen et Travail sont des notes sur 20) :

$$\text{TotalCours} = \begin{cases} 0.9 * \text{Examen} + 0.1 * \text{Travail} & \text{si Travail} > \frac{6}{20} \\ \min(\text{Travail}, \text{Examen}) & \text{sinon} \end{cases}$$

Ne pas remettre le travail signifie donc obtenir 0 à celui-ci et, par conséquent, 0 pour ce cours. Notez bien dans la formule précédente que l'échec absorbant n'est présent que sur le travail.

Concernant la seconde session :

- Vous pouvez refaire le travail pour la session d'août si vous l'avez raté (note strictement inférieure à 10/20), et il faudra d'office représenter l'examen écrit.
- Le principe d'échec absorbant sur la note du travail n'est pas modifié pour la session d'août.
- Suivant la note obtenue pour le travail en janvier :
  - Si la note de votre travail était inférieure ou égale à 6/20, vous devrez surement refaire le travail, suite à l'échec absorbant.
  - Si la note de votre travail était supérieure ou égale à 10/20, celle-ci est automatiquement reportée – vous ne pouvez donc plus refaire le travail si vous l'avez réussi.
  - Entre les deux ($6 < \text{Travail} < 10$), c'est au choix mais il faut prévenir le professeur si vous décidez de refaire le travail (sinon nous reporterons la note de janvier).

Bon courage!

1. Attention, nous ne corrigeons pas les projets qui ne respectent pas cette consigne : vous devrez resoumettre le projet avec pénalités.