

INF 435

TP 3 de graphes

Yannis Haralambous (Télécom Bretagne)

Dans ce TP nous allons utiliser des données récupérées sur un dépôt de données pour machine learning (<https://archive.ics.uci.edu/ml/datasets/Movie>) que nous avons nettoyées et épurées (du moins jusqu'à un certain point).

Il s'agit de données sur des films et des acteurs :

1. le fichier `movies.tsv` contient les identifiants, titres, années et réalisateurs de 11 343 films ;
2. chaque ligne du fichier `casts.tsv` correspond à une apparition d'un acteur dans un film, elle est structurée de la manière suivante : identifiant du film, titre du film, nom de l'acteur. Il contient 44 180 tels identifiants.

À partir de ces données (et uniquement ces données !) on essaiera de trouver les acteurs, les réalisateurs et les films les plus importants.

Nos hypothèses de travail seront les suivantes :

- un acteur est important s'il joue dans plusieurs films avec d'autres acteurs importants ;
- un film est important s'il a un casting d'acteurs importants ou s'il a été réalisé par un réalisateur important ;
- un réalisateur est important s'il a fait des films importants.

Nous allons essayer de modéliser ces hypothèses de travail à l'aide de la théorie de graphes.

0.1 Données restreintes

Les données décrites ci-dessous sont assez volumineuses, il serait plus pertinent de n'en utiliser qu'une petite partie pendant la phase de programmation et de débogage. Nous mettons donc à votre disposition deux autres jeux de données :

1. `movies1990.tsv` et `casts1990.tsv`, les films de la seule année 1990 (parmi lesquels des succès comme *Total Recall*, *Robocop 2*, *Arachnophobia* ou *Le Parrain 3*, des films délicieux comme *Alice* de Woody Allen ou *Le Cuisinier, le voleur, sa femme et son amant* de Greenaway, sans oublier *La Nuit des morts-vivants* de George A. Romero). En tout, on y trouve 251 films et 1 016 casts ;
2. `movies1990s.tsv` et `casts1990s.tsv`, les films de toute la décennie quatre-vingts-dix (2 094 films, 8 154 casts).

Selon la puissance de votre machine et votre degré de patience, nous vous conseillons d'utiliser l'un ou l'autre jeu de données restreintes pour programmer et déboguer votre programme. Une fois celui-ci opérationnel, vous pourrez l'appliquer à la totalité des données.

1 Étape 0 : installer igraph, l'apprivoiser

Avant d'installer igraph vérifier que vous n'avez pas déjà la dernière version (0.7.1 en février 2016) :

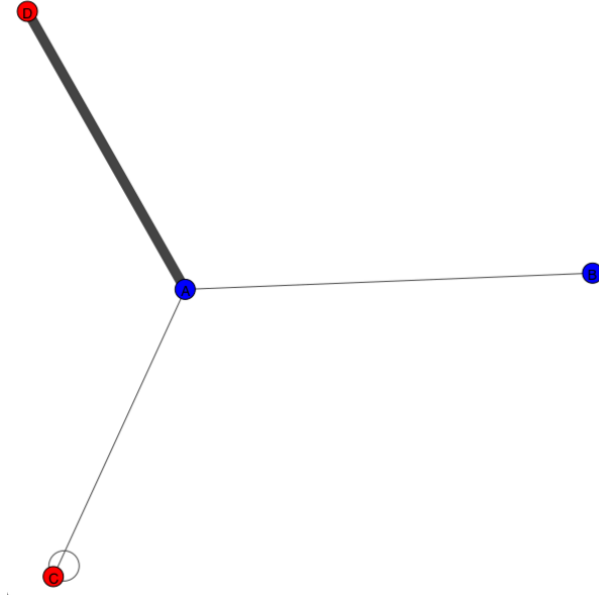
```
python
>>> import igraph
>>> print igraph.__version__
```

Le cas échéant, l'installer — ou le mettre à jour — de la manière suivante :

```
pip install --upgrade python-igraph --user
```

Vous trouverez un tutoriel d'igraph à l'adresse suivante : <http://igraph.org/python/doc/tutorial/tutorial.html>

Pour vous entraîner, créer sous igraph le graphe suivant :



où les sommets sont A, B, C, D , les deux premiers étant bleus et les deux autres rouges, l'arête AD a une épaisseur de 10 pixels et les autres une épaisseur de 1 pixel. (Voir la solution en fin d'énoncé.)

2 Étape 1 : lecture des données

Créer deux graphes :

1. un graphe **A** dont les sommets sont des acteurs. On trace une arête entre deux acteurs s'ils ont joué ensemble dans au moins un film. Le poids $w_{\text{act},1}$ de cette arête est le nombre de films où ils ont joué ensemble ;
2. un graphe **F** dont les sommets sont les films. On trace une arête entre deux films s'ils ont au moins un acteur en commun. Le poids $w_{\text{film},0}$ de cette arête sera défini dans § 3.2.

3 Étape 2 : première itération

3.1 Centralité des acteurs

Calculer la centralité PageRank $C_{\text{act},1}$ des acteurs du graphe **A** avec le poids $w_{\text{act},1}$. Quels sont les dix acteurs les plus centraux ?

On va normaliser $C_{\text{act},1}$ en calculant

$$NC_{\text{act},1}(a) := \frac{C_{\text{act},1}(a) - \min C_{\text{act},1}}{\max C_{\text{act},1} - \min C_{\text{act},1}}, \text{ pour tout acteur } a.$$

3.2 Centralité des films

Ensuite on va utiliser la centralité des acteurs pour pondérer le graphe \mathbf{F} : au lieu de compter chaque acteur ayant joué dans les deux films représentés par les extrémités de l'arête, comme une seule unité, on va prendre la somme des centralités normalisées $NC_{\text{act},1}$ des acteurs. Cela nous donnera un poids $w_{\text{film},1}$ pour les arêtes de \mathbf{F} .

Calculer la centralité normalisée PageRank $NC_{\text{film},1}$ des films pour le poids $w_{\text{film},1}$.

Quels sont les dix films les plus centraux ?

3.3 Pondération des réalisateurs

Enfin, on va utiliser la centralité des films $C_{\text{film},1}$ pour obtenir un poids moyen de réalisateur : $p_{\text{réal},1}$ sera la moyenne des centralités normalisées $NC_{\text{film},1}$ des films réalisés par le même réalisateur.

3.4 Nouvelle centralité des films

On va faire en sorte que la centralité (= importance) d'un réalisateur affecter tous ses films, cela aura comme résultat de «lisser» un peu les centralités des films (un film est central par sa position dans le graphe mais aussi par le fait qu'il a été réalisé par un réalisateur central). On définit donc une centralité enrichie de film

$$EC_{\text{film},1}(f) := p_{\text{réal},1}(r) \times NC_{\text{film},1}(f)$$

où r est le réalisateur du film f .

À la fin de cette étape, nous disposons maintenant d'une centralité (normalisée) des acteurs ($NC_{\text{act},1}$) et d'une centralité enrichie des films ($EC_{\text{film},1}$).

4 Étape 3 : itérations suivantes

On va re-calculer plusieurs fois les centralités des acteurs et des films (qui s'influencent mutuellement), et comparer à chaque fois la liste des dix acteurs les plus centraux.

En § 3.1 on a pondéré le graphe \mathbf{A} en comptant une unité pour chaque film dans lequel deux acteurs jouent ensemble. Nous allons maintenant utiliser comme pondération la centralité enrichie des films.

Le même calcul de pagerank nous donnera une nouvelle centralité des acteurs.

Qui à son tour nous donnera une nouvelle centralité des films.

Et une nouvelle pondération des réalisateurs.

Et une nouvelle centralité enrichie des films.

Et donc des nouvelles listes des dix acteurs les plus centraux, des dix films les plus centraux et des dix réalisateurs les plus lourds.

Et on peut donc recommencer en utilisant les

Itérer ces calculs dix fois en affichant à chaque fois la liste des dix acteurs les plus centraux.

Est-ce que cette liste se stabilise-t-elle ? Que peut-on conclure de cette suite d'opérations ? Avez-vous des modifications à proposer qui pourraient améliorer le résultat ? (Essayer avec au moins les données de l'année 1990 et de la décennie 1990–1999.)

Solution de l'exercice de la section 1 :

```
from igraph import *
g = Graph()
g.add_vertices(4)
g.add_edges([(0,1),(0,2),(0,3),(2,2)])
g.vs["name"]=["A","B","C","D"]
g.vs["color"]=["blue","blue","red","red"]
visual_style={}
visual_style["vertex_label"]=g.vs["name"]
visual_style["edge_width"]=[1,1,10,1]
layout=g.layout("kk")
visual_style["layout"]=layout
plot(g, **visual_style)
```