TELECOM Bretagne

Majeure INFO - INF 435 - TP3

Dans la peau d'un Community Manager

Cécile Bothorel

Automne 2014

1 Objectif

Dans le cadre du mini-projet du module de graphes, vous avez implémenté les parcours de graphe (BFS et DFS) et la découverte d'arbre couvrant (Kruskal et Prim), qui sont deux étapes de la proposition de Chaudhury [1]. Pour ce dernier TP, nous vous fournissons le programme python implémentant la proposition de Chaudhury dans `k_cores.py`. Ce programme permet d'identifier k influenceurs et sert à faire du ciblage marketing sur les réseaux sociaux.

L'objectif de cette séance de TP est d'utiliser cet algorithme et de le comparer à d'autres solutions que vous, vous allez implémenter.

Au terme de ce TP, vous serez capables :

- de décrire le principe et le résultat d'un algorithme de calcul de k-influenceurs,
- d'appliquer cet algorithme à une problématique de e-marketing,
- de mobiliser les métriques vues en cours pour définir d'autres solutions de choix de k influenceurs,
- de manipuler deux algorithmes de diffusion dans les réseaux sociaux,
- de comparer et analyser les résultats produits par vos solutions et la solution du mini-projet.

2 Votre mission

Vous allez jouer le rôle de Community Manager pour une marque de Yaourts. Votre rôle est d'accroître la notoriété de votre entreprise, de fidéliser les clients ou d'en acquérir de nouveaux [2]. Parmi vos tâches quotidiennes, vous devez :

- animer des communautés en ligne : susciter des débats, répondre aux questions des internautes, rassurer des clients mécontents, etc.
- mettre en oeuvre des opérations événementielles sur le web,
- analyser le buzz marketing et suivre l'image de la marque,
- et proposer des axes d'amélioration, en vous coordonnant avec les équipes techniques.

Votre marque va bientôt lancer une gamme innovante de yaourts, les *Tatie Novio*, déclinés sur différents parfums. Le service Marketing a imaginé de faire tester les nouveaux produits en phase beta par quelques clients, et en fonction de leurs retours, la marque lancera tout ou juste une partie des *Novio*.

Habituellement, vous vous fiez à votre instinct pour repérer les "leaders d'opinion". Mais vous êtes curieux et vous vous intéressez aux travaux de recherche en eMarketing. Vous souhaitez aujourd'hui tester la solution proposée par Chaudhury [1].

Vous avez donc demandé aux équipes techniques de télécharger une partie du réseau social Gazoulli.com. Vous savez fédérer vos collègues, et vous avez donc obtenu ces données toutes bien préparées, sous la forme d'un graphe d'internautes : le graphe `socialnet.graph`.

Votre problème est : **Comment repérer les internautes influents qui vont permettre de faire connaître les Tatie Novio à plus d'internautes possibles sur le Web.** Une contrainte cependant, vous n'avez que 10 yaourts à faire goûter (ils ne sont pas encore produits en masse) et vous devez donc sélectionner au plus $k = 10$ personnes pour le test, en espérant que ces $k <= 10$ personnes diffusent à leurs proches de manière convaincante !

Question 1 : Comment repérer les internautes influents qui vont permettre de faire connaître les Tatie Novio à plus d'internautes possibles ? Imaginez d'autres méthodes que celle préconisée par Chaudhury.

Question 2 : Implémentez deux mesures d'influence pour obtenir deux listes de k influenceurs.

Pour cela, nous vous conseillons de respecter les mêmes paramètres d'entrée/sortie que la fonction `k_cores.k_core_algorithm()`. Prenez donc le temps de bien comprendre le programme fourni, et la manière de l'utiliser dans `tp3.py`.

Question 3 : Comparez vos 2 listes de k influenceurs avec la solution de Chaudhury. Vous pouvez utiliser le graphe des Dauphins et le graphe de Promotion qui vous connaissez déjà. Vous pouvez aussi utiliser le graphe SocialNet (attention, il est beaucoup plus gros). Les k influenceurs détectés sont ils les mêmes ? Y-a-t'il des points communs ? Même question avec la mesure PageRank qui est fournie également.

Question 4 : Appliquez des algorithmes de diffusion pour comparer l'efficacité des listes d'influenceurs.

Pour comprendre les deux modèles de diffusion que nous vous avons implémentés, vous pouvez regarder les slides fournis jusqu'à la page 23 [3]. Pour les utiliser, vous donnez en entrée une liste initiale de k noeuds, et en sortie, vous obtenez la liste des noeuds activés. Ce sont les noeuds qui ont reçu l'information diffusée au départ à partir des influenceurs.

3 Instructions

Pour commencer la séance de TP, prenez le temps de bien regarder le code fourni. Il commence à être complexe.

Comme toujours, le point d'entrée du programme est `tp3.py`. Il contient le chargement du graphe en mémoire à partir d'un fichier, et lance les diverses méthodes de détection d'influenceurs : `k_cores.k_core_algorithm()`, `measures.page_rank()`. C'est là que vous ajouterez l'appel à vos 2 méthodes. Puis le programme applique des algorithmes de diffusion pour évaluer l'efficacité de votre action marketing en fonction des k influenceurs que vous sollicitez : `diffusion.independent_cascade()` et `diffusion.linear_threshold()`. Grâce à graphviz.show_active_nodes() vous visualisez les internautes touchés par l'action de marketing.

En complément de `tp3.py`, vous trouverez des fichiers additionnels :

- les données de graphes : `promotion.graph`, `dolphin.graph` et `socialnet.graph`.
- `k_cores.py` qui contient tout le code relatif à la méthode de Chaudhury. Vous appellerez cette méthode grâce à la fonction `k_cores.k_core_algorithm()`.

- `graphviz.py` offre deux méthodes de visualisation. La fonction `graphviz.show_core_network()` qui permet de visualiser l'arbre couvrant maximal et les k cores sélectionnés par la solution `k_cores` ; dans ce fichier, vous pourrez aussi utiliser la fonction `graphviz.show_active_nodes()` qui génère une image du graphe entier avec les k influenceurs sélectionnés au départ mais aussi les noeuds "infectés" par la diffusion d'un message.
- `measures.py` qui contient une autre mesure pour sélectionner les k noeuds initiaux : la fonction `measures.page_rank()`.
- Et enfin, dans `diffusion.py` se trouvent deux algorithmes de diffusion `diffusion.linear_threshold()` et `diffusion.independent_cascade()`.

Votre mission consiste d'abord à compléter le fichier `measures.py` avec d'autres mesures vues en cours, avec les mêmes paramètres d'entrée sortie que le Pagerank, puis de tester différents scénarios de diffusion dans `tp3.py`.

Références

[1] Arpan Chaudhury, Partha Basuchowdhuri and Subhashis Majumder. Spread of Information in a Social Network Using Influential Nodes. PAKDD 2012, Part II, LNAI 7302, pp. 121-132, 2012

[2] Fiche métier sur le site du Pôle Emploi, consultée en mars 2014, `http://www.pole-emploi.fr/actualites/le-metier-de-community-manager-@/suarticle.jspz?id=105714`

[3] David Kempe, Jon Kleinberg, Éva Tardos. Maximizing the Spread of Influence through a Social Network. Presented to University of Washington Colloquium, January 2004, consultée en mars 2014, `http://www.cs.washington.edu/affiliates/meetings/talks04/kempe.pdf`