Détection de communautés dans des réseaux sociaux

Introduction :

Le but du projet est de vous initier à la lecture et la compréhension d'articles scientifiques. Dans ce but, vous allez implémenter certains algorithmes de graphes issus de la littérature scientifique récente. Le contexte est celui des réseaux sociaux et de leur étude. Un réseau social peut facilement être vu comme un graphe où les nœuds représentent les utilisateurs et les arêtes les liens qui existent entre eux dans le réseau. Par exemple pour Facebook, le lien peut représenter un lien d'amitié. La problématique principale du projet est d'énumérer certaines structures dans des graphes modélisant théoriquement certains réseaux sociaux. En particulier on va chercher dans ces graphes des communautés, c'est-à-dire des ensembles de nœuds très densément connectés entre eux. Connaître la structure des communautés dans un graphe trouve de nombreuses applications. Cela permet, par exemple, de simplifier le ciblage publicitaire ou bien de détecter des clusters potentiels où peuvent se propager rapidement des virus. En théorie des graphes, il existe plusieurs modélisations possibles pour la notion de communautés. On peut par exemple modélise utiliser la notion de cliques maximales ou de bicliques maximales, dont les définitions sont données dans le projet.

Instructions :

Le but du projet est d'implémenter des algorithmes d'énumération de bicliques maximales dans des graphes générés aléatoirement. Le projet s'articule en trois parties.

La première partie consiste à générer des graphes aléatoires Etant donné un nombre fixe de sommets, chaque arête apparaît avec probabilité *p* choisie aléatoirement telle que $0<p<1$. Vous devrez stocker vos graphes à l'aide de listes d'adjacences. L'objectif est de créer des graphes les plus grands possibles en termes de sommets. Dans une second temps, on vous demande d'utiliser la base de données des grands graphes de Stanford [1] et de stocker en mémoire les graphes des réseaux sociaux suivants : ego-Facebook, feather-lastfm-social et email-Eu-core. Vous donnerez pour les graphes aléatoires que vous avez engendrés et les graphes de Stanford, lorsque c'est possible :

1. Le degré maximum du graphe.
2. Un graphique donnant pour chaque degré du graphe le nombre de sommets ayant ce degré.
3. Le nombre de chemins induits dans le graphe de longueur 2. Il s'agit des chemins à 3 sommets tels qu'il n'y ait pas d'arêtes entre le premier et le dernier sommet du chemin.

Attention, les graphes générés dans cette partie seront utilisés dans les partie suivantes. Il faut bien choisir vos structures de données et les fonctions pour la manipulation de ces graphes.

Dans la deuxième partie on vous demande d'écrire un algorithme de complexité exponentielle pour l'énumération des cliques d'un graphe. Soit un graphe G=(V,E) avec V l'ensemble des sommets et E

l'ensemble des arêtes de G. Une clique de G est un ensemble K de sommets du graphe tous connecté deux à deux. La clique K est maximale si elle n'est incluse dans aucune autre clique de G, c'est-à-dire qu'il n'existe pas de sommet dans V\K connecté à tous les sommets de K. L'algorithme de Bron Kerbosch est un des algorithmes les plus connus énumérant les cliques maximales d'un graphe. Il en existe de nombreuses variantes. On vous demande d'implémenter la version avec pivot de l'algorithme. **Attention** pour cette partie, on vous demande de trouver l'algorithme vous-même sur le web et de l'implémenter. Vous pouvez utiliser du code existant sur internet. Vous devrez alors très précisément donner vos sources. Tout code qui sera pris sans citation sera considéré comme du plagiat. On vous demande dans le code, de mettre un exemple d'exécution sur un graphe aléatoire de la première partie. Vous donnerez un exemple sur un petit graphe puis vous ferez une exécution sur un graphe aussi grand que possible en termes de sommets.

Finalement, dans **la troisième partie,** on vous demande d'implémenter divers algorithmes liés à l'énumération des cliques et de bicliques maximales. Une clique est un graphe complet. Elle est maximale s'il n'est pas possible d'ajouter d'autres sommets du graphe et d'obtenir une clique de taille plus grande. Une biclique est un graphe biparti complet. C'est à dire que toutes les arêtes sont présentes entre les deux parties du graphe. Elle est maximale s'il n'est pas possible de rajouter des sommets à la biclique et d'obtenir une biclique de taille (en termes de sommets) plus grande. On vous demande d'implémenter les algorithmes suivants :

1. Les lignes 1 et 2 de l'algorithme 1 dans le papier [2]. Pour ce faire, vous aurez besoin de lire et de comprendre la Définition 1 qui se trouve dans le même papier. On demande un exemple d'exécution.
2. Bonus : vous implémenterez la ligne 5 dans la boucle FOR qui commence à la ligne 4 de l'algorithme 1. Vous aurez besoin de l'algorithme d'énumération de la partie 2. On vous rappelle qu'un ensemble indépendant maximal dans un graphe forme une clique dans le graphe complémentaire.

Rendu : Un rapport est attendu, dans lequel on veut une présentation théorique des algorithmes et une comparaison sur différentes instances des graphes générés en première partie de projet en termes de temps et d'espace d'exécution (en fonction du nombre de sommets et d'arêtes). Dans le rapport, vous mettrez les résultats attendus dans les diverses parties du projet. On attend aussi à ce que vous décriviez précisément le contenu de ce que vous avez fait. Le langage de programmation sera du C++ ou du Python. Vous devrez mentionner les librairies utilisées. On demande que le code soit propre, commenté et fournit avec un ReadMe décrivant comment le compiler et l'exécuter, ainsi qu'un exemple d'exécution des algorithmes sur des graphes générés en première partie de projet.

Références :

[1] https://snap.stanford.edu/data/

[2] Hermelin, Manoussakis. "Efficient enumeration of maximal induced bicliques," Discrete Applied Mathematics, 2021.