Compte Rendu TME 4

Wenzhuo ZHAO, Zhaojie LU, Chengyu YANG, Zhen HOU

Mars 2021

1 k-core decomposition

Nous avons implémenté l'algorithme de core décomposition (**Core value of graph**), le calcul de la moyenne de densité de degré (**the average density**), le calcul de densité d'arête et le calcul de la taille du core le plus dense en ordre préfixe (**the size of a densest core ordering prefixe**).

1.1 Core Décomposition

Introduit dans le cours, cet algorithme permet de calculer la core valeur d'un graphe. Le principe de cet algorithme est de trouver le noeud avec le minimum degré, puis compare son degré avec la valeur c (initialement à 0) puis fait la réassignation que $c = max(c, degree)$. Supprimer ce noeud et tous les arêtes reliées en tenant compte ses voisins. Bien évidemment, le degré de ses voisins va diminuer par 1. Refaire cette boucle, jusqu'à ce qu'il n'existe aucun noeud dans le graphe et nous obtenons la valeur s à la fin.
La difficulté d'implémenter cet algorithme est de trouver le noeud avec le minimum degré. Vu que la suppression d'un noeud va provoquer le changement de degrés de noeuds, donc il faut rechercher le noeud avec le minimum degré chaque foi. Pour résoudre ce problème, soit on utilise la fonction $min()$ qui a une complexité en temps $O(n)$, soit on trie avec un algorithme en complexité $O(nlogn)$ le tableau contenant le degrés de noeuds après chaque modification puis prends le premier élément dans ce tableau. Donc une implémentation naïve de cet algorithme a une complexité minimum de $O(n^2)$ qui est inacceptable.
Pour diminuer la complexité, il faut mettre à jour le tableau contenant les degrés de noeuds dynamiquement avec un coût constant.
Soit un noeud i avec le degré n, si un de ses voisins est supprimé alors son degré va diminuer par 1 donc son nouveau degré sera $n - 1$. Le tableau de degrés n'est plus trié, donc il faudra échanger ce noeud avec la tête de cette section de degrés.

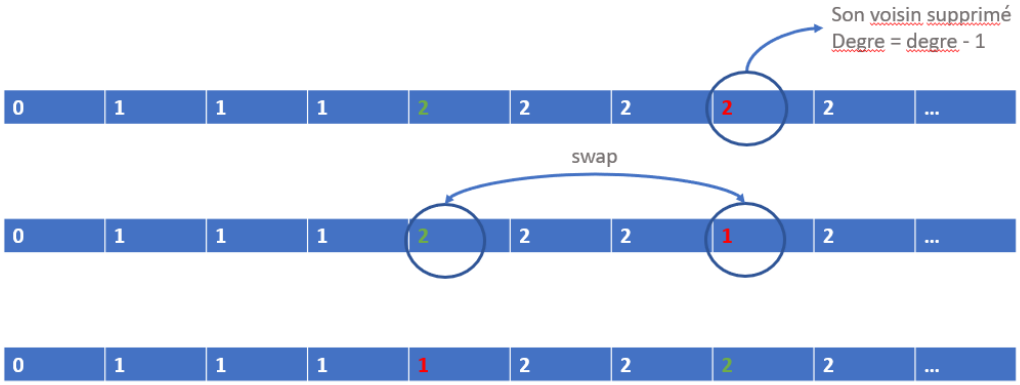

Figure 1: Trier un tableau

Dans ce figure, le noeud en rouge dont le degré a diminué par 1 (de 2 à 1). On échange ce noeud, avec la tête de cette section ayant le même degré (le noeud en vert). Après cet échange, ce tableau est bien trié. Soit le degré en moyenne est m, la complexité de notre implémentation de cet algorithme est $O(nm)$ qui est de loin inférieur à $O(n^2)$.

1.2 Performance

Après l'exécution, nous avons obtenu les résultats ci-dessous :

- Pour le graphe Amazon product co-purchasing network [1]
  - Calculation time : 3.695700168609619 secondes
  - Core value: 6
  - average degree density : 2.7649277465709856
  - the edge density : 1.6513834036534368e-05
  - maximum value of densest prefix: 4.0
  - the size of a densest core ordering prefix: 51
- Pour le graphe LiveJournal social network and ground-truth communities [3]
  - Calculation time: 826.6175100803375 secondes
  - Core value: 360
  - average degree density : 8.674717018320834
  - the edge density : 4.339570605276457e-06
  - maximum value of densest prefix: 190.98445595854923
  - the size of a densest core ordering prefix: 386
- Pour le graphe Orkut social network and ground-truth communities [4]

- Calculation time: 11351.06044793129 secondes
- CoreCore value: 253
- average degree density : 38.140710594605395
- the edge density : 2.4827635751783857e-05
- maximum value of densest prefix: 227.87243515292295
- the size of a densest core ordering prefix: 25830

2 Graph mining with k-core

Dans cette partie, nous donnerons une visualisation de la corrélation entre le degré et la valeur core de chaque noeud sur le graphe concernant le lien entre des auteurs d'articles sur Google Scholar [2].

2.1 Performance

En appliquant l'algorithme au-dessus , nous avons obtenu les résultats ci-dessous.

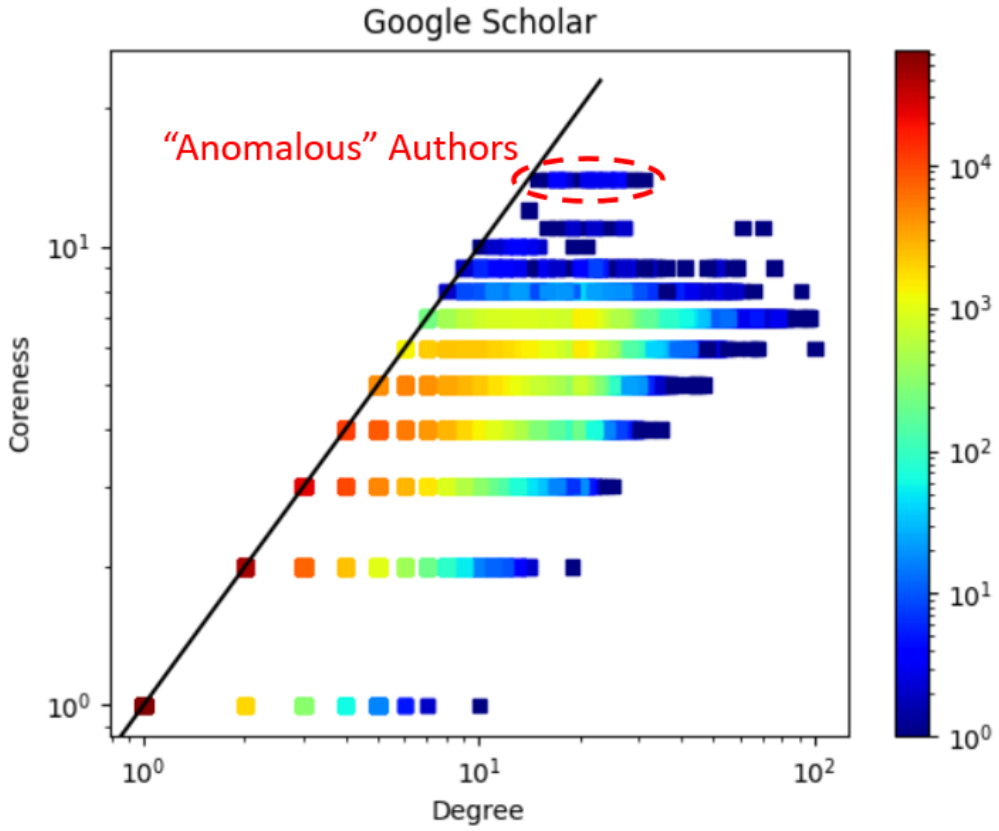

Figure 2: résultat

Nous pouvons constater qu'il existe une ligne courte dans le graphe.La ligne signifie qu'il y a un groupe d'auteur avec la valeur de core le plus grande parmi tous les auteurs mais leurs degrés sont relativement petites. Cela signifie que ces auteurs peuvent-être avoir les références mutuelles. Donc nous pouvons dire qu'il sont les auteurs anormaux.

Voici l'affichage des auteurs anormaux : Sa-kwang Song Sung-Pil Choi Chang-Hoo Jeong Yun-soo Choi Hong-Woo Chun Jinhyung Kim Hanmin Jung Do-Heon Jeong Myunggwon Hwang Won-Kyung Sung Hwamook Yoon Minho Lee Won-Goo Lee Jung Ho Um Dongmin Seo Mi-Nyeong Hwang Sung J. Jung Min-hee Cho Sungho Shin Seungwoo Lee Heekwan Koo Jinhee Lee Taehong Kim Mikyoung Lee Ha-neul Yeom Seungkyun Hong Yun-ji Jang

Appendix A Source Code

Le source code de ce TME est disponible sur notre répertoire de GitHub [5].

References

[1] *Amazon product co-purchasing network and ground-truth communities.* `http://snap.stanford.edu/data/com-Amazon.html`.

[2] *Google Scholar.* `https://drive.google.com/file/d/0B6cGK503Ibt0dXA3Z2lJcHlLX28/view`.

[3] *LiveJournal social network and ground-truth communities.* `http://snap.stanford.edu/data/com-LiveJournal.html`.

[4] *Orkut social network and ground-truth communities.* `http://snap.stanford.edu/data/com-Orkut.html`.

[5] W.Zhao. *K Core decomposition.* `https://github.com/valeeraZ/Sorbonne_CPA_Graph/tree/master/TME4`.