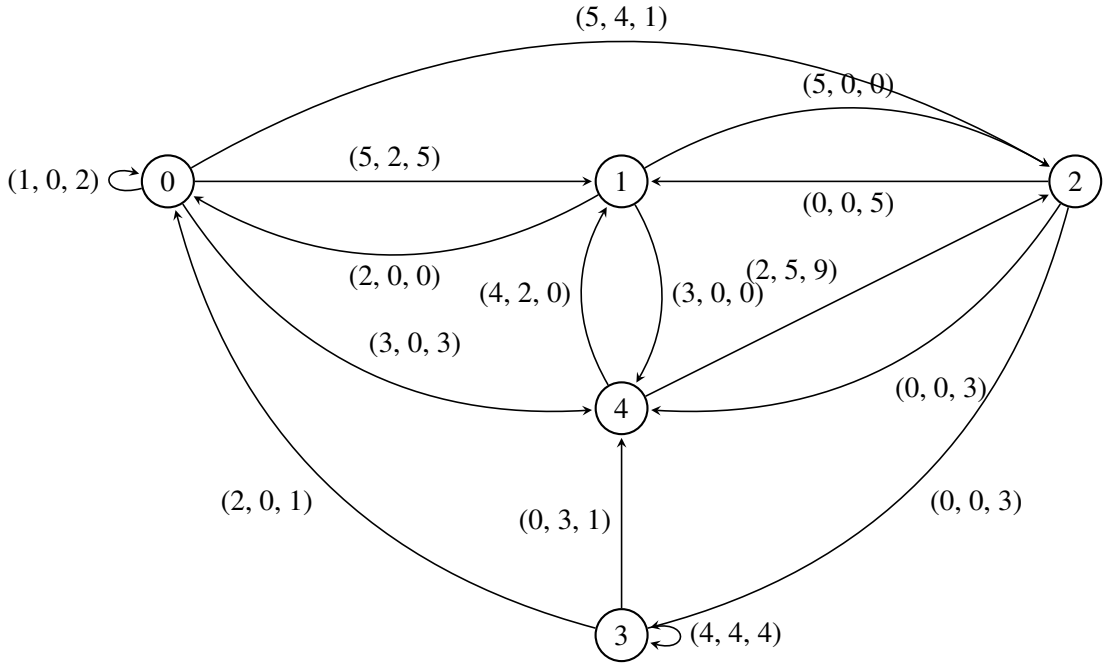
École national Supérieure d'Informatique
2ème année cycle supérieure (2CSSIQ2, 2CSSIL2)
Machine Learning (2022-2023)

Workshop : Numpy

ARIES Abdelkrime

Nous voulons étudier la popularité des utilisateurs sur Twitter. Pour ce faire, nous allons étudier les interactions (Reply, Retweet, Like) entre 5 utilisateurs. Le nombre des tweets de chaque utilisateur est représenté par le vecteur : [2, 5, 10, 4, 3]. Les réponses ne sont pas considérées comme des tweets. Le graphe suivant représente les interactions de la forme (Reply, Retweet, Like) entre les utilisateurs. Lorsqu'il n'y a pas un arc d'un nœud vers un autre, ceci veut dire que les trois interactions sont (0, 0, 0).



- Représenter les interactions sous forme de trois matrices 5X5 pour chaque type d'interaction.

Statistiques d'engagement

Ici, nous voulons extraire quelques statistiques sur l'engagement des utilisateurs avec les autres. Dans la théorie des graphes, nous nous intéressons par les arcs sortants.

1. Trouver l'utilisateur qui répond (Reply) plus que les autres.
2. Trouver l'utilisateur qui partage (Retweet) plus que les autres.
3. Trouver les utilisateurs qui ne répondent pas (Reply) et qui ne partagent pas (Retweet).
4. Afficher le nombre des Likes que chaque utilisateur a donné à ses propres tweets.
5. Afficher le nombre des Likes que chaque utilisateur a donné aux tweets autre que ces propres tweets.
6. Trouver les utilisateurs qui aiment leurs propres tweets plus qu'ils aiment les tweets des autres.
7. Calculer leur nombre.
8. Afficher le nombre des interactions que chaque utilisateur a fait.

Statistiques sur les interactions

1. Afficher le max des Likes que chaque utilisateur a reçu (des utilisateurs autres que lui-même). *Pour annuler la diagonale, nous pouvons créer une matrice diagonale 5X5 et inverser les valeurs. Ensuite, appliquer une multiplication élément par élément.*
2. Trouver les utilisateurs qui ont reçu des Likes pour tous leurs tweets au moins par une personne autre qu'elle. Par exemple, l'utilisateur 1 a publié 5 postes et il existe au moins un autre utilisateur qui a Liker ces 5 tweets (dans ce cas, 0 et 2).
3. Calculer le nombre des arcs entrants (ayant au moins une interaction).
4. Calculer le nombre des arcs sortants (qui ont fait au moins une interaction).
5. Trouver les nœuds où le nombre des arcs entrants est égale au nombre de ceux sortants.
6. Afficher les Likes qu'ils ont fait entre eux.

Popularité

- Nous ne nous intéressons pas par les interactions réflexives (Ex. quelqu'un qui aime ses propres tweets) qui sont représentées par la diagonale pour les Likes et les Reply. Par contre, nous nous intéressons par les Retweets réflexives. *Pour annuler la diagonale, nous pouvons créer une matrice diagonale 5X5 et inverser les valeurs. Ensuite, appliquer une multiplication élément par élément.*
- On veut pondérer les matrices à tel sorte que :
  - Score Reply : nous supposons que les Reply sont des objections en général ; donc, ils ont un effet négatif. Pour chaque interaction, si le nombre des Reply reçus d'une personne est plus que celui de ces Likes, nous multiplions le nombre des Reply par (-0.5). S'il est égale, nous multiplions ces propres Likes par (-0.25). Sinon, il n'a aucun effet (0). **HINT : utiliser numpy.where**
  - Score Retweet : nous supposons que les Retweet sont neutres en général ; donc, nous n'appliquons aucune transformation sur ces interactions. Mais, si quelqu'un fait des Retweets à ces propres tweets, il va être sanctionné par (-0.25) pour chaque retweet.
  - Score Likes : nous supposons que les Likes ont un effet positif sur la popularité. Pour chaque interaction, si le nombre des Likes reçus d'une personne est plus grand que la moitié des tweets publiées, nous le multiplions par (1.5). Sinon, le score est le nombre des Likes.
- Le score de popularité de chaque utilisateur pour chaque interaction est la moyenne des scores reçus.
- Le score final est la moyenne des scores des différentes interactions.