

Détection de communautés dans des réseaux sociaux

Introduction :

Le but du projet est d'écrire un algorithme issu de la littérature scientifique et de le rendre le plus compétitif possible, en termes de temps d'exécution. Dans ce but, nous demandons de lire un article scientifique récent et d'implémenter un algorithme de graphes, issu de cet article.

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 denses dans des graphes modélisant théoriquement 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.

En théorie des graphes, on modélise souvent les communautés par des cliques maximales. Leur définition est donnée plus tard dans le projet.

Instructions :

Le but du projet est d'implémenter un algorithme d'énumération de cliques maximales dans des graphes générés aléatoirement, qui vous sont donnés. Il faut que votre algorithme soit le plus rapide possible en terme de temps d'exécution. Le projet s'articule en deux parties.

La première partie consiste à étudier rapidement les graphes qui vous sont donnés. On vous demande, pour chacun des graphes du projet de remplir un tableau contenant les informations suivantes. Le degré maximal du graphe, son nombre d'arêtes, son degré moyen. Il faut aussi faire un graphique qui donne le nombre de nœuds pour chaque degré possible du graphe.

La seconde partie du projet consiste à écrire un algorithme d'énumération des cliques maximales. On rappelle quelques définitions. 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 $V \setminus K$ connecté à tous les sommets de K .

On vous demande d'implémenter l'algorithme 1 du papier donné avec le projet. Vous aurez donc aussi besoin dans un premier temps de calculer l'ordre de dégénérescence du graphe (l'algorithme est décrit dans le même papier). Vous êtes libre pour le choix de vos structures de données, hormis celles concernant le graphe qui seront stockés sous forme de listes d'adjacence. Vous pourrez aussi changer librement la façon dont sont stockés certains résultats dans l'algorithme (par exemple un arbre des suffixes est utilisé dans le papier ; vous pouvez le remplacer par une table de hachage etc.). Pour la sous routine d'énumération des cliques maximales, vous pouvez utiliser l'algorithme que vous voulez, au choix. Vous êtes autorisé à utiliser des algorithmes déjà implémentés ou à l'implémenter grâce à l'intelligence artificielle. Une approche classique est d'utiliser l'algorithme de Bron Kerbosch.

On veut un tableau qui donne pour chaque graphe donné dans le projet, les informations suivantes : leur dégénérescence, le nombre de cliques maximales trouvées, leur taille maximale, et le temps d'exécution.

Rendu :

Un rapport est attendu, dans lequel on veut une présentation théorique de l'algorithme que vous avez implémenté avec des détails techniques concernant votre code. Il faut aussi donner les tableaux demandés dans le projet. Le langage de programmation sera du python. On demande que le code soit propre, commenté et fournit avec un ReadMe décrivant comment le compiler et l'exécuter. Le projet peut se faire par groupe de deux personnes. Le rendu sera un fichier ZIP: NOM1-NOM2.zip