∽ Projet 2 : Social Network Analysis with Python ∾

Fourth Year of Data Science - ESPRIT School of Engineering

Notion de complexité : La complexité, ou le coût, d'un algorithme ou d'une fonction Python est le nombre d'opérations élémentaires nécessaires à son exécution dans le pire cas. Lorsque cette complexité dépend d'un ensemble de paramètres $(n, p, ...)$, on pourra donner cette estimation sous forme asymptotique. On rappelle qu'une application $c(n, p, ...)$ est dans la classe $O(f)$ s'il existe une constante $\alpha > 0$ telle que $|c(n, p, ...)| < \alpha * f(n, p, ...)$, pour toutes les valeurs de $n, p, ...$ assez grandes. Nous préciserons plus loin le coût de chacune des opérations utilisées dans ce projet. Notons qu'on donne toujours une complexité asymptotique, c'est-à-dire quand n et p tendent vers $+\infty$.

1 Introduction

Qu'est-ce que l'analyse de réseau social ?

— 53% d'utilisateurs de Twitter recommandent des produits de consommation dans leurs tweets.

— 90% de consommateurs ont confiance dans les recommandations en ligne (peer recommendations).

— Uniquement 14% ont confiance dans les publicités.

— Les new-yorkais ont reçus des tweets sur un possible séisme 30 secondes avant sa frappe.

— 93% des commerçants utilisent les réseaux sociaux dans leur business.

Dans ce projet, on se propose de regrouper des personnes par affinité dans un réseau social. Pour cela, on cherche à répartir les personnes en deux groupes de sorte à minimiser le nombre de liens d'amitié entre les deux groupes. Une telle partition s'appelle une coupe minimale du réseau.

Première partie

Partie théorique et implémentation d'un algorithme opérant sur les graphes (partie obligatoire)

2 Réseaux sociaux

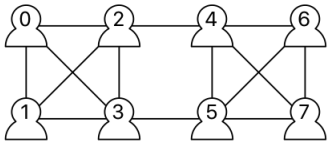
Structure de données : On suppose que les individus sont numérotés de 0 à n-1 où n est le nombre total d'individus. On représentera chaque lien d'amitié entre deux individus i et j par une liste contenant leurs deux numéros dans un ordre quelconque, c'est-à-dire par la liste [i,j] ou la la liste [j,i] indifféremment.

Un réseau social R entre n individus sera représenté par une liste **reseau** à deux éléments où :

— **reseau[0] = n** contient le nombre d'individus appartenant au réseau

— **reseau[1] = la liste non-ordonnée** (éventuellement vide) des liens d'amitié déclarés entre les individus.

La figure 1 donne l'exemple d'un réseau social et d'une représentation possible sous la forme de liste. Chaque lien d'amitié entre deux personnes est représenté par un trait entre elles.



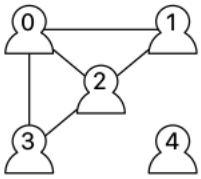

```
reseau = [ 8,
           [ [0,1], [1,3], [3,2], [2,0], [0,3], [2,1], [4,5],
             [5,7], [7,6], [6,4], [7,4], [6,5], [2,4], [5,3] ]
         ]
```

FIGURE 1 – Un réseau à 8 individus ayant 14 liens d'amitié déclarés et une de ses représentations possibles en mémoire sous forme d'une liste [Nombre d'individus, liste des liens d'amitié].

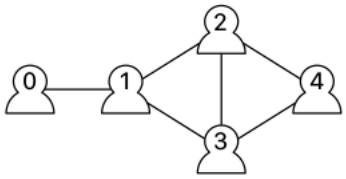
❑ **Q1** – Donner une représentation sous forme de listes pour chacun des deux réseaux sociaux ci-dessous :

❑ **Q2** – Écrire une fonction creerReseauVide(n) qui crée, initialise et renvoie la représentation sous forme de liste du réseau à n individus n'ayant aucun lien d'amitié déclaré entre eux.

❑ **Q3** – Écrire une fonction estUnLienEntre(paire,i,j) où paire est une liste à deux éléments et i et j sont deux entiers, et qui renvoie True si les deux éléments contenus dans paire sont i et j dans un ordre quelconque ; et renvoie False sinon.



Réseau A



Réseau B

❑ **Q4** – Écrire une fonction sontAmis(reseau,i,j) qui renvoie True s'il existe un lien d'amitié entre les individus i et j dans le réseau reseau ; et renvoie False sinon.

Quelle est la complexité en temps de votre fonction dans le pire cas en fonction du nombre m de liens d'amitié déclarés dans le réseau ?

❑ **Q5** – Écrire une procédure declareAmis(reseau,i,j) qui modifie le réseau reseau pour y ajouter le lien d'amitié entre les individus i et j si ce lien n'y figure pas déjà.

Quelle est la complexité en temps de votre procédure dans le pire cas en fonction du nombre m de liens d'amitié déclarés dans le réseau ?

❑ **Q6** – Écrire une fonction listeDesAmisDe(reseau,i) qui renvoie la liste des amis de i dans le réseau reseau.

Quelle est la complexité en temps de votre fonction dans le pire cas en fonction du nombre m de liens d'amitié déclarés dans le réseau ?

3 Partitions

Une partition en k groupes d'un ensemble A à n éléments consiste en k sous-ensembles disjoints non-vides $A_1, ..., A_k$ de A dont l'union est A, c'est-à-dire, tels que $A_1 \cup ... \cup A_k = A$ et pour tout $i \neq j, A_i \cap A_j = \emptyset$. Par exemple $A_1 = 1,3$, $A_2 = 0,4,5$, $A_3 = 2$ est une partition en trois groupes de $A = [|6|]$. Dans cette partie, on implémente une structure de données très efficace pour coder des partitions de $[|n|]$.

Le principe de cette structure de données est que les éléments de chaque groupe sont struc- turés par une relation filiale : chaque élément a un (unique) parent choisi dans le groupe et l'unique élément du groupe qui est son propre parent est appelé le représentant du groupe. On s'assure par construction que chaque élément i du groupe a bien pour ancêtre le représentant du groupe, c.-à-d. que le représentant du groupe est bien le parent du parent du parent etc. (autant de fois que nécessaire) du parent de l'élément i. La relation filiale est symbolisée par une flèche allant de l'enfant au parent dans la figure 2 qui présente un exemple de cette structure de données. Dans l'exemple de cette figure, 14 a pour parent 11 qui a pour parent 1 qui a pour parent 9 qui est son propre parent. Ainsi, 9 est le représentant du groupe auquel appartiennent 14, 11, 1 et 9. Notons que ce groupe contient également 8, 13 et 15.

À noter que la représentation n'est pas unique (si l'on choisit un autre représentant pour un groupe et une autre relation filiale, on aura une autre représentation du groupe).

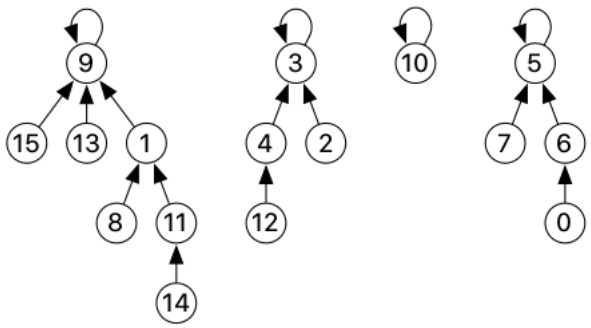


FIGURE 2 – Une représentation filiale de la partition suivante de [|16|] en quatre groupes : 1, 8, 9, 11, 13, 14, 15, 2, 3, 4, 12, 10 et 0, 5, 6, 7 dont les représentants respectifs sont 9, 3, 10 et 5.

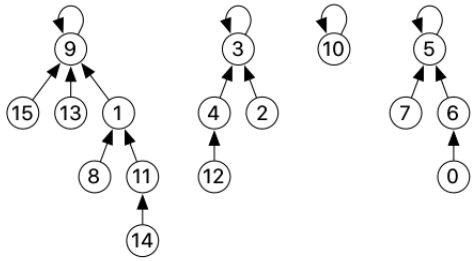
Pour coder cette structure, on utilise un tableau parent à n éléments où la case parent[i] contient le numéro du parent de i. Par exemple, les valeurs du tableau parent encodant la représentation filiale donnée dans la figure 2 sont :

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
parent[i]	6	9	3	3	3	5	5	5	1	9	10	1	4	9	11	9

❑ **Q7** – Donner les valeurs du tableau parent encodant les représentations filiales des deux partitions de [|10|] ci-dessous, et préciser les représentants de chaque groupe :

Initialement, chaque élément de [|n|] est son propre représentant et la partition initiale consiste en n groupes contenant chacun un individu. Ainsi, initialement, parent[i] = i pour tout $i \in [|n|]$.

❑ **Q8** – Écrire une fonction creerPartitionEnSingletons(n) qui crée et renvoie un tableau parent à n éléments dont les valeurs sont initialisées de sorte à encoder la partition de [|n|] en n groupes d'un seul élément.



Nous sommes intéressés par deux opérations sur les partitions :

— Déterminer si deux éléments appartiennent au même groupe dans la partition.

— Fusionner deux groupes pour n'en faire plus qu'un. Par exemple, la fusion des groupes $A_1 = 1, 3$ et $A_3 = 2$ dans la partition de [|6|] donnée en exemple au tout début de cette partie donnera la partition en deux groupes $A_2 = 0, 4, 5$ et A_4 où $A_4 = A_1 \cup A_3 = 1, 2, 3$.

❑ **Q9** – Écrire une fonction representant(parent,i) qui utilise le tableau parent pour trouver et renvoyer l'indice du représentant du groupe auquel appartient i dans la partition encodée par le tableau parent.

Quelle est la complexité dans le pire cas de votre fonction en fonction du nombre total n d'éléments ? Donnez un exemple de tableau parent à n éléments qui atteigne cette complexité dans le pire cas.

Pour réaliser la fusion de deux groupes désignés par l'un de leurs éléments i et j respective- ment, on applique l'algorithme suivant :

— 1 - Calculer les représentants p et q des deux groupes contenant i et j respectivement.

— 2 - Faire parent[p] = q.

La figure 3 présente la structure filiale obtenue après la fusion des groupes contenant respectivement 6 et 14 de la figure 2.

❑ **Q10** – Écrire une procédure fusion(parent,i,j) qui modifie le tableau parent pour fusionner les deux groupes contenant i et j respectivement.

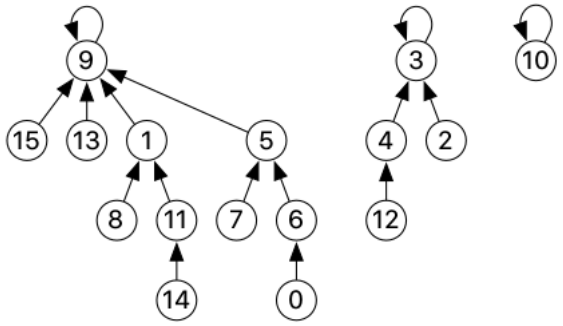


FIGURE 3 – Représentation filiale obtenue après la fusion des groupes contenant respectivement 6 et 14 de la figure 2.

Pour l'instant, la structure de données n'est pas très efficace comme le montre la question suivante.

❑ **Q11** – Proposer une suite de (n - 1) fusions dont l'exécution à partir de la partition en n singletons de [|n|], nécessite de l'ordre de n^2 opérations élémentaires.

Pour remédier à cette mauvaise performance, une astuce consiste à compresser la rela- tion filiale après chaque appel à la fonction representant(parent,i). L'opération de com- pression consiste à faire la chose suivante : si p est le résultat de l'appel à la fonction representant(parent,i), modifier le tableau parent de façon à ce que chaque ancêtre (c.- à-d. parent de parent . . . de parent) de i, i y compris, ait pour parent direct p. Noter bien que même si un appel à representant(parent,i) renvoie le représentant de i elle modifie également le tableau parent. Si l'on reprend l'exemple de la figure 2, le résultat de l'appel representant(parent,14) est 9, que l'on a calculé en remontant les ancêtres successifs de 14 : 11, 1 puis 9. L'opération de compression consiste alors à donner la valeur 9 aux cases d'indices 14, 11, et 1 du tableau parent. La structure filiale obtenue après l'opération de compression menée depuis 14 est illustrée dans la figure 4.

❑ **Q12** – Modifier votre fonction representant(parent,i) pour qu'elle modifie le tableau parent pour faire pointer directement tous les ancêtres de i vers le représentant de i une fois qu'il a été trouvé.

En quoi cette optimisation de la structure filiale peut-elle être considérée comme gratuite du point de vue de la complexité ?

Afin d'afficher de manière lisible la partition codée par un tableau parent, on souhaite calculer à partir du tableau parent la liste des listes des éléments des différents groupes. Une sortie possible pour le tableau parent correspondant à la figure 2 serait :

❑ **Q13** – Écrire une fonction listeDesGroupes(parent) qui renvoie la liste des différents groupes codés par le tableau parent sous la forme d'une liste des listes des éléments des différents groupes.

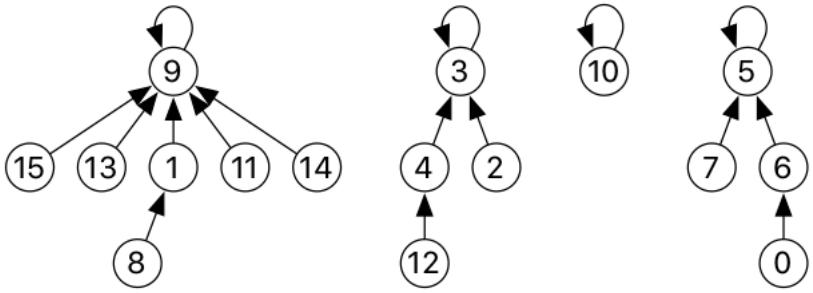


FIGURE 4 – Résultat de la compression depuis 14 dans la représentation filiale de la figure 2..

```
[ [ 15, 8, 1, 9, 11, 13, 14 ],
  [ 4, 3, 2, 12 ],
  [ 7, 5, 6, 0 ],
  [ 10 ] ]
```

4 Algorithme randomisé pour la coupe minimum

Revenons à présent à notre objectif principal : Trouver une partition des individus d'un réseau social en deux groupes qui minimise le nombre de liens d'amitiés entre les deux groupes.

Pour résoudre ce problème nous allons utiliser l'algorithme randomisé suivant :

— Entrée : un réseau social à n individus
— 1. Créer une partition P en n singletons de [|n|]
— 2. Initialement aucun lien d'amitié n'est marqué
— 3. Tant que la partition P contient au moins trois groupes et qu'il reste des liens d'amitié non-marqués dans le réseau faire :
— (a) Choisir un lien uniformément au hasard parmi les liens non-marqués du réseau, notons-le [i,j].
— (b) Si i et j n'appartiennent pas au même groupe dans la partition P, fusionner les deux groupes correspondants
— (c) Marquer le lien [i,j].
— 4. Si P contient $k \geq 3$ groupes, faire k 1 fusions pour obtenir deux groupes.
— 5. Renvoyer la partition P.

La figure 5 présente une exécution possible de cet algorithme randomisé sur le réseau de la figure 1.

❑ **Q14** – Écrire une fonction coupeMinimumRandomisee(reseau) qui renvoie le tableau parent correspondant à la partition calculée par l'algorithme ci-dessus.

Indication. Au lieu de marquer explicitement les liens déjà vus, on pourra avantageusement les positionner à la fin de la liste non-ordonnée des liens du réseau et ainsi pouvoir tirer simplement les liens au hasard parmi ceux placés au début de la liste.

Quelle est la complexité de votre procédure en fonction de n, m et α(n), où m est le nombre de liens d'amitié déclarés dans le réseau et où α(n) désigne la complexité d'un appel à la fonction representant ?

❑ **Q15** – Écrire une fonction tailleCoupe(reseau, parent) qui calcule le nombre de liens entre les différents groupes de la partition représentée par parent dans le réseau reseau.

On peut démontrer que cet algorithme renvoie une coupe de taille minimum avec une pro- babilité supérieure à 1/n, ce qui fait que la meilleure parmi n exécutions indépendantes de cet algorithme est effectivement minimum avec probabilité supérieure à 1/e 0.36787....

La structure de données filiale avec compression pour les partitions est particulièrement efficace aussi bien en pratique qu'en théorie. En effet, la complexité de k opérations est de O(kα(k)) opérations élémentaires où α(k) est l'inverse de la fonction d'Ackermann, une fonction qui croit extrêmement lentement vers l'infini (par exemple $\alpha(10^{80}) = 5$).

Deuxième partie

Analyse des vraies données d'un réseau social (Obligatoire mais le choix du sujet est libre)

5 Introduction

Dans cette partie, l'étudiant choisit un réseau social entre Twitter, Facebook, YouTube, Instagram et il essaye de télécharger les données à travers les APIs (si c'est possible) sinon avec les modules "beautiful Soup : https ://www.crummy.com/software/BeautifulSoup/bs4/doc/" et "scrapy : https ://scrapy.org/".

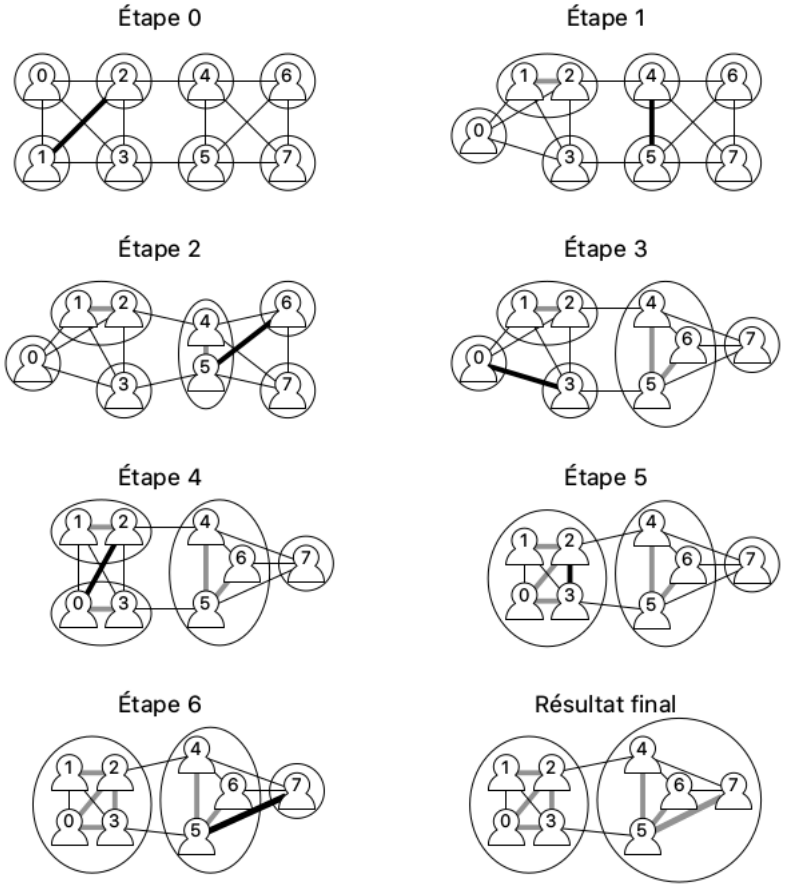


FIGURE 5 – Une exécution de l'algorithme randomisé sur le réseau de la figure 1 où les liens sélectionnés aléatoirement sont dans l'ordre : [2,1], [4,5], [6,5], [0,3], [2,0], [3,2] et [5,7]. Les liens représentés en noir épais sont les liens sélectionnés au hasard à l'étape courante ; les liens épais et grisés sont les liens marqués par l'algorithme ; les ronds représentent la partition à l'étape courante.

6 projet 1 : Analyse des sentiments sur Twitter avec la théorie des graphes

6.1 sujet :

Twitter est un réseau social permettant aux utilisateurs de partager de micro-messages limités à 280 caractères, appelés tweets. Aujourd'hui, Twitter gère 500 millions de tweets par jour envoyés par 320 millions d'utilisateurs actifs par mois, ce qui montre l'importance de ce réseau social dans le monde. En effet, Twitter a été utilisé par les jeunes en Egypte en janvier 2011 pour partager les nouvelles et pour motiver les révolutionnaires (comme cette vidéo le montre sur YouTube https ://www.youtube.com/watch ?feature=player_embedded&v=2guKJfvq4uI). D'autre part, des articles de recherche scientifique en économie ont montré que les tweets du président actuel des Etats-Unis ont un impact sur le cours des dévises (US Dollar vs Mexican Peso).

Il est alors important d'analyser les données circulant de Twitter avec les outils fournis par la théorie des graphes pour quantifier les sentiments (sentiment analysis of Twitter)

6.2 Travail à faire :

L'étudiant utilise les outils suivants :

— python et les modules importants : numpy, pandas, matplotlib, scipy, scikit-learn, seaborn, beautiful Soup, scrapy
— Le module igraph de python http ://igraph.org/python/
— Twitter API.
— Gephi : The Open Graph Viz Platform https ://gephi.org/users/download/ (pour visualiser les graphes interactivement) (optionnel)

Pour effectuer les tâches suivantes :

— Chargement de données directement à partir de Twitter ou bien à partir d'un fichier.
— Analyse des données avec python.
— Transformation des données brutes en graphes.
— Analyse de l'impact des résultats sur un marché financier par exemple. (l'étudiant est libre dans son choix de sujet)
— Rédaction d'un rapport avec jupyter notebook.

7 projet 2 : Analyse du réseau social Facebook avec la théorie des graphes

7.1 sujet :

Facebook est le premier réseau social au monde. Il compte 2 milliards de profiles avec 1 milliards d'utilisateurs actifs par jour qui partagent leurs activités journalières (textes, photos, vidéos, ...) avec leurs amis et qui échangent des messages instantanés. Facebook est le site le plus visité au monde et il a dépassé google. Aujourd'hui, le business model de Facebook est basé sur la vente des données personnelles. Ces données sont achetées par des entreprises afin de :

- Planifier la politique de marketing.
- Prédire le cours des marchés financiers.
- Comprendre l'évolution des sociétés.
- Lutter contre le terrorisme et la cyber-criminalité.

L'étudiant choisit une problématique à traiter pour comprendre les liens entre les utilisateurs et les modéliser par des graphes.

7.2 Travail à faire :

L'étudiant utilise les outils suivants :

— python et les modules importants : numpy, pandas, matplotlib, scipy, scikit-learn, seaborn, beautiful Soup, scrapy

— Le module igraph de python http ://igraph.org/python/

— Facebook API.

— Gephi : The Open Graph Viz Platform https ://gephi.org/users/download/ (pour visualiser les graphes interactivement) (optionnel)

Pour effectuer les tâches suivantes :

— Chargement de données directement à partir de Facebook ou bien à partir d'un fichier.

— Analyse des données avec python.

— Transformation des données brutes en graphes.

— Analyse de l'impact des résultats sur un marché financier par exemple. (l'étudiant est libre dans son choix de sujet)

— Rédaction d'un rapport avec jupyter notebook.

8 projet 3 : Réaliser les mêmes tâches sur YouTube par exemple