

Projet d'algorithmes sur les graphes Les réseaux sociaux

1 Contexte

Les réseaux sociaux en ligne ont connu un essor spectaculaire ces dernières années, et les entreprises commencent à s'approprier ce concepts pour des applications plus ciblées comme le travail collaboratif, la recherche de compétences ou encore le recrutement.

Un réseau social peut être défini comme un ensemble d'entités sociales connectées entre elles en fonction d'intérêts communs. Ces relations peuvent être uniques (i.e. ami de) ou multi dimensionnelles (client, collègue, relation spatiale, etc.). Elles peuvent également être homogènes (individus du même type) ou hétérogènes, symétriques ou non. Les réseaux sociaux d'entreprises sont caractérisés par des relations multiples et un contenu riche. Les graphes constituent une représentation adaptée pour modéliser l'information ou les interactions sociales. L'analyse des réseaux sociaux consiste à comprendre et interpréter le comportement d'un réseau. Cette analyse peut également fournir des informations sur la manière dont les communautés se forment et interagissent. Les réseaux sociaux ont été étudiés d'un point de vue mathématique et statistique, mais aussi en informatique pour les aspects recherche, navigation et visualisation sociale.

Il y a donc un besoin important de collecter les données pour les représenter par un graphe de données sémantiquement riche, les analyser pour trouver des relations et groupes cachés, faire de la prédiction de liens, caractériser le réseau et ses principaux acteurs, permettre une recherche sémantique et par facettes, établir un réseau de confiance.

Nous proposons pour ce projet de réaliser une étude (simplifiée) d'un réseau social (simplifié) de type Facebook. Pour mettre en place cette étude, nous utiliserons quelques notions de théorie des graphes qui vont nous permettre de représenter ces réseaux et de les analyser efficacement.

2 Modélisation du graphe

On considère un réseau social. Les utilisateurs de ce réseau créent des comptes de deux types : Utilisateur, Page. Les comptes sont représentés par des sommets dans ce graphe. Chaque compte de type Utilisateur peut suivre d'autres comptes de type Utilisateur pour voir leurs notifications. Chaque compte de type Utilisateur peut aimer des comptes de type Page et ainsi devenir capable d'en consulter le contenu. En revanche, les comptes de type Page n'ont aucune action possible. La relation x "suit" y , avec x et y des comptes Utilisateur est représentée par un arc du sommet x vers le sommet y dans le graphe. La relation x aime y , avec x un compte Utilisateur et y un compte Page est représentée par un arc du sommet x vers le sommet y dans le graphe.

3 Gestion du graphe

Les comptes Utilisateur sont identifiés par un nom et ont des informations supplémentaires comme le prénom et l'âge. Les comptes Page sont également identifiés par un nom et ont une liste d'administrateurs de type Utilisateur. Afin de gérer le graphe modlisant le réseau, il vous est demandé de :

- créer une classe abstraite Sommet, définir un constructeur ainsi que des accesseurs publics. Chaque Sommet contiendra la liste de ses voisins sortants;
- écrire les classes Utilisateur et Page qui héritent de la classe Sommet;
- coder la classe Graphe qui devra représenter un graphe sous forme de liste d'adjacence et donc contenir un ensemble de Sommet.

4 Opérations de base sur le graphe

Pour commencer à étudier les propriétés du graphe d'un réseau social, il faut tout d'abord être capable d'en explorer le contenu facilement et de disposer de quelques mesures simples mais très utilisées. Pour cela, vous devez implémenter les fonctionnalités suivantes :

- connaître le nombre de sommets, d'arcs;
- obtenir l'ensemble des sommets;

- obtenir l'ensemble des sommets trié par nom;
- obtenir l'ensemble des sommets trié par degré sortant;
- obtenir l'ensemble des arcs;
- ajouter/Supprimer un sommet (et les arcs qui y sont liés);
- ajouter/Supprimer un arc;
- obtenir les informations sur un sommet via son nom;
- connaître le nombre de comptes de type Page, Utilisateur;
- Connaître l'âge moyen des Utilisateur;
- Connaître tous les comptes Utilisateur qui sont des administrateurs de Page;
- écrire le graphe dans un fichier sous forme de liste d'adjacence;
- lire un graphe dans un fichier sous forme de liste d'adjacence.

Une interface devra être proposée dans votre application pour qu'un utilisateur puisse appeler toutes ces fonctionnalités, en visualiser les résultats et ainsi commencer à fouiller les données de son graphe.

5 Qui est le plus influent ?

De nombreuses méthodes sont utilisées pour mesurer l'influence ou la pertinence d'un utilisateur sur un réseau social, notamment sur Twitter. Une méthode possible est d'appliquer un algorithme de PageRank, également utilisé par Google pour réaliser le classement des résultats sur son moteur de recherche. L'intuition de cet algorithme est celle-ci : Plus le nombre de comptes qui me suivent est élevé, plus mon compte est influent et pertinent; Plus les comptes qui me suivent sont influents et pertinents, plus mon compte est influent et pertinent;

Afin de pouvoir déterminer les comptes Utilisateur et Page qui sont les plus influents au sein du graphe, cet algorithme devra être implémenté dans votre application. Par ailleurs, l'interface utilisateur sera enrichie de la possibilité de lancer cet algorithme et d'en visualiser les résultats.

6 Des réseaux Small world

En 1967, Stanley Milgram, sociologue, déclarait que que chaque être humain sur terre était relié à n'importe quel autre être humain via quelques (six) degrés de connaissance seulement. Cette affirmation est également connue comme le paradoxe de Milgram [2]. Pour vérifier cette hypothèse, vous implémenterez l'algorithme de la section 7.3. Il prend en paramètre un sommet du graphe (qu'on appelle source) et permet de calculer la plus petite distance entre ce sommet source et chacun des autres sommets du graphe. En exécutant cet algorithme avec pour source chacun des sommets du graphe, on peut connaître la plus petite distance entre chaque paire de sommets du graphe.

7 Algorithmes

7.1 Notations

On appelle degré entrant (resp. sortant) d'un sommet $v \in V$ le nombre d'arcs dont l'extrémité (resp. l'origine) est v . On le note $d^-(v)$ (resp. $d^+(v)$).

On dit que $v \in V$ est un voisin sortant (resp. entrant) de $u \in V$ s'il existe un arc (u, v) (resp. (v, u)) $\in A$. Pour $v \in V$, nous définissons $N^+(v)$ (resp. $N^-(v)$) comme l'ensemble des voisins sortants (resp. entrants) de v .

7.2 Calcul du Pagerank

```
// Initialisation du PageRank à 1 pour tous les sommets
Pour tout  $v \in V$  faire
     $PR(v) \leftarrow 1$ ;
// Calcul du PageRank
 $i \leftarrow 0$ ;
Tant que  $i \leq 100$  faire
    // Pour tous les sommets du graphe
    Pour tout  $v \in V$  faire
        // On calcule le PageRank du sommet
         $PR(v) \leftarrow \frac{0.15}{|V|} + 0.85 \sum_{u \in N^-(v)} \frac{PR(u)}{d^+(u)}$ ;
     $i \leftarrow i + 1$ ;
```

7.3 Plus courtes distances entre la source s et les sommets de V

```
//Initialisation des distances entre chaque sommet et la source  $s$  à 10000000
 $\forall u \in V, dist(v) \leftarrow 10000000$ ;
// Initialisation de la distance du sommet source à 0  $dist(s) \leftarrow 0$ ;
//  $P$  prend l'ensemble des sommets du graphe
 $P \leftarrow V$ ;
// Calcul de la plus petite distance entre chaque sommet et la source
Tant que  $P$  non Vide faire
    // On prend le sommet  $u$  de  $P$  le plus proche de la source
     $u \leftarrow$  sommet de  $P$  avec  $dist$  minimale;
    Supprimer  $u$  de  $P$ ;
    // Pour tous les voisins de ce sommet  $u$ 
    Pour tout  $v \in N^+(u)$  faire
         $alt \leftarrow dist(u, s) + 1$ ;
        Si  $alt \leq dist(v, s)$  alors
             $dist(v, s) \leftarrow alt$ ;
```

8 Modalités

Ce projet doit être réalisé par groupes de deux étudiants exactement. Le cas échéant, et avec accord du responsable de TD, le projet pourra être réalisé seul ou bien par un unique groupe de trois personnes.

Le langage de programmation sera de préférence Python, mais vous pouvez aussi choisir Ocaml ou Java.

Le code source, le code exécutable et le rapport devront être envoyés sous forme d'archive par courrier électronique, ceci au plus tard **dimanche 8 janvier 2017, à minuit**, à Jean-Philippe Dubernard à l'adresse suivante :

`jean-philippe.dubernard@univ-rouen.fr`

Le rapport devra comprendre une notice d'utilisation, la description des structures de données, les algorithmes principaux ainsi que leur complexité, des choix liés à la programmation, des difficultés rencontrées et des jeux d'essai.

Les projets donneront lieu à une soutenance avec démonstration qui aura lieu en janvier.

References

- [1] Sciences
et Avenir, "Facebook envoie les citoyens aux urnes", Septembre 2012,
<http://sciencesetavenir.nouvelobs.com/fondamental/20120917.OBS2643/>
- [2] Wikipedia, "Etude du petit monde",
http://fr.wikipedia.org/wiki/Etude_du_petit_monde