TP 4 : Igraph/R - Détection de communuatés

Rushed Kanawati

31 octobre 2016

Ce TP est à réaliser en binôme. Un seul rapport est à rendre à la fin de la séance

Résumé

Dans Ce TP nous étudions différentes approches de détection de communautés dans des réseaux complexes.

Détection et évaluation de communautés

1. Télécharger les graphes suivants `dolphins.gml, polblogs.gml, football.gml, karate.gml` sur le site `http://lipn.fr/~kanawati/ars`. Dans ces graphes les communautés sont désignées par l'attribut `value`.
2. La bibliothèque `igraph` offre différentes fonctions de détection de communautés. Pour chaque graphe de l'exercice 1, appliquer les méthodes suivantes : `cluster_edge_betweenness, cluster_walktrap, cluster_louvain`. Comparer les qualités des communautés retrouvées par rapport à la verité terrain (en termes de NMI et de ARI). conclusion ?

   utiliser la fonction `compare(com1,com2,method="nmi")`
3. Utiliser la fonction `plot(com,g)` pour visualiser les communautés retrouvées.
4. Comparer les similarité des structures communautaires retournées par les différents algorithmes entre elles (toujours en fonction de NMI et ARI). Conclusion ?
5. Donner la modularité de chaque structure communautaire retrouvée et comparer la avec la modularité de la partition de référence. Conclusion ?

   utiliser la fonction `modularity(com)`

Cœurs de communautés

1. Appliquer l'algorithme de détection de communautés `cluster_label_prop` $n = 10$ fois sur chacun des graphes de l'exercice 1.1. Pour chaque graphe, et chaque exécution donner la qualité de la structure communautaire retrouvée par rapport à la partition de référence. Conclusion ?

2. Développer une fonction de fusion des différentes partitions d'un même graphe (comme vu en cours), et appliquer cette fonction sur les résultats de l'exécution de `cluster_label_prop` n fois sur chaque graphe. Etudier l'effet du paramètre α (seuil de fréquence minimale) sur la qualité de la fusion.
3. Varier n de 10 à 100 et étudier son effet sur la qualité de la structure communautaire retrouvée (pour chaque graphe)
4. Fixer le paramètre α à 1 et utiliser la structure communautaire retrouvée après fusion pour réduire la taille du graphe (utiliser la fonction `contract`)
5. Télécharger le graphe `dblp72-75.gml`. La structure communautaire de ce graphe n'est pas donnée. Extraire la plus grande composante connexe de ce graphe. Sur le sous-graphe obtenue, appliquer la procédure suivante : réduire le graphe en appliquant la fonction développée dans la question précédente, pour une valeur donnée de n (nombre d'exécution de label propagation), appliquer `cluster_louvain` sur le graphe et utiliser la structure communautaire retrouvée pour calculer une partition du graphe d'origine. Faites varier n et étudier son effet sur la qualité de la structure communautaire (mesurée en termes de modularité), conclusion ?