Recherche de communautés dans les grands réseaux sociaux

Emmanuel Viennet

Université de Paris-Nord, L2TI - Institut Galilée
99, avenue Jean-Baptiste Clément
93430 Villetaneuse, France
emmanuel.viennet@univ-paris13.fr

Résumé. Cet article décrit quelques méthodes récentes pour la recherche de communautés dans les grands réseaux sociaux (dizaines voire centaines de millions de nœuds). Après avoir rappelé quelques notions de base sur ce sujet, nous décrivons quelques approches récentes pour l'extraction de micro-communautés et de communautés globales, et montrons quelques résultats prouvant que ces méthodes sont parfaitement utilisables pour la fouille d'ensembles de données parmi les plus grands rencontrés aujourd'hui dans les applications industrielles.

1 Introduction

Un "réseau social" est un graphe dont les nœuds sont des individus ou organisations, connectés par des liens représentant une relation "sociale" : appartenance à la même famille, échange de messages, goûts communs... (voir l'article de P. Kuntz et F. Picarougne dans ce même numéro). L'étude des réseaux sociaux est très active depuis quelques années (voir par exemple Barabasi (2002)), et les techniques automatiques permettent d'étudier les propriétés statistiques de réseaux de très grandes tailles, comme celui formé par les sites web de l'Internet ou l'ensemble des appels téléphoniques passés sur un opérateur de Télécommunications. L'étude des réseaux sociaux intéresse depuis quelques décennies les chercheurs en sciences sociales (voir par exemple Peter J. Carrington (2005)) et a réuni depuis la fin des années 90 une importante communauté de chercheurs d'horizons divers, attirés tant par la découvertes d'intéressantes propriétés théoriques de ces structures que par la richesse des applications potentielles.

La connaissance des liens entre individus (étude de la formation de communautés, de la propagation des rumeurs ou modes au sein de celles-ci, etc.) est d'une grande importance pour la fouille de données : les applications sont nombreuses en marketing, en bioinformatique, en analyse de données textuelles.

Les premiers travaux sur les réseaux sociaux ont cherché à caractériser ceux-ci (types de structures) et à décrire des classes de nœuds : individus "influents" (*hubs*) ou "suiveurs", etc., puis à développer des outils collaboratifs exploitant ces structures. L'analyse des réseaux sociaux rejoint aussi les préoccupations de la communauté des chercheurs en *Link Analysis* (voir par exemple Workshop on Link Analysis (2006)), qui s'intéressent à des problèmes dans lesquels les données sont hétérogènes et arrivent de sources variées, incluant des représentations

de personnes, organisations, actions, évènements... chacune de ces entités étant définie par ses propres attributs, et pouvant être en relation avec d'autres entités.

Dans cet article, nous présentons un bref état de l'art sur les méthodes de recherche de communautés dans les réseaux sociaux. Nous nous focalisons sur les approches "structurelles" exploitant l'architecture du graphe et applicables à de *très grands* jeux de données (graphes pouvant dépasser le million d'arêtes). Ceci exclut à priori de nombreuses approches basées sur une modélisation fine des propriétés du graphe, que leur complexité algorithmique limite au traitement de petits graphes (quelques centaines ou milliers de nœuds). D'autre part, ce champ d'étude étant actuellement en fort développement, nous ne prétendons couvrir toutes les méthodes proposées.

1.1 Notions de base sur les réseaux sociaux

Les réseaux sociaux sont représentés par des graphes ; tous les outils de la théorie des graphes s'appliquent donc à l'étude des réseaux sociaux.

Un graphe est défini comme un couple $G = (E, N)$ où N est un ensemble de nœuds et E un ensemble d'arêtes, dirigées ou pas : on parle donc de graphe dirigé ou de graphe non dirigé (selon que l'on fait ou non une distinction entre l'arête du nœud i vers le nœud j et celle de j vers i).

Un graphe peut être *valué* dans le cas où chaque arête porte une valeur (un nombre réel). Les nœuds peuvent être décorés de plusieurs attributs.

On peut représenter un graphe par sa *matrice d'adjacence* A, de dimension $n \times n$ où n est le nombre de nœuds du graphe, avec

$$A_{ij} = \begin{cases} 1 & \text{s'il y a une arête du nœud } i \text{ vers le nœud } j \\ 0 & \text{sinon} \end{cases}$$

Les réseaux sociaux ne peuvent cependant se réduire à des graphes : la description d'un réseau social est généralement beaucoup plus complexe qu'un simple ensemble de nœuds et d'arêtes. Le réseau social est caractérisé par des données hétérogènes associées tant aux nœuds qu'aux arêtes (imaginons par exemple une application où des clients, sur lesquels on disposerait d'informations variées, échangeraient entre eux des messages électroniques : les attributs des clients stockés dans la base clients peuvent alors décorer les nœuds, alors que diverses caractéristiques des messages – nombre sur une semaine, taille du message... – décoreraient les arêtes).

D'autre part, les graphes issus des réseaux sociaux ont souvent une caractéristique particulière, que l'on retrouve dans de nombreux réseaux naturels (réseaux de distribution, réseaux de communication, biologie, etc, voir Barabasi (2002)) : ils sont *sans échelle*, c'est à dire que la distribution statistique des degrés dans le graphe (nombre d'arêtes liant un nœud à ses voisins) suit une loi de puissance : la probabilité $P(k)$ qu'un nœud du graphe possède k voisins est de la forme

$$P(k) = k^{-\gamma}$$

Cette propriété (figure 1), qui distingue les réseaux sans échelles des graphes aléatoires, est liée à une autre propriété intéressante, l'effet "*petit monde*". Dans un réseau "petit monde", le nombre de liens à traverser pour connecter deux nœuds quelconques est en moyenne petit, et ce même pour de très grands réseaux.

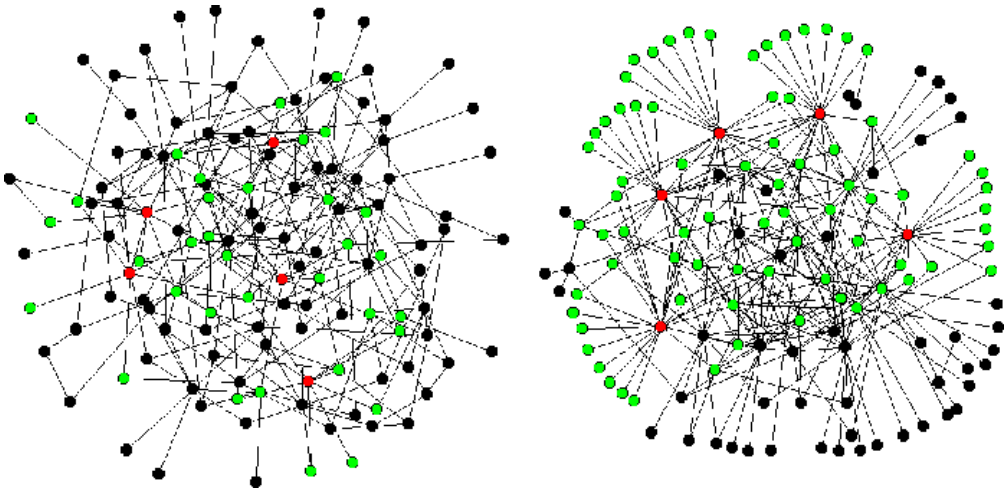

FIG. 1 – *Illustration de la différence entre un graphe aléatoire (à gauche) et un graphe sans échelle (à droite). Le graphe sans échelle est très in-homogène : la majorité des nœuds ont seulement un ou deux liens, mais quelques uns (nommés* hubs*) ont un très grand nombre de liens et assurent la connexion de l'ensemble (source : Albert et al. (2000)).*

Les figures 2 et 3 montrent quelques exemples classiques de "réseaux sociaux".

Les applications Internet récentes engendrent de nombreux jeux de données de type "réseaux sociaux", et le modèle économique de plusieurs d'entre elles repose explicitement sur l'exploitation de l'information portée par le réseau social des utilisateurs (par exemple pour leur envoyer de la publicité ciblée). Les plus connues sont Facebook, Flickr, MySpace, LinkedIn, Friendster, etc.

L'échelle des réseaux manipulés est très variable, les applications récentes générant de gigantesques structures. Quelques ordres de grandeurs donnés par Leskovec (2007) :

- 100 à 1000 nœuds : réseaux d'échanges de messages e-mails dans de petites organisations ;
- 50 000 nœuds : e-mails d'une université durant deux ans ;
- 4,5 millions de nœuds : "amitiés" déclarées sur un site de blog collaboratif ;
- 240 millions de nœuds : réseau formé par toutes les communications IM de Microsoft Instant Messenger pendant un seul mois.

1.2 Communautés

La plupart des réseaux sociaux sont structurés en "communautés". La définition précise d'une communauté est floue et dépend des applications visées, aussi peut-on se contenter, dans un premier temps, de définir une communauté comme *un ensemble de nœuds qui ont plus de liens entre eux qu'avec le reste du réseau* (figure 4). Le sous-graphe correspondant est souvent très densément connecté (le cas extrême serait un groupe d'amis dont chacun connaîtrait tous les autres). La communauté est alors un ensemble d'individus qui interagissent beaucoup entre

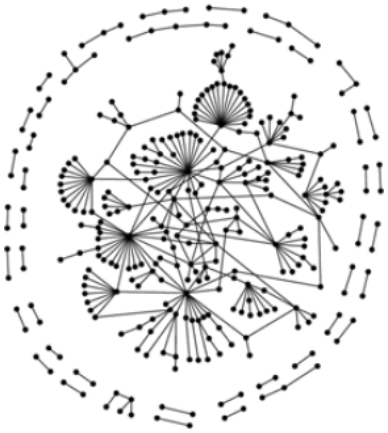

FIG. 2 – *Réseau d'interactions génétiques entre des levures (source : Albert et al. (2000)).*

eux, mais n'ont que peu d'interactions avec le reste du monde. La communauté est ainsi un moyen de "segmenter" les clients.

La connaissance de ces communautés peut constituer un atout précieux pour aborder des problèmes d'apprentissage dans des situations où les "entités" font partie de réseaux sociaux. On pourrait par exemple envisager, dans la perspective de son classement, d'enrichir la description de chaque nœud par des indicateurs calculés sur sa communauté. on peut aussi s'intéresser aux communautés de clients partageant les mêmes goûts pour les produits et tenter utiliser cette information pour affiner des recommandations (publicité ciblée) au sein d'une communauté donnée. Par ailleurs, quand on travaille sur des réseaux de très grande taille, le confinement du problème à une communauté donnée permet la réduction de la complexité de traitement (moins de clients). Pour autant qu'on puisse extraire les communautés simplement !

Dans la suite de ce document, nous allons nous intéresser aux méthodes permettant de détecter automatiquement les communautés dans un réseau donné.

2 Types d'approche pour l'extraction de communautés

La recherche de communauté dans des graphes est évidemment liée aux travaux, lancés dès les années 50, sur les algorithmes de partitionnement de graphes. Si certaines méthodes issues de la théorie des graphes sont exploitables, la problématique générale est cependant assez différente : tout d'abord, on s'intéresse spécialement à des graphes parcimonieux (*sparses*) et/ou sans échelle. D'autre part, la majorité des travaux antérieurs sur le partitionnement de graphe s'adressent à des problèmes pour lesquels le nombre de partitions (ou leur taille) est connu. Un exemple classique est le problème du placement de tâches sur des N processeurs en minimisant les communications inter-processeurs, qui se ramène à un partitionnement de graphe en N partitions. A l'opposé, les problèmes de détection de communautés s'apparentent plus à ceux traités par les systèmes d'apprentissage non supervisés (clustering), dans lesquels on ne connaît pas à priori le nombre de classes à découvrir.

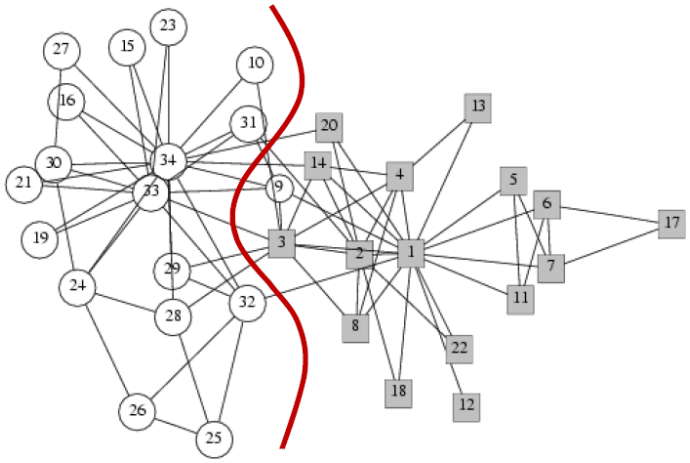

FIG. 3 – *Le réseau du club de Karaté étudié par Zachary (1972), avec deux classes. Ce petit réseau a été souvent utilisé comme test pour des méthodes d'analyse.*

Cependant, certains outils classiques de la théorie des graphes, comme les méthodes de partitionnement spectrales, restent très utiles dans tous les cas (voir Newman (2006)). Notons que si les recherches sur le partitionnement de graphe sont menées par des informaticiens et mathématiciens, la plupart des travaux sur la détection de communautés viennent des sociologues et, depuis un petite dizaine d'années, des physiciens et de la communauté de la fouille de données (*data mining*) [1]

Au niveau général, on peut distinguer différentes familles d'approches (figure 5). Les approches par partitionnement, souvent issues des travaux en théorie des graphes, donnent un découpage unique du graphe. Les approches hiérarchiques donnent un ensemble de découpages emboîtés ; on peut alors utiliser un critère externe pour sélectionner le niveau de découpage le plus pertinent.

Notons qu'une notion plus simple de "micro-communauté" peut être utilisée si l'on veut simplifier les calculs : il s'agit cette fois d'une méthode "bottom-up" où on s'intéresse à l'environnement qu'un nœud est capable d' "influencer". La micro-communauté peut être définie comme l'ensemble des nœuds "voisins" du nœud : c'est le cercle du nœud (c'est-à-dire l'ensemble des nœuds auxquels le nœud est connecté) ou de façon plus large l'ensemble des nœuds qui peuvent être atteints "rapidement" à partir du nœud. La définition précise de "rapidement" indique le nombre de liens qu'on est prêt à franchir à partir du nœud initial, généralement 2 ou 3, c'est-à-dire le "temps" maximal qu'on est prêt à attendre pour voir se propager une information à partir du nœud initial. Comme la détermination du cercle est très rapide en temps de calcul, la micro-communauté se calcule également très rapidement. Les techniques dont nous parlons maintenant ne concernent donc que la détection de communautés au sens de la section 1.2.

[1] Voir notamment les sessions et conférences invitées à la conférence KDD'08 : `http://www.kdd.org/kdd2008`.

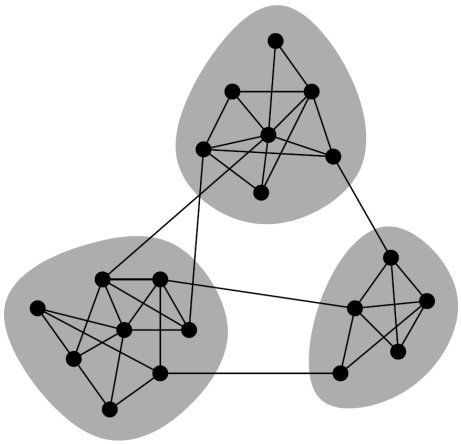

FIG. 4 – *Graphe structuré en trois communautés.*

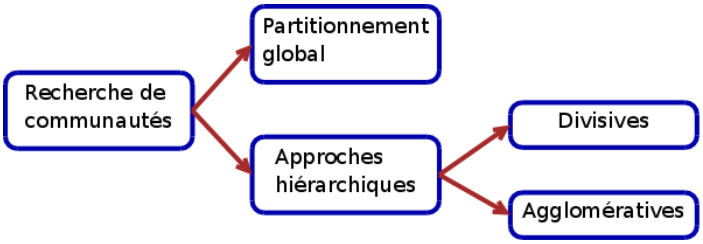

FIG. 5 – *Approches pour la recherche de communautés.*

2.1 Approches par partitionnement

Les principales méthodes classiques de partitionnement global sont les suivantes.

Bissection spectrale ces méthodes sont basées sur la résolution approchée du problème d'optimisation global (*graph cut*) par une recherche des valeurs propres de la matrice Laplacienne du graphe Pothen et al. (1990).

Ces méthodes fonctionnent bien lorsque le graphe se sépare naturellement en deux, mais résistent mal au bruit. Pour diviser le graphe en $N > 2$ communautés, il faut itérer la bissection, avec des résultats qui ne sont pas toujours très satisfaisants. D'autre part, ces méthodes ne donnent aucune indication sur le nombre optimal de communautés.

La matrice Laplacienne d'un graphe G de matrice d'adjacence A est la matrice $L = D - A$ où D est la matrice de degré du graphe :

$$D_{ij} = \begin{cases} \text{degré de } n_i & \text{si } i = j \\ 0 & \text{sinon} \end{cases}$$

Algorithme de Kernighan-Lin Cet algorithme (Kernighan et Lin (1970)) utilise une approche gloutonne pour optimiser un critère de qualité de partition défini comme la différence entre le nombre des d'arêtes internes des deux groupes et le nombre d'arêtes reliant les deux groupes. Comme la précédente, cette méthode ne traite que la bissection du graphe. De plus, on doit spécifier au préalable les tailles des deux communautés. La complexité de cet algorithme est $O(n^2)$.

2.2 Méthodes hiérarchiques

L'idée générale des méthodes hiérarchiques est de construire une hiérarchie de communautés imbriquées (figure 6). Ces méthodes peuvent être *agglomératives* (partant d'un nœud et y associant progressivement des voisins) ou *divisives* (découpage itératif du graphe). Les méthodes agglomératives calculent les similarités entre les paires de nœuds et créent des liens, en commençant par les nœuds les plus semblables, tandis que les méthodes divisives retirent petit à petit des arêtes au graphe de départ.

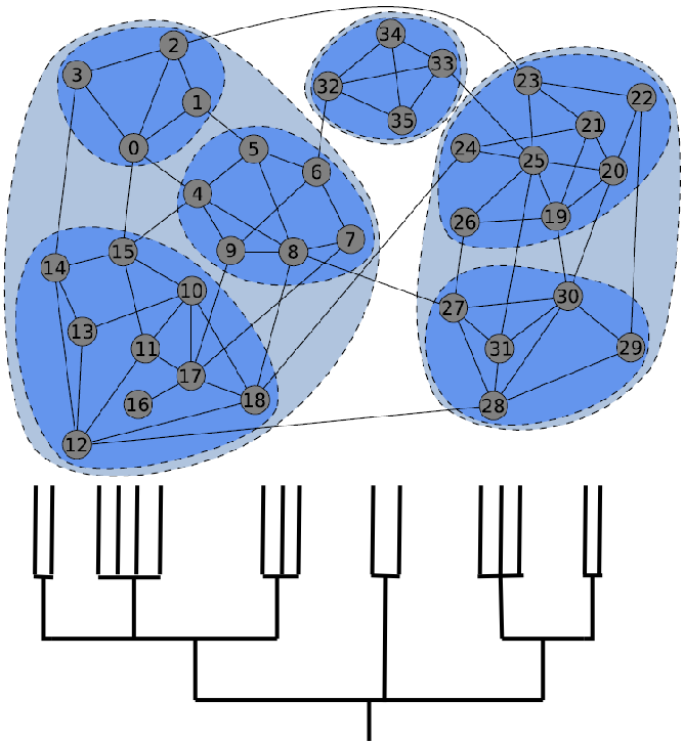

FIG. 6 – *Exemple de structure de communautés dans un graphe : deux partitions en communautés correspondant à deux échelles différentes sont représentées (source Pons (2007)). Le dendogramme représente cette hiérarchie de communautés.*

On peut définir différentes mesures de similarité entre nœuds du graphe :
– similarités structurelles, basées sur la comparaison du voisinage des nœuds (e.g. nombre de voisins en commun, ou corrélation de Pearson de la matrice d'adjacence) ;
– mesures basées sur les chemins dans le graphe : par exemple, compter le nombre de chemins différents reliant les deux nœuds (ce qui est lié à la notion de flot).

2.3 Critères de qualité

La mesure de qualité la plus utilisée (Pons (2007)) est la modularité, introduite par Newman et Girvan (2004). On distingue les arêtes internes aux communautés des arêtes reliant des nœuds de communautés différentes. Si on a c communautés, on peut définir D comme la matrice $c \times c$ dont les éléments d_{ij} donnent la proportion d'arêtes reliant des nœuds de la communauté i à ceux de la communauté j. Les termes diagonaux d_{ii} donnent la proportion d'arêtes internes au cluster i parmi toutes les arêtes du graphe. La modularité est alors définie comme

$$M = \sum_i (d_{ii} - (\sum_j d_{ij})^2)$$

La modularité présente l'intérêt de pouvoir être directement optimisée (Newman (2006)).

D'autres mesures de qualité ont été proposées (rapports de liens, de coupes, Kernighan-Lin) ; d'autre part, toutes les mesures classiques utilisées pour le *clustering* (classification non supervisée) sont applicables.

2.4 Approches étudiées

Nombreuses sont les méthodes proposées ces dernières années pour la recherche de communautés. Dans ce article, nous avons choisi de ne présenter que quelques approches représentatives, dont les développements ont permis de proposer des algorithmes applicables aux très grands graphes.

Nous commencerons (section 3) par la méthode proposée en 2003 par Wu et Huberman (2004), et qui peut s'appliquer aussi bien à l'identification de (micro) communautés qu'à la recherche globale de (macro) communautés. Dans la lignée de ses travaux, nous pourrions citer Bagrow et Bollt (2005) (méthode rapide et locale), da Fontoura Costa (2004) (propagation de labels), Fortunato et al. (2004) (méthode plus lente mais peut être plus précise), Clauset et al. (2004).

Nous étudierons ensuite les approches globales basées sur le degré d'intermédiarité (*betweenness*), dans la lignée des propositions de Girvan et Newman (2002); Newman et Girvan (2004). Le lecteur voulant aller plus loin trouvera un panorama (déjà ancien !) sur les méthodes basées sur le degré d'intermédiarité dans (Pinney (2006)).

3 Méthode Wu & Huberman

Nous présentons rapidement ici la méthode proposée par Wu et Huberman (2004).

3.1 Le graphe comme un circuit électrique

3.1.1 Principe général

On considère le graphe comme un circuit électrique (figure 7).

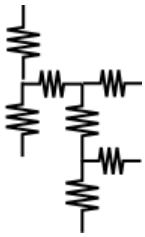

FIG. 7 – *Circuit électrique associé à un graphe.*

Loi de Kirchhoff sur le nœud C exprime le fait que la somme des courants entrants et sortants d'un nœud est nulle :

$$\sum_{i=1}^{n} I_i = \sum_{i=1}^{n} \frac{V_{D_i} - V_C}{R} = 0$$

Si les arcs du graphes sont valués par w_{ij}, on défini la résistance associée comme $R_{ij} = w_{ij}^{-1}$

On fixe la tension en deux nœuds : $V_1 = 1, V_2 = 0$ et on a :

$$V_i = \frac{1}{k_i} \sum_{j=3}^{n} V_j a_{ij} + \frac{1}{k_i} a_{i1} \text{ pour } i = 3, \ldots, n$$

où k_i est le degré du nœud i et a_{ij} la matrice d'adjacence du graphe.

Ce système d'équations linéaires se résout en $O(n^3)$.

3.1.2 Résolution approchée rapide

On peut utiliser une méthode itérative pour résoudre le système d'équations :

1. fixer $V_1 = 1, V_2 = \cdots = V_n = 0$ (en temps $O(V)$)
2. mettre à jour la tension de chaque nœud (en $O(E)$)
3. répéter l'étape 2

La précision après l'étape 2 ne dépend que du nombre d'itérations, pas de la taille du graphe. En pratique, quelques dizaines d'itérations suffisent pour converger : on a donc une complexité en $O(E + V)$.

3.2 Utilisation pour la recherche de communautés

Supposons que l'on veuille découper le graphe en deux communautés. Pour appliquer les principes précédents, il faut choisir deux nœuds de départ (pôles) dans des communautés différentes, calculer les tensions de tous les nœuds et utiliser un seuil pour affecter chacun à une communauté. Wu et Huberman proposent des heuristiques permettant d'aborder ces problèmes.

Pour le choix des pôles, ils proposent de prendre le premier au hasard, puis de sélectionner comme deuxième pôle le nœud le plus éloigné du premier (en général, des nœuds lointains ne font pas partie de la même communauté). Cette recherche s'effectue en complexité $O(E+V)$. On peut itérer cette recherche (en repartant du deuxième pôle) quelques fois pour trouver une paire très éloignée. Une autre approche consiste à choisir plusieurs paires de pôles au hasard, puis à rechercher à chaque fois les communautés et à utiliser un vote pour prendre la décision finale.

La tension est censée chuter assez brutalement lorsqu'on passe d'une communauté à une autre (forte intensité de courant, reflétant la forte intermédiarité des arêtes concernées). Le repérage de cette chute permet de séparer les communautés.

Pour rechercher un découpage en $N > 2$ communautés, il est possible d'utiliser une méthode de *clustering* sur les valeurs des tensions, car les nœuds appartenant à une même communauté ont des tensions voisines.

3.3 Utilisation pour l'identification de communauté

L'identification de communauté consiste à trouver l'ensemble des nœuds appartenant à la même communauté qu'un nœud de départ donné.

Remarquons que la simple utilisation de la distance au nœud de départ sur le graphe ne permet pas de trouver sa communauté. En effet, deux nœuds voisins n'appartiennent pas toujours à la même communauté ; mais surtout, dans le cas des réseaux de type "petit monde", le nombre de voisins à distance 2 ou 3 est souvent très élevé (dans un exemple réel, on a 40% du graphe total à distance 3 d'un nœud donné).

Pour l'identification de communauté, le nœud de départ va être utilisé comme premier pôle du réseau électrique. On va ensuite choisir aléatoirement des voisins à distance 2, et à chaque fois rechercher deux communautés par la méthode exposée ci-dessus. On procède ensuite à un vote. Cette méthode semble donner de bons résultats. Notons cependant que sa complexité dépend (linéairement) de la taille du réseau complet. Il ne s'agit donc pas d'une méthode purement locale.

Note : implémentation

La méthode de Wu & Huberman est implémentée dans la bibliothèque JUNG (et accessible depuis l'environnement GUESS), voir en particulier le composant
`cluster/VoltageClustererL.java` sur
`http://www.cs.duke.edu/csed/harambeenet/guess/src/DukeGuess/edu/uci/ics/jung/algorithms`.

4 Identification de micro-communautés

De nombreuses méthodes ont été proposées pour identifier une communauté à partir d'un nœud donné. Ces méthodes reposent en général sur des approches agglomératives, avec un critère d'arrêt ad-hoc. Ainsi, la méthode proposée par Bagrow et Bollt (2005) se fonde sur l'évolution du "degré sortant" de la communauté en construction. On ajoute de proche en proche des nœuds à la communauté, et on stoppe lorsque le taux de croissance du degré sortant passe sous un seuil arbitraire. L'utilisation de ces méthodes impose toujours de déterminer empiriquement un (voire plusieurs) seuil qui va influencer la taille de la communauté calculée.

La méthode de Wu & Huberman mentionnée plus haut s'applique aussi à l'identification de micro-communautés.

5 Recherche globale de communautés basée sur l'intermédiarité

Nous décrivons dans cette partie différents algorithmes qui effectuent un découpage hiérarchique du graphe en utilisant la notion d'intermédiarité.

5.1 Notion d'intermédiarité

L'intermédiarité (*betweenness*) a été introduite par Newman et Girvan (2004) et est depuis très utilisée[2] Le degré d'intermédiarité d'une arête est proportionnel au nombre de plus courts chemins dans le graphe (pour toutes les paires de nœuds) qui passent par cette arête (figure 8). Cette quantité peut être calculée pour toutes les arêtes en temps $O(mn)$ sur un graphe de n nœuds et m arêtes.

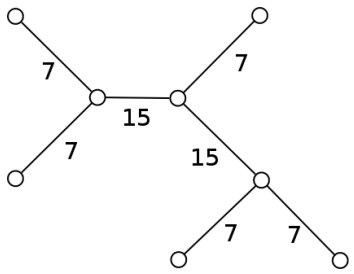

FIG. 8 – *Degrés d'intermédiarité.*

Il existe d'autres définitions du degré d'intermédiarité basées sur les marches aléatoires ou sur les flux qui donnent des résultats similaires.

[2]D'après google scholars, l'article (Newman et Girvan (2004)) a été cité plus de 800 fois.

5.2 Algorithme de Girvan et Newman (GN)

L'algorithme de Girvan et Newman est une méthode divisive fonctionnant selon le principe suivant :

1. Calculer l'intermédiarité de toutes les arêtes ;
2. Supprimer l'arête de plus forte intermédiarité ;
3. Recalculer la intermédiarité entre tous les nœuds affectés par la suppression ;
4. S'il reste des arêtes, recommencer à l'étape 2.

Les auteurs ont montré que cette méthode simple donne de bons résultats. Ils proposent d'utiliser la modularité (voir section 2.3) pour déterminer le nombre optimal de partitions.

Cette méthode est cependant relativement lente ; la complexité dans le pire des cas est en $O(m^2n)$, soit $O(n^3)$ pour un réseau faiblement connecté ($m \propto n$).

5.3 Améliorations de l'algorithme GN

Plusieurs variantes de l'algorithme GN ont été proposées pour le but d'en diminuer la complexité.

5.3.1 Radicchi et al.

Radicchi et al. Radicchi et al. (2004) remplacent l'intermédiarité par le coefficient de clustering, qui est une mesure classique en théorie des graphes, basée sur le nombre de voisins d'un nœud qui sont eux mêmes reliés entre eux. Formellement, le coefficient de clustering du nœud v de degré k s'écrit

$$CC(v) = \frac{2n_v}{k_v(k_v - 1)}$$

où n_v est le nombre de triangles dont v est un sommet. On considère que deux nœuds voisins sont similaires si l'arête qui les relie contribue beaucoup à leurs coefficients de clustering. La similarité s'écrit :

$$S_{cc}(v_i, v_j) = CC_{v_i} + CC_{v_j} - CC'_{v_i} - CC'_{v_j}$$

où CC' est le coefficient de clustering calculé sans l'arête (i, j). On remarquera que S_{cc} est nulle si les deux nœuds ne sont pas voisins dans le graphe.

L'algorithme de Radicchi est donc basé sur un comptage des triangles dans le graphe. Cette mesure locale est plus rapide à calculer que le degré d'intermédiarité. On arrive ainsi à une complexité en $O(m^4/n^2)$, ou $O(n^2)$ pour un réseau faiblement connecté.

5.3.2 Tyler et al.

Tyler et al. (2005) ont présenté en 2003 un algorithme de recherche qui conjugue l'approche GN (intermédiarité) avec une détection locale des communautés, basée sur une heuristique. Le principe général de l'algorithme est le suivant :

1. Rechercher les composantes connexes du graphe ;
2. Pour chaque composante, vérifier si elle forme une communauté ;
   – si oui, la retirer ;

– si non, supprimer les arêtes de forte intermédiarité, jusqu'à couper en deux la composante (deux méthodes différentes sont utilisées pour recalculer les intermédiarités).

3. Répéter l'étape 2 jusqu'à ce que tous les nœuds aient été affectés à des communautés.

L'identification de communauté (étape 2) est heuristique : tout groupe de moins de 6 nœuds est considéré comme une communauté ; pour les groupes plus grands, on calcule l'intermédiarité entre les nœuds feuilles et le reste du graphe et on applique un seuil.

Cette approche est moins précise que les précédentes.

5.3.3 Clauset, Newman et Moore (CNM)

Clauset et al. (2004) ont proposé un algorithme rapide de recherche de communautés, que nous désignerons par l'acronyme CNM. La complexité de cet algorithme lui permet de traiter de grands graphes car elle est en $O(md \log n)$, où d est la profondeur du dendogramme qui décrit la structure de communautés.

Dans les graphes faiblement connectés (*sparses*), on a $m \propto n$. Les réseaux sans échelle, souvent rencontrés dans les applications, ont une structure de communautés à toutes les échelles, et on a donc $d \propto \log n$. La complexité serait donc en pratique de l'ordre de $O(n \log^2 n)$, mais ce point est contesté, on n'observe pas toujours ce comportement (voir Wakita et Tsurumi (2007)).

L'algorithme CNM a été testé par ses auteurs sur le réseau formé par les articles vendus en ligne par Amazon (chaque article est lié au dix artciles les plus fréquemments achetés avec lui). Ce réseau comportait à l'époque 409 687 nœuds et 2 464 630 arêtes.

5.4 Algorithme de Wakita et Tsurumi

Wakita et Tsurumi (2007), après avoir observé que l'algorithme CNM était trop lent pour traiter facilement des réseaux de plus 500 000 nœuds, ont proposé une série d'améliorations basées sur différentes heuristiques permettant d'accélérer significativement la recherche (en particulier sur l'équilibrage des tailles de communautés, pour diminuer le facteur $\log d$ intervenant dans la complexité de CMN).

5.5 La méthode de Louvain

La méthode de Louvain a été proposée par Blondel et al. (2008) et améliore l'algorithme de Wakita et Tsurumi. Il s'agit aussi d'une méthode de recherche globale optimisant la modularité de la décomposition en communautés. Cette méthode élégante donne de meilleurs résultats que la précédentes (en terme de modularité) tout en réduisant significativement les temps de calculs.

La méthode, de type agglomérative, procède en répétant deux phases. On part d'un graphe dans lequel chaque nœud est associé à une communauté unique (il y a donc autant de communautés que de nœuds), et on va progressivement fusionner ces communautés, comme indiqué ci-dessous.

	Karate	Arxiv	Internet	Web nd.edu	Phone	Web uk-2005	Web WebBase 2001
Nodes/links	34/77	9k/24k	70k/351k	325k/1M	2.6M/6.3M	39M/783M	118M/1B
CNM	.38/0s	.772/3.6s	.692/799s	.927/5034s	-/-	-/-	-/-
PL	.42/0s	.757/3.3s	.729/575s	.895/6666s	-/-	-/-	-/-
WT	.42/0s	.761/0.7s	.667/62s	.898/248s	.56/464s	-/-	-/-
Our algorithm	.42/0s	.813/0s	.781/1s	.935/3s	.769/134s	.979/738s	.984/152mn

TAB. 1 – *Résultats publiés par Blondel et al. (2008) : recherche de communautés sur divers jeux de données standards, de tailles croissantes. La première ligne indique la taille du graphe (nombre de nœuds et de liens). Les quatre méthodes mesurées sont : CNM (Clause, Newman et Moore), PL (Pons et Latapy), WT (Wakita et Tsurimi) et, sur la dernière ligne, la méthode de Louvain. Chaque case donne la modularité obtenue et le temps de calcul nécessaire (où rien si temps supérieur à 24 heures).*

Principe de l'algorithme de "Louvain" (Blondel et al. (2008)) :

Répéter :

1. *Optimisation locale de la modularité* : on cherche parmi les voisins de tous les nœuds ceux qui permettrait de gagner en modularité en les changeant de communauté. Cette phase s'arrête lorsqu'on atteint un maximum local de la modularité (on ne peut plus l'augmenter en attribuant un nœud à une communauté voisine). Notons que l'efficacité de l'algorithme repose en bonne part sur l'expression de la variation de la modularité lorsqu'on change l'attribution d'un nœud.
2. *Fusion* : on construit un nouveau graphe dont les nœuds sont les communautés trouvées à l'issue de la phase précédente. Le poids d'une arête liant deux communautés dans ce nouveau graphe est donné par la somme des poids des arêtes reliant des nœuds appartenant à ces deux communautés dans le graphe d'origine.

Jusqu'à stabilisation (plus de changement des communautés, maximisation de la modularité).

Note : des implémentations C++, matlab et Python de la méthode de Louvain sont disponibles sur `http://findcommunities.googlepages.com`.

6 Récapitulatif

Le tableau 6 montre les résultats d'une comparaison expérimentale des principaux algorithmes discutés ici. Il en ressort clairement que la méthode de Louvain est la plus intéressante, tant en terme de qualité de la décomposition obtenue qu'en temps de calcul. Les temps de calculs de cette méthode varient de manière quasi-linéaire avec la taille du graphe à traiter, ce qui autorise le traitement des grands réseaux sociaux susceptibles d'être rencontrés dans les applications modernes.

7 Conclusion

Ces dernières années ont vu apparaître de nouvelles méthodes de détection de communautés, suffisamment rapides pour être appliquée à de très grands graphes. Ainsi, la méthode de Louvain que nous avons décrite permet de segmenter un réseau de plusieurs dizaines de millions de nœuds en quelques minutes sur un PC de bureau. Ces méthodes ouvre la porte à l'utilisation de la notion de "communauté" pour la fouille de données sur les graphes produits par les applications modernes (bases de données clients, applications Web 2.0).

Remerciements

Ce travail a été réalisé dans le cadre du projet CADI *Composants Avancés pour la Distribution*, financé par l'Agence Nationale de la Recherche (ANR).

Références

Albert, R., H. Jeong, et A.-L. Barabasi (2000). Error and attack tolerance of complex networks. *Nature 406*(6794), 378–382.

Bagrow, J. et E. Bollt (2005). A local method for detecting communities. *Physical Review E 72*, 046108.

Barabasi, A.-L. (2002). *Linked*. Perseus Publishing.

Blondel, V., J. Guillaume, R. Lambiotte, et E. Mech (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment P10008*, 1742–5468.

Clauset, A., M. E. J. Newman, et C. Moore (2004). Finding community structure in very large networks. *Physical Review E 70*, 066111.

da Fontoura Costa, L. (2004). Hub-based community finding. arXiv :cond-mat/0405022v1.

Fortunato, S., V. Latora, et M. Marchiori (2004). A method to find community structures based on information centrality.

Girvan, M. et M. E. Newman (2002). Community structure in social and biological networks. *Proc Natl Acad Sci USA 99*(12), 7821–7826.

Kernighan et Lin (1970). An effective heuristic procedure for partionning graphs. *Bell System Technical Journal 29*, 291–307.

Leskovec, J. (2007). Diffusion and cascading behavior in networks. In F. Fogelman-Soulié et al (Eds.), *NATO ASI Workshop on Mining Massive Data Sets for Security*, pp. 169–185. IOS Press.

Newman, M. E. J. (2006). Modularity and community structure in networks. *PNAS 103*(23), 8577–8582.

Newman, M. E. J. et M. Girvan (2004). Finding and evaluating community structure in networks. *Physical Review E 69*(026113).

Peter J. Carrington, John Scott, S. W. (2005). *Models and Methods in Social Network Analysis (Structural Analysis in the Social Sciences)*. Cambridge University Press.

Pinney, J. &. W. D. (2006). Betweenness–based decomposition methods for social and biological networks. In P. B. K. &. R. W. E. S. Barber (Ed.), *Interdisciplinary Statistics and Bioinformatics : Proceedings*, pp. 87–90. Leeds University Press.

Pons, P. (2007). *Détection de communautés dans les grands graphes de terrain*. Ph. D. thesis, Université Paris 7.

Pothen, A., H. D. Simon, et K.-P. Liou (1990). Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl. 11*(3), 430–452.

Radicchi, F., C. Castellano, F. Cecconi, V. Loreto, et D. Parisi (2004). Defining and identifying communities in networks. *Proc Natl Acad Sci U S A 101*(9), 2658–2663.

Tyler, J. R., D. M. Wilkinson, et B. A. Huberman (2005). Email as spectroscopy : automated discovery of community structure within organizations. *The Information Society 21*(2), 143–153.

Wakita, K. et T. Tsurumi (2007). Finding community structure in mega-scale social networks : [extended abstract]. In *WWW '07 : Proceedings of the 16th international conference on World Wide Web*, New York, NY, USA, pp. 1275–1276. ACM Press.

Workshop on Link Analysis (2006). Workshop on link analysis : Dynamics and static of large networks (linkkdd2006).

Wu, F. et B. A. Huberman (2004). Finding communities in linear time : A physics approach. *The European Physics Journal B 38*, 331–338.

Summary

This paper review some recent approaches to find communities in large social networks (millions of nodes). It rapidly recalls some basic notions about social network concepts, then describes different methods proposed to extract micro-communities or global communities. Some experimental results are showed, which demonstrate that the presented methods are perfectly able to cope with the huge graphs produced by modern data applications.