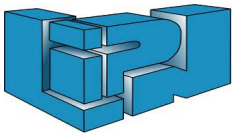


N° d'Ordre : D.U. ...
EDSPIC : ...

Université Paris 13 - Sorbonne Paris Cité
Ecole Doctorale Galilée

THÈSE

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ PARIS 13

Spécialité : ***Informatique***

Détection et évaluation des communautés dans les réseaux complexes

présentée et soutenue publiquement par :

Zied YAKOUBI

04 décembre 2014

Composition du jury :

Directeur de thèse :	Henry SOLDANO	-	Maître de Conférences-HDR, LIPN, Univ. Paris 13
Encadrant :	Rushed KANAWATI	-	Maître de Conférences, LIPN, Univ. Paris 13
Rapporteurs :	Lynda TAMINE-LECHANI	-	Professeur, IRIT, Univ. Paul Sabatier
	Michel CRAMPES	-	Maître de Conférences-HDR, École des Mines d'Alès
Examinateurs :	Hocine CHERIFI	-	Professeur, LE2I, Univ. de Bourgogne
	Roberto WOLFLER CALVO	-	Professeur, LIPN, Univ. Paris 13

Remerciements

Cette thèse doit beaucoup aux nombreuses personnes qui m'ont encouragé, soutenu et conforté à l'élaboration de ce mémoire de thèse. Qu'elles trouvent dans ce travail l'expression de mes plus sincères remerciements.

Ce travail n'aurait pu aboutir sans mes encadrants. Je tiens à remercier vivement mon directeur de thèse **Henry Soldano**, qui a participé à la réflexion de cette thèse, nourrie par les échanges que j'ai pu avoir au cours des différentes réunions. Je le remercie pour sa confiance, ses relectures, ainsi que pour son aide et ses conseils précieux qui m'ont permis d'avancer au cours de ces années.

Je remercie aussi mon encadrant **Rushed Kanawati**, pour la confiance qu'il m'a accordée, pour sa disponibilité, son appui total tout au long de cette thèse, ainsi que pour l'aide et le temps qu'il m'a consacré et sans qui cette thèse n'aurait pas vu le jour. Ses conseils m'ont permis de surmonter mes difficultés et de progresser sur le plan professionnel et personnel.

Je tiens à exprimer toute ma gratitude au Professeur **Lynda Tamine-Lechani** et à **Michel Crampes**, Maitre de Conférences-HDR, d'avoir accepté d'être rapporteurs de ce travail. Je remercie également le Professeur **Roberto Wolfler Calvo** et le Professeur **Hocine Cherifi** d'avoir accepté d'examiner cette thèse

Je remercie tous les membres du LIPN et l'équipe A^3 pour leur sympathie, notamment tous ceux qui ont été très aimables à mon égard et dont les discussions et les conseils m'ont été très précieux. Je garde en mémoire les discussions avec Younès Bennani, Faouzi Boufarès, Mustapha Lebbah, Kais Klai, Nistor Grozavu et Mohammed Hindawi, qui m'ont permis de progresser dans mes réflexions autour de diverses questions. Je remercie également le cadre familial qui s'était installé dans notre bureau et sur notre palier (au 3ème étage) : Sarra, Hanene, Manisha, Aïcha, Leila, Hanane, Amine, Abdoulaye, Paolo, Aloïs, Ehab et Mohammed.

Un grand merci à ma famille, en particulier mes parents Salem et Zohra, mon frère Oussema et ma sœur Fadwa, pour leur soutien moral et leurs encouragements tout au long de mon cursus scolaire.

Finalement, cette thèse est dédiée à la mémoire de mes grands parents Amor et Dhraifa, qui ont quitté ce monde au moment que j'écrivais mon manuscrit.

Résumé :

Dans le contexte des réseaux complexes, cette thèse s'inscrit dans deux axes : (1) Méthodologie de la détection de communautés et (2) Evaluation de la qualité des algorithmes de détection de communautés.

Dans le premier axe, nous nous intéressons en particulier aux approches fondées sur les *Leaders* (sommets autour desquels s'agrègent les communautés). Premièrement, nous proposons un enrichissement de la méthodologie LICOD qui permet d'évaluer les différentes stratégies des algorithmes fondés sur les leaders, en intégrant différentes mesures dans toutes les étapes de l'algorithme. Deuxièmement, nous proposons une extension de LICOD, appelée it-LICOD. Cette extension introduit une étape d'auto-validation de l'ensemble des leaders. Les résultats expérimentaux de it-LICOD sur les réseaux réels et artificiels sont bons par rapport à LICOD et compétitifs par rapport aux autres méthodes. Troisièmement, nous proposons une mesure de centralité semi-locale, appelée *TopoCent*, pour remédier au problème de la non-pertinence des mesures locales et de la complexité de calcul élevée des mesures globales. Nous montrons expérimentalement que LICOD est souvent plus performant avec TopoCent qu'avec les autres mesures de centralité.

Dans le deuxième axe, nous proposons deux méthodes orientées-tâche, CLE et PLE, afin d'évaluer les algorithmes de détection de communautés. Nous supposons que la qualité de la solution des algorithmes peut être estimée en les confrontant à d'autres tâches que la détection de communautés en elle-même. Dans la méthode CLE nous utilisons comme tâche la classification non-supervisée et les algorithmes sont évalués sur des graphes générés à partir des jeux de données numériques. On bénéficie dans ce cas de la disponibilité de la vérité de terrain (les regroupements) de plusieurs jeux de données numériques. En ce qui concerne la méthode PLE, la qualité des algorithmes est mesurée par rapport à leurs contributions dans une tâche de prévision de liens. L'expérimentation des méthodes CLE et PLE donne de nouveaux éclairages sur les performances des algorithmes de détection de communautés.

Mots clés : Réseaux complexes, Détection de communautés fondée sur les leaders, Identification de leaders, Évaluation orientée-tâche des algorithmes de détection de communautés

Abstract :

In this thesis we focus, on one hand, on community detection in complex networks, and on the other hand, on the evaluation of community detection algorithms.

In the first axis, we are particularly interested in Leaders driven community detection algorithms. First, we propose an enrichment of LICOD : a framework for building different leaders-driven algorithms. We instantiate different implementations of the provided hotspots. Second, we propose an extension of LICOD, we call it-LICOD. This extension introduces a self-validation step of all identified leaders. Experimental results of it-LICOD on real and artificial networks show that it outperform the initial LICOD approach. Obtained results are also competitive with those of other state-of-the art methods.

Thirdly, we propose a semi-local centrality measure, called TopoCent, that address the problem of the irrelevance of local measures and high computational complexity of global measures. We experimentally show that LICOD is often more efficient with TopoCent than with the other classical centrality measures.

In the second axis, we propose two task-based community evaluation methods : CLE and PLE. We examine he hypothesis that the quality of community detection algorithms can be estimated by comparing obtained results in the context of other relevent tasks. The CLE approach, we use a data clustering task as a reference. The PLE method apply a link prediction task. We show that the experimentation of CLE and PLE methods gives new insights into the performance of community detection algorithms.

Keywords : Complex networks, Leader-driven community detection algorithms, Leaders identification, Task-based evaluation of community detection algorithms, Data clustering based evaluation, Link prediction based evaluation.

Table des matières

Liste des tableaux

Table des figures

Introduction générale

Contexte

Réseaux complexes

Dans différents contextes, des entités qui sont en relation sont modélisées par des graphes où les entités sont représentées par des nœuds et les relations par des liens. Ces graphes, appelés *graphes de terrain* ou *réseaux complexes*, sont rencontrés dans de nombreux domaines. Nous citons à titre d'exemples :

— Les réseaux sociaux : les nœuds correspondent à des individus ou des entités sociales et les liens peuvent representer des interactions de natures différentes. De ce fait, on trouve plusieurs types de réseaux sociaux : les réseaux d'accointance (deux individus sont connectés s'ils se connaissent), les réseaux de collaboration où deux individus sont reliés s'ils ont travaillé ensemble comme les réseaux de co-publication scientifique [Newman, 2001], les réseaux d'échanges (les entités sont connectées si elles ont échangé un fichier ou un courrier électronique [Ebel *et al.*, 2002]), les réseaux d'appels téléphoniques (deux individus sont reliés s'il y a eu un appel entre eux), etc.

— Les réseaux biologiques : ce sont des représentations abstraites d'un système biologique et offrent un cadre d'étude du fonctionnement du système plutôt qu'une réalité biologique. L'objectif étant d'offrir un support pour faciliter la compréhension de interactions au sein des processus biologiques. Des réseaux sont couramment utilisés pour modéliser des processus de métabolisme, de régulation de gènes, ou des transductions de signaux [Jeong *et al.*, 2000]. Sur une autre échelle de la règne du vivant nous pouvons parler aussi des réseaux trophiques [Williams et Martinez, 2000] : les espèces d'un écosystème qui sont reliées pour représenter les chaînes alimentaires.

— Les réseaux d'infrastructure : ils représentent des liaisons matérielles entre des objets distribués dans un espace géographique comme les réseaux de transport (les routes entre les villes ou les liaisons aériennes entre les aéroports), les réseaux électriques (les câbles entre les lieux de production et de consommation) ou aussi le réseau physique d'Internet [Faloutsos *et al.*, 1999].

Les graphes de terrain étudiés sont souvent de très grande taille. C'est typiquement le cas des réseaux sociaux en-ligne. Prenons comme exemple les deux réseaux les plus connus, Facebook [1] dont le nombre d'utilisateurs dépasse 900 millions, et Twitter [2] dont le nombre d'utilisateurs actifs dépasse le 271 millions d'utilisateurs échangeant 500 million de tweets par jour [3]. La grande taille des réseaux et leurs taux de croissance rapide pose beaucoup de défis pour le développement et la mise en œuvre d'algorithmes efficaces de fouille et d'exploration des graphes de terrain.

1. `https://www.facebook.com/`
2. `http://twitter.com`
3. `https://about.twitter.com/company`

Structure communautaire

La variété de l'origine des différents réseaux complexes ne les a pas empêché d'avoir des structures similaires. Différentes études ont montré que les graphes de terrain exhibent des propriétés structurelles communes et non triviales [Faloutsos *et al.*, 1999, Albert *et al.*, 1999]. Par exemple, la distance moyenne entre deux nœuds est généralement faible par rapport à la taille globale du graphe [Albert *et al.*, 1999] et le nombre de voisins d'un nœud est distribué selon une loi de puissance [Faloutsos *et al.*, 1999]. Cette propriété implique que peu de nœuds ont beaucoup de voisins, et beaucoup de nœuds possèdent peu de voisins.

Parmi les propriétés communes des graphes de terrains, on trouve également qu'ils sont souvent composés de sous-graphes denses faiblement inter-connectés, appelés *communautés* [Girvan et Newman, 2002]. Les termes *clusters* ou *modules* sont aussi employés pour désigner ces structures. Une communauté regroupe un ensemble d'entités qui peuvent partager des propriétés communes et\ou jouent des rôles similaires. Prenons par exemple un groupe d'amis dans un réseau social, un ensemble de protéines qui ont la même fonction biologique ou un ensemble de pages web traitant des sujets ayant la même thématique. Il peut y avoir aussi des communautés chevauchantes où chaque nœud peut appartenir à plusieurs communautés en même temps. Dans un réseau social, une personne peut être connectée aux membres de sa famille, à des collègues de travail ou aussi à des coéquipiers des activités de loisirs. **Dans cette thèse, nous nous limitons à l'étude de la détection de communautés disjointes**.

Identifier les communautés permet d'avoir une vue *mésoscopique* du réseau complexe et aide à comprendre sa structure. Cela aide également à mener des opérations plus complexes sur les réseaux comme la visualisation, la compression ou la parallélisation de calcul. Beaucoup d'autres applications peuvent se servir de la tâche de détection de communautés. Nous en décrivons quelque unes dans le chapitre 1.

Les algorithmes de détection de communautés ont fait l'objet de nombreux travaux. La majorité d'entre eux consistent à déterminer une partition des nœuds du graphe en maximisant une certaine fonction de qualité. Souvent ces algorithmes utilisent la fonction de la *modularité* [Girvan et Newman, 2002]. L'inconvénient majeur de ce critère est que sa maximisation sur l'ensemble des partitions des nœuds est un problème NP-difficile [Brandes *et al.*, 2008] et il est impossible de calculer l'optimum sur des graphes de grande taille. De ce fait, on ne peut donc qu'employer des méthodes heuristiques pour chercher des solutions approchées. Généralement, on obtient des partitions de qualité similaires et il n'y a aucun critère qui favorise l'une relativement à l'autre. Il existe donc un problème d'indéterminisme. De plus, les méthodes de maximisation de la modularité possèdent une *limite de résolution* qui les empêchent de détecter les petites communautés. En outre, ces algorithmes peuvent détecter des communautés avec une modularité élevée dans des graphes n'ayant aucune structure communautaire comme les graphes aléatoires.

Différentes approches alternatives pour la détection de communautés ont été proposées. On trouve principalement les approches basées sur la propagation des labels [Raghavan *et al.*, 2007], et les approches fondées sur l'identification de leaders. La deuxième famille d'approches constitue une nouvelle tendance dans le domaine de détection de communautés. Elles combinent des méthodes qui partitionnent globalement un graphe en com-

munautés avec des méthodes de calcul de communautés locales égo-centrées. De plus, elles exploitent la nature de la distribution de degré des réseaux réels, où on distingue peu de nœuds bien centraux dans le réseau. Les approches fondées sur l'identification de leaders utilisent généralement des mesures de centralité pour identifier les leaders. La centralité d'un nœud peut être calculée d'un manière locale (par rapport aux voisins) ou d'une manière globale (par rapport à tout le réseau). Les mesures globales sont plus efficaces que les mesures locales, mais ces dernières sont moins coûteuses en temps de calcul. Une mesure semi-locale semble être une alternative intéressante pour réaliser un compromis.

La tâche d'évaluation des communautés est une tâche centrale dans notre domaine. le problème reste d'actualité malgré le grand nombre de travaux menés dans ce contexte [Fortunato, 2010, Labatut, 2012]. La fonction de la modularité est le critère le plus utilisé parmi d'autres critères topologiques pour l'évaluation des partitions. Or, les inconvénients des méthodes optimisant la modularité ont remis en cause la significativité de ce critère dans le processus d'évaluation des partitions. Les chercheurs ont donc recouru à des mesures utilisées dans la classification non-supervisée de la fouille de données classique. Ces mesures consistent à comparer la partition trouvée avec ne partition de référence, appelée aussi *vérité de terrain*. Celle-ci est disponible pour peu de réseaux réels vu qu'elle est déterminée manuellement par des experts. Ainsi, différents travaux ont tenté de proposer des générateurs de réseaux similaires aux réseaux réels. Le problème de ces réseaux artificiels est qu'ils ne couvrent que les caractéristiques étudiées des réseaux complexes réels.

Contributions

Les travaux menés dans cette thèse traitent principalement deux problèmes :

— le problème de la détection de communautés en appliquant des approaches d'identification de leaders,

— et le problème d'évaluation et de comparaison des algorithmes de détection de communautés.

En s'appuyant sur le *framework* Licod, proposé initialement dans [Kanawati, 2011], nous explorons différentes méthodes de détection de communautés fondées sur l'identification de leaders. Nous testons et évaluons différentes fonctions dans toutes les étapes de LICOD. Nous avons étudié, d'une manière expérimentale, sur des réseaux réels et artificiels, la pertinence de chaque mesure proposée par rapport aux caractéristiques des réseaux analysés.

Ensuite, nous proposons une extension itérative de LICOD, appelée it-LICOD. L'objectif de it-LICOD est de valider l'ensemble des leaders identifiés. Elle intègre une étape itérative qui calcule les leaders à l'échelle des communautés détectées. it-LICOD converge quand l'ensemble des leaders se stabilise. Les résultats expérimentaux montrent que l'approche it-LICOD trouve des bons résultats par rapport à LICOD et à d'autres méthodes dans quelques réseaux réels et dans tous les réseaux artificiels.

En troisième lieu, nous proposons une nouvelle mesure de centralité semi-locale appelée *TopoCent*. L'objectif de cette mesure est de faire un compromis entre la non-efficacité des mesures locales et la complexité élevée des mesures globales. Dans TopoCent, nous avons

choisi de calculer l'importance d'un nœud par rapport à trois niveaux de voisinage. Ce choix est motivé par le faible diamètre des réseaux réels. Nous avons également pris en compte l'importance topologique des voisins en distinguant les plus importants des moins importants. L'expérimentation de TopoCent avec l'approche LICOD montre qu'elle trouve toujours des bons résultats par rapport aux autres mesures de centralité, y compris les mesures globales.

En quatrième lieu, nous proposons deux méthodes orientée-tâche pour l'évaluation de la qualité des algorithmes de détection de communautés. L'idée est d'estimer la solution des algorithmes en les confrontant à d'autres tâches. La première méthode est orientée classification non-supervisée de données, nous l'appelons CLE. Elle exploite la ressemblance entre la tâche de détection de communautés et la tâche de classification non-supervisée, et la disponibilité de la vérité de terrain de plusieurs jeux de données numériques. Le principe de CLE est de transformer les jeux de données en des graphes benchmarks. Dans la deuxième méthode, que nous la notons PLE, nous avons choisi la tâche de prévision de liens. L'idée est d'évaluer les algorithmes de détection de communautés par rapport à leurs contributions dans la précision des algorithmes de prévision de liens. Nous avons adopté la prévision de liens non-supervisée. Le calcul de score des couples de nœuds est fait via des mesures topologiques intégrant l'information communautaire. L'expérimentation de PLE sur des réseaux bibliographiques dynamiques a montré, à son tour, d'autres performances de certains algorithmes de détection de communautés.

Liste de publications

Les travaux menés dans le cadre de cette thèse ont donné lieu aux publications suivante :

Revue internationale

- Zied Yakoubi, Rushed Kanawati. Licod : A Leader-driven algorithm for community detection in complex networks. Vietnam Journal of Computer Science, Volume(1), Issue 4, pp. 241-256. Springer, 2014.

Chapitre de livre

- Zied Yakoubi, Rushed Kanawati. Leader-driven approaches for community detection in complex networks in Complex Systems. Stéphane Cordier, Nicolas Debarsy, Christel Varin. (Eds.) Cambridge Scholar Publishing, 2014.

Conférences internationales

- Zied Yakoubi, Rushed Kanawati, Applying Leaders Driven Community Detection Algorithms to Data Clustering. The 36th Annual Conference of the German Classification Society on Data Analysis, Machine Learning and Knowledge Discovery (Gfkl'12), Allemagne, 2012.

Conférence nationale

- Zied Yakoubi, Rushed Kanawati, Classification non-supervisée par application d'un algorithme de détection de communautés dans les réseaux complexes. 19ème Rencontre de la société Francophone de Classification (SFC'12), Marseille, 2012.

Ateliers

- Zied Yakoubi. Évaluation des algorithmes de détection des communautés : une méthode orientée-tâche. 3ème Journée thématique. Fouille de grands graphes (JFGG'12), Villetaneuse, 2012.
- Zied Yakoubi, Rushed Kanawati. Data clustering Using Leaders Driven Community Detection Algorithm. Big Data Mining and Visualization. Tours, 2012.

Organisation du mémoire

Ce mémoire est organisé comme suit :

- Le premier chapitre traite le problème de détection de communautés fondée sur les leaders. Nous énumérons les limites des algorithmes classiques basés sur la maximisation de la modularité, et nous présentons un état de l'art des travaux qui portent sur la nouvelle tendance de détection des communautés : les algorithmes guidés par l'identification de leaders.
- Le deuxième chapitre aborde le problème d'évaluation des algorithmes de détection de communautés. Nous décrivons les différentes méthodes existantes, que ce soit celles qui se basent sur la vérité de terrain ou les méthodes topologiques. L'objectif recherché des deux premiers chapitres est d'introduire et de positionner nos contributions au regard de l'existant.
- Dans le chapitre 3, nous présentons les deux approches LICOD et it-LICOD.
- Dans le chapitre 4, nous présentons la mesure de centralité TopoCent.
- Dans le chapitre 5, nous présentons les deux approches CLE et PLE.
- Le chapitre 6 conclut ce mémoire en présentant un bilan général de l'ensemble de nos contributions et en évoquant de nouvelles perspectives de recherche.

Chapitre 1

Communautés centrées leaders

Sommaire

1.1 Introduction

Nous présentons dans ce chapitre le contexte scientifique de nos travaux. Nous commençons par présenter les définitions et les notations utilisées dans ce mémoire. Ensuite, nous exposons la fonction de qualité nommée modularité, qui est fréquemment utilisée dans l'état de l'art comme un critère d'évaluation de la qualité de découpage d'un graphe en communautés. Nous décrivons ainsi quelques méthodes optimisant ce critère avant de souligner les principales limites de l'optimisation de la modularité. Puis, nous exposons les différentes approches alternatives pour la détection de communautés en détaillant les approches fondées sur l'identification de leaders. Enfin, nous concluons ce chapitre par une synthèse de l'existant qui motive nos contributions faisant l'objet du chapitre 3.

1.2 Définitions et notations

Nous considérons dans ce travail des graphes simples non-orientés. Soit $G =< V, E >$ un graphe non orienté. Le tableau 1.1 donne les principales notations employés dans la suite de ce mémoire.

Notation	Description
$G =< V, E >$	Graphe non orienté, V : ensemble de nœuds, E : ensemble de liens
$n = \|V\|$	Nombre de nœuds
$m = \|E\| = e(G)$	Nombre de liens
A_G	Matrice d'adjacence du graphe G
G_X	Sous-graphe de G induit sur l'ensemble $X \subset V$
$\Gamma(v)$	Ensemble de voisins directs d'un nœud v
$d(v) = \mid \Gamma(v)\mid$	Degré d'un nœud v
$d(G) = \frac{1}{n}\sum_{i=1}^{n} d(v_i)$	Degré moyen du graphe G
$dist_G(u, v)$	Distance géodésique entre les nœuds u et v
D_G	Diamètre du graphe G
d_G	Densité du graphe G
$cc(v)$	Coefficient de clustering d'un nœud v
cc_G	Coefficient de clustering du graphe G

TABLE 1.1 – Notations utilisées

Nous rappelons dans la suite quelques définitions de base.

Matrice d'adjacence : l'interconnexion entre les nœuds au sein d'un graphe peut être décrite par une matrice d'adjacence A_G de dimensions $n \times n$. Pour un graphe non pondéré, nous avons $A_{ij} \in \{0, 1\}$, $1 < i, j < n$.

Sous-graphe induit : $G_X = (X, E_X)$ est un sous-graphe induit de $G =< V, E >$ si $X \subseteq V$ et $E_x \subset X \times X \subseteq E$.

Chemin : un chemin est une suite $(v_1, ..., v_k)$ de nœuds de G tels que deux nœuds consécutifs quelconques v_i et v_{i+1} sont connectés par un lien : $\forall i, 0 \leq i \leq k-2, (v_i, v_{i+1}) \in E$. La longueur du chemin correspond au nombre de liens parcourus k.

Distance géodésique : la distance géodésique $dist_G(u, v)$ entre deux nœuds u et v de G est la longueur du plus court chemin entre eux.

Graphe connexe : un graphe est connexe s'il existe un chemin entre tout couple de nœuds.

Composante connexe : une composante connexe d'un graphe est un sous-graphe connexe maximal.

Graphe complet : un graphe est complet si tous les nœuds sont deux à deux connectés entre eux.

Clique : une clique est un graphe complet.

Densité : la densité d'un graphe d est le rapport entre le nombre de liens divisés par le nombre de liens possibles :

$$d_G = \frac{2m}{n \times (n-1)} \tag{1.1}$$

Diamètre : le diamètre D_G d'un graphe G est le plus long des plus courts chemins du G.

Coefficient de clustering : le coefficient de clustering d'un nœud $cc(v)$ est la probabilité que deux nœuds, ayant au moins un voisin commun, soient liés. Le coefficient de clustering d'un nœud est donné par la formule suivante :

$$cc(v) = \frac{\#\triangle}{\#\wedge} \tag{1.2}$$

où $\#\triangle$ est le nombre de triangles dans le graphe et $\#\wedge$ est le nombre de triades.

Le coefficient de clustering de tout le graphe cc_G correspond à la moyenne des valeurs locales :

$$cc_G = \frac{1}{n}\sum_{i=1}^{n} cc(v_i) \tag{1.3}$$

où n est nombre de nœuds de G. $cc_G = 1$ si G est un graphe complet.

1.3 Caractéristiques des graphes de terrain

Les graphes de terrain, dits aussi les graphes de grands réseaux d'interaction, sont au cœur de nombreuses applications comportant des systèmes d'interaction complexes (ex. réseau social, graphe biologique). La plupart de ces graphes exhibent des caractéristiques topologiques communes non-triviales [Faloutsos *et al.*, 1999, Albert *et al.*, 1999]. Certaines de ces caractéristiques sont aussi partagées par les graphes aléatoires [Erdős et Rényi, 1959, Watts et Strogatz, 1998], notamment le cas de deux caractéristiques emblématiques suivantes :

La faible densité : les graphes de terrain ont des densités très faibles où nous observons souvent que le nombre de liens m est proportionnel au nombre de nœuds n.

Effet petit-monde : cette caractéristique est mise en évidence historiquement par l'expérience de Milgram [Milgram, 1967]. Elle exprime le fait que les graphes de terrain ont souvent des diamètres très faibles.

D'autres caractéristiques spécifiques aux graphes de terrain sont les suivantes :

Distribution hétérogène des degrés : les graphes de terrain sont caractérisés par une distribution des degrés très hétérogène : peu de nœuds ont un degré élevé et plusieurs nœuds ont un degré faible [Barabasi et Albert, 1999, Reka et Barabási, 2002]. La figure 1.1 illustre la nature de cette distribution et celle des graphes aléatoires. Cette distribution est approximée par une loi de puissance de la forme de $P(k) = Ck^{-\gamma}$, où C est une constante et $P(k)$ est la probabilité d'un nœud possédant k voisins. Les graphes ayant cette caractéristique sont appelés *graphes sans-échelle* [1].

1. `http://www.network-science.org`

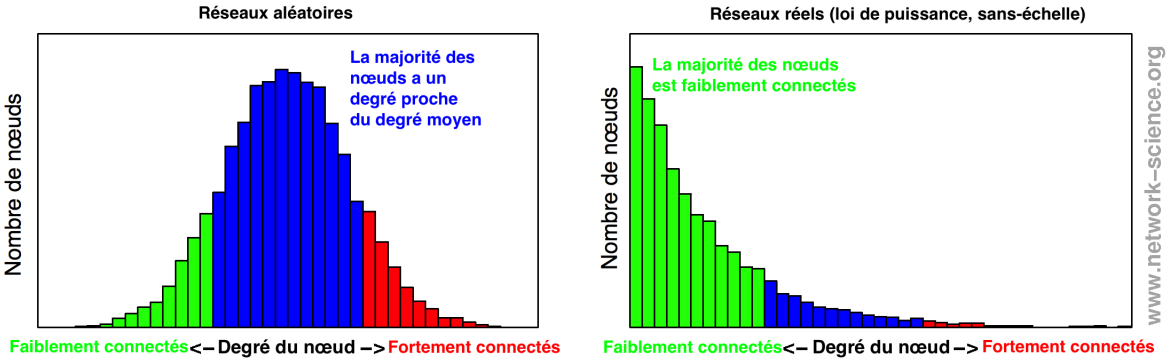


FIGURE 1.1 – Distribution des degrés dans des réseaux réels et aléatoires

Un coefficient de clustering local élevé : cette caractéristique décrit la probabilité de deux nœuds liés ayant au moins un voisin commun, est bien plus élevée que la probabilité de liens entre deux nœuds aléatoirement choisis.

Dans le tableau 1.2, nous donnons les valeurs des caractéristiques décrites ci-dessus pour quelques graphes étudiés dans [Guillaume et Latapy, 2006].

Graphe	n	m	d_G	D_G	γ	cc_G
Internet	75885	357317	1.2e-4	5.80	2.5	0.171
Web	325729	1090108	2.1e-5	7	2.3	0.466
Acteurs	392340	15038083	1.9e-4	3.6	2.2	0.785
Co-publication	16401	29552	2.2e-4	7.18	2.4	0.638
Protéines	2113	2203	9.9e-4	2.4	6.74	0.153

TABLE 1.2 – Exemples de caractéristiques des graphes analysés dans [Guillaume et Latapy, 2006]

Structure communautaire : Les graphes de terrain sont généralement parcimonieux. Ils ont un degré moyen faible et indépendant de la taille du graphe [Reka et Barabási, 2002]. On considère que le nombre de liens d'un graphe est linéaire en le nombre de nœuds du graphe : $m = O(n)$. À l'opposé, la plupart des graphes de terrain exhibent un coefficient de clustering élevé et une densité globale faible. Prenons l'exemple d'un réseau social, cela implique que deux personnes ayant un ami commun ont une grande probabilité de se connaître que deux personnes choisis aléatoirement. On constate donc que la densité varie selon l'échelle ; allant de très forte localement à très faible globalement.

La variation de la densité entre les niveaux locaux et globaux dans les graphes est expliquée par la présence de groupes de nœuds fortement connectés entre eux et faiblement connectés au reste du réseau, appelés *communautés*.

Donner une définition formelle d'une communauté n'est pas une tâche évidente. Toutes les définitions proposées sont restrictives [Coscia *et al.*, 2011]. Néanmoins, la définition la

plus adoptée est celle liée à la topologie du réseau, qui considère qu'une communauté est un sous-graphe dont les nœuds sont densément inter-connectés et faiblement connectés au reste du réseau [Girvan et Newman, 2002]. La figure 1.2 présente un exemple de graphe avec trois communautés.

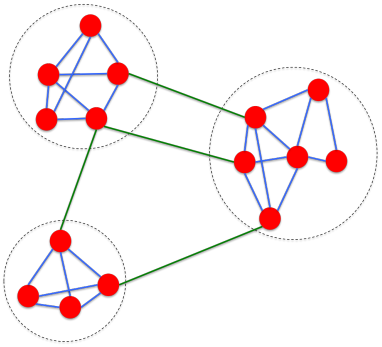


FIGURE 1.2 – Exemple de graphe avec trois communautés entourées par des cercles en pointillés

Les communautés permettent d'avoir une vue macroscopique du système étudié en exposant des groupes de nœuds qui jouent des rôles similaires. Connaître la structure communautaire d'un réseau aide non seulement à mener plusieurs applications cibles mais aussi à aider la réalisation des traitements complexes. Nous décrivons de ce qui suit quelques exemples de ces deux types d'applications.

Compréhension du réseau : la grande taille des graphes de terrain complexifie la tâche de compréhension. Par exemple, Facebook contient plus de 900 millions d'utilisateurs, Google indexe plus d'un trillion d'URLs, le nombre de clients dans le réseau de téléphonie mobile Vodaphone est environ 200 millions. L'identification de communautés permet de réduire cette complexité et de découvrir des relations entre les entités du réseau.

À titre d'exemple, nous citons le travail de [Blondel *et al.*, 2008] qui propose une étude sur un réseau téléphonique entre les clients d'un opérateur Belge. Le réseau a été modélisé par un graphe de 2.6 millions nœuds représentant les utilisateurs. Les liens entre les nœuds sont pondérés par la durée cumulative des appels téléphoniques entre les utilisateurs. La détection de communautés dans ce réseau a produit 2 groupes principaux qui correspondent aux deux communautés francophone et flamande de la population belge (voir figure 1.3).

Parallélisation des calculs : avec l'explosion du volume des données ces dernières années, la structure communautaire sert à réduire la complexité de calcul de certaines opérations sur des grands graphes de terrain. En effet, le partitionnement d'un graphe en communautés permet d'effectuer des calculs séparés moins coûteux sur chaque communauté avant d'agréger les résultats.

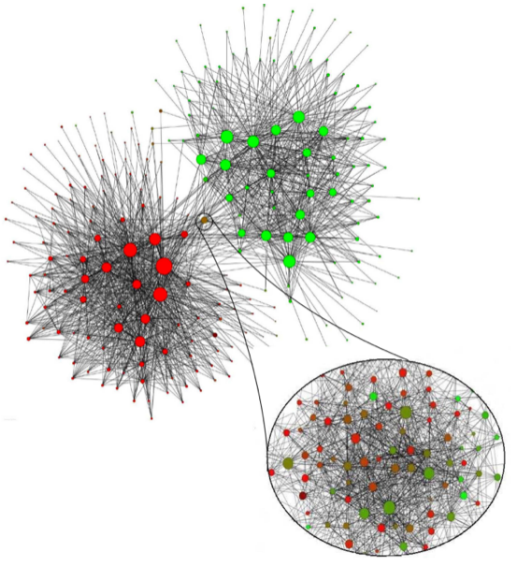


FIGURE 1.3 – Structure communautaire d'un réseau social de communication téléphonique belge [Blondel *et al.*, 2008]. Les points colorés indiquent les sous-communautés au niveau hiérarchique. La coloration du rouge au vert représente la fraction des langues parlées dans chaque communauté (rouge pour les francophones et vert pour les flamands). Les deux grandes communautés sont linguistiquement homogènes, avec plus de 85% des personnes qui parlent la même langue. La communauté qui se trouve entre les deux groupes (partie zoomée) possède une répartition de langues plus équilibrée

Visualisation : la visualisation constitue une aide précieuse pour la compréhension et l'analyse des réseaux. Néanmoins, les outils de visualisation actuels ne permettent pas de traiter des graphes de terrain à cause de leur taille. La visualisation pourrait se faire facilement au niveau macroscopique à l'aide des communautés. La visualisation via les communautés aide donc à réduire la complexité du graphe d'une manière à qu'il soit interprétable par l'œil humain.

Détection des fonctions inconnues : il est également envisageable d'identifier des fonctions inconnues d'un acteur du réseau en fonction de sa communauté. À titre d'exemple, les communautés dans les réseaux biologiques peuvent correspondre à des modules fonctionnels dans lesquels les membres d'un module fonctionnent de manière cohérente pour réaliser des tâches cellulaires. Une fois que les modules (communautés) de tels réseaux détectés, on peut classer les protéines dont la fonction est inconnue, en déterminant le module auquel elles appartiennent [Palla *et al.*, 2005].

Système de recommandation : les communautés ont fait l'objet de plusieurs travaux dans le cadre de la conception des systèmes de recommandation [Specia et Motta, 2007, Nanopoulos *et al.*, 2009, Benchettara *et al.*, 2010, Schifanella *et al.*, 2010, Papadopoulos *et al.*, 2012, Pujari et Kanawati, 2013]. Dans

le contexte de réseaux sociaux en ligne, la détection de communautés peut servir pour recommander d'établir de nouveaux liens d'amitié, un service fréquemment proposé dans les sites des réseaux sociaux en ligne. Prenons le cas des réseaux bibliographiques, on peut penser à la recommandation de nouvelles collaborations scientifiques [Benchettara *et al.*, 2010, Pujari et Kanawati, 2013]. Pour les réseaux d'achat, le concept de communauté peut être vu comme une généralisation de l'approche classique de filtrage collaboratif [Resnick *et al.*, 1994] où on peut recommander à une personne les produits bien évalués par les membres de sa communauté. Les produits peuvent être aussi regroupés en communautés selon les motifs de leurs achats, ce qui permet de recommander à un client les produits similaires à ce qu'il a aimé d'acheter au passé.

1.4 Approches de détection de communautés

Formellement, une partition de graphe est une division de l'ensemble des nœuds du graphe en des ensembles disjoints et non vides : soient $G = (V, E)$ un graphe, $P = \{c_1, ..., c_p\}$ est une partition du G en p communautés si $\bigcup_{i=1}^{p} c_i = V$; $\forall c_i \in P, c_i \neq \emptyset$; et $\forall i, j, i \neq j \Rightarrow c_i \cap c_j = \emptyset$.

De nombreux algorithmes de détection de communautés ont été proposés, on trouve des méthodes basées sur la marche aléatoire, des méthodes spectrales, des méthodes basées sur la diffusion, etc. (Des études détaillées de l'état de l'art dans ce domaine sont présentées dans [Fortunato, 2010, Plantié et Crampes, 2013]). Ces algorithmes s'appuient uniquement sur les données structurelles des réseaux sans considérer tous les autres attributs particuliers aux nœuds. Depuis le travail fondateur de [Girvan et Newman, 2002], la majorité des approches consistent à trouver une partition des nœuds du graphe tout en optimisant un critère de qualité d'un partitionnement, défini à partir de la structure du graphe. Le critère le plus utilisé est *la modularité* [Girvan et Newman, 2002]. Nous décrivons par la suite le principe de ce critère.

1.4.1 Modularité : qualité d'une partition

Soit $\frac{\sum_c e_c}{m}$ la fraction de liens qui se trouve à l'intérieur de la communauté c. Selon la définition de la communauté de [Girvan et Newman, 2002], cette fraction doit être élevée. Or, la valeur maximale de $\frac{\sum_c e_c}{m}$ est atteinte que lorsque l'ensemble du réseau est considéré comme une seule communauté ($P = \{V\}$), parce que tous les liens se trouvent dans cette communauté, c.à.d. $\frac{\sum_c e_c}{m} = 1$. Pour y remédier, [Girvan et Newman, 2002] ont proposé la fonction de modularité. Elle est fondée sur le fait que les réseaux aléatoires ne disposent pas de structure communautaire. Ainsi, pour une partition trouvée, on souhaite non seulement que $\frac{\sum_c e_c}{m}$ soit élevée mais aussi que la même partition sur un graphe aléatoire donne une faible valeur de $\frac{\sum_c e_c}{m}$. Afin de comparer la même partition sur G et sur un graphe aléatoire, il faut considérer un modèle de graphe aléatoire ayant le même nombre de nœuds et la même distribution de degrés que G. La fraction des liens au sein des communautés dans un réseau aléatoire se calcule à partir du degré des nœuds de la manière suivante : pour une partition P, si un lien choisi au hasard, la probabilité a_c qu'une de ses extrémités de celui-ci mène à la communauté c, est égale au nombre de liens ayant une extrémité dans la communauté

c divisé par le nombre total de liens du réseau m. La probabilité qu'un lien soit connecté à un nœud dans la communauté c est la proportion de demi-liens dans c, soit la somme des degrés des nœuds de c divisée par deux fois le nombre de liens : $a_c = \frac{\sum_{i \in c} d(n_i)}{2m}$. La probabilité que les deux extrémités d'un lien soient dans la communauté c est donc a_c^2. Ainsi, la modularité est définie par :

$$Q(P) = \sum_{c \in P} (e_c - a_c^2) \tag{1.4}$$

D'une autre manière, la modularité est exprimée comme suit : soit A la matrice d'adjacence du graphe G dont les éléments A_{ij} sont les poids des liens entre les nœuds i et j, et valent 0 ou 1 dans le cas d'un graphe non-pondéré. L'équation 1.4 peut alors être reformulée par :

$$Q(P) = \frac{1}{2m} \sum_{c \in P} \sum_{i,j \in c} (A_{ij} - \frac{d(i)d(j)}{2m}) \tag{1.5}$$

La modularité est comprise entre -1 et 1. Une bonne modularité a une valeur positive et la qualité de la partition augmente avec la modularité. Bien que plusieurs fonctions de qualité aient été proposées pour évaluer la qualité d'une partition d'un graphe donné [Mancoridis *et al.*, 1998], la modularité reste la fonction la plus utilisée.

1.4.2 Détection de communautés par maximisation de modularité

La maximisation de la modularité est un problème NP-difficile [Brandes *et al.*, 2008]. Des méthodes d'optimisation sont proposées pour calculer, en temps et en espace polynomiaux, des partitions que l'on espère proches de l'optimum. Des méthodes d'optimisation directe utilisant les techniques de l'algorithmique génétique [Li et Song, 2013, Pizzuti, 2012, Cai *et al.*, 2011], du recuit-simulé [Reichardt et Bornholdt, 2006, Guimera *et al.*, 2004] ou de l'optimisation extrémale [Duch et Arenas, 2005], ont été proposées. Cependant, les heuristiques les plus appliquées sont fondées sur le principe de la classification hiérarchique. Deux approches contradictoires sont largement expérimentées :

— Les approches agglomératives (ou ascendantes) selon lesquelles on part de la partition atomique (ensemble des singletons), et on fusionne deux communautés à chaque itération. Les communautés à fusionner sont celles qui promettent une modularité maximale.

— Les approches séparatrices (ou descendantes) dans lesquelles on part du graphe entier. À chaque itération, on cherche à scinder une communauté en deux, de sorte à maximiser la modularité.

Dans les deux cas, l'algorithme produit une hiérarchie de communautés. On retient généralement la partition qui a la modularité maximale. Nous décrivons ci-dessous quelques algorithmes de ces deux familles sans exhaustivité.

Algorithme de Girvan-Newman (GN)

C'est une méthode divisive proposée par [Girvan et Newman, 2002]. Elle s'appuie sur la mesure de centralité d'intermédiarité des liens comme une heuristique de sélection des liens à supprimer. Étant donné un lien (u, v), cette centralité est calculée par :

$$C_b((u,v)) = \sum_{i,j \in n, i \neq j} \frac{\sigma_{ij}((u,v))}{\sigma_{ij}} \tag{1.6}$$

où $\sigma_{ij}((u,v))$ est le nombre de plus courts chemins allant de v_i à v_j passant par le lien (u,v) et σ_{ij} le nombre total de plus courts chemins allant de v_i à v_j. L'objectif de cette centralité d'intermédiarité est de repérer les liens centraux du graphe qui se ressemblent à des ponts connectant les communautés. Cela est naturellement vrai puisqu'un lien inter-communautaire serait traversé par une fraction élevée de plus courts chemins entre les nœuds appartenant à différentes communautés. Considérons un graphe G, l'algorithme itère m fois en coupant à chaque fois le lien qui a le maximum d'intermédiarité. Cela permet de construire une hiérarchie de communautés dont la racine est l'ensemble du graphe et les feuilles sont les communautés composées de nœuds isolés. La partition qui a la modularité la plus élevée est retenue. Cet algorithme nécessite un calcul itératif de la centralité d'intermédiarité ce qui le rend coûteux en temps avec une complexité $O(m^2 n)$. Il est donc difficile de l'appliquer sur les grands graphes.

FastGreedy

La méthode FastGreedy est une approche agglomérative [Clauset *et al.*, 2004]. C'est une version optimisée de l'approche proposée dans [Newman, 2004] avec une complexité de l'ordre de $O(n \log^2 n)$. Au départ, chaque nœud est dans sa propre communauté, puis, à chaque étape, l'algorithme regroupe deux communautés afin de maximiser le gain de la modularité.

Walktrap

Walktrap est une méthode agglomérative proposée dans [Pons et Latapy, 2004]. Elle est fondée sur l'idée qu'une marche aléatoire partant d'un nœud a plus de probabilité de rester piégée pendant un certain temps dans la communauté du nœud de départ. Supposons que nous effectuons une marche aléatoire courte sur le graphe partant d'un nœud v, alors la probabilité d'accéder à chacun des voisins de v en une étape est $\frac{1}{|\Gamma(v)|}$. On peut donc calculer de même manière, la probabilité de se trouver au nœud j en partant de i après avoir effectué aléatoirement k pas. Cette probabilité permet de définir une distance entre les paires des nœuds du graphe dans laquelle deux nœuds u et v sont proches si leurs vecteurs de probabilité d'atteindre les autres nœuds, sont similaires. L'algorithme de la méthode Walktrap utilise ces probabilités calculées pour toutes les paires de nœuds, afin de partitionner le graphe par le biais d'une méthode de clustering hiérarchique. Commençons par n communautés ne contenant chacune qu'un seul nœud, l'algorithme cherche les deux communautés les plus proches, les fusionne, recalcule les distances, puis effectue une

nouvelle fusion et ainsi de suite, jusqu'à n'obtenir qu'une seule communauté recouvrant tout le graphe. La complexité de Walktrap est à l'ordre de $O(n^2 \log n)$.

Méthode de Louvain

C'est une méthode agglomérative proposée par Blondel et al. [Blondel *et al.*, 2008]. Elle implante une méthode d'optimisation gloutonne locale de la modularité. À l'état initial, chaque nœud est affecté à une communauté différente des autres. La méthode applique ensuite une itération de successions de deux phases :

1. Phase d'affectation des nœuds : pour chaque nœud x, on évalue le gain de la modularité si on le déplace dans la communauté de ses voisins directs. On déplace x dans la communauté du voisin qui maximise le gain de la modularité. x reste dans sa communauté si aucun gain n'est trouvé.
2. Phase de compression : on compresse le graphe obtenu en remplaçant chaque communauté par un seul nœud. Deux nœuds c_x, c_y dans le nouveau graphe sont liés par un lien s'il existe un lien entre un nœud de la communauté représentée par c_x et un nœud de la communauté représenté par c_y. Le poids de liens entre deux communautés est égal à la somme des poids des liens reliant des nœuds de deux communautés.

La méthode de Louvain s'arrête s'il y a plus de possibilités de réaffectation de nœuds ou si un maximum de modularité est atteint. La complexité théorique de cette approche n'est pas étudiée, mais d'une manière expérimentale, elle est évaluée à $O(n \log n)$, ce qui fait de Louvain, la méthode la plus rapide pour la détection des communautés.

La méthode spectrale de Newman

C'est une méthode séparatrice à base des vecteurs propres (que nous notons EV ; Eigenvectors) [Newman, 2006a]. Dans ce travail, la modularité est exprimée en fonction des vecteurs propres du graphe, et elle devient "la matrice de modularité". [Newman, 2006a] propose une méthode spectrale pour trouver la partition du graphe. Cette méthode commence par effectuer une décomposition spectrale de la matrice de modularité, puis elle répète cette étape pour chaque sous-graphe. Quand il n'y a aucun gain de modularité, la méthode EV quitte le sous-graphe indivisible correspondant. Cette méthode se termine lorsque l'ensemble du réseau est décomposé en des sous-graphes indivisibles. La complexité de cette méthode est de l'orde de $O(n^2 \log n)$.

1.4.3 Limites de l'optimisation de la modularité

Les approches fondées sur l'optimisation de la modularité font implicitement les hypothèses de travail suivantes :

1. La meilleure décomposition en communautés d'un graphe est celle correspondant à la modularité maximale ;
2. Si un réseau a une structure communautaire alors on peut trouver une partition pour laquelle la modularité est maximale ;

3. Pour un réseau à structure communautaire, les partitions correspondant à des grandes valeurs de modularité sont structurellement similaires.

Or, des études récentes ont montré que l'optimisation de la modularité a quelques limites qui rendent ces trois hypothèses fausses. Nous décrivons dans ce qui suit ces limites.

Limite de résolution

[Fortunato et Barthélemy, 2007] ont relevé que toutes les méthodes de détection de communautés implémentant l'optimisation de la modularité, ont une *limite de résolution* qui ne leur permet pas de détecter les petites communautés dans les grands réseaux. Ils ont prouvé que les algorithmes de maximisation de modularité ne peuvent pas trouver les communautés ayant moins de $\sqrt{\frac{m}{2}}$ liens, où m est le nombre de liens du réseau. Or, dans plusieurs réseaux, les communautés de petite taille marquent une forte présence pour des raisons biologiques et sociologiques [Dunbar, 1998]. Cette taille est bien inférieure à la limite de résolution inhérente à de nombreux grands réseaux. Les deux figures 1.4 et 1.5 montrent deux exemples typiques de graphes dont on peut voir clairement le problème de limite de résolution.

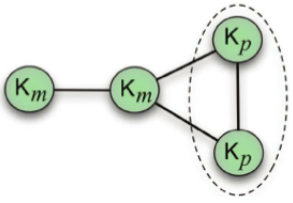


FIGURE 1.4 – Exemple d'un graphe formé de deux cliques de taille m et deux autres cliques de taille p. Si $p \ll$ m (e.g. p=5, m=20), les deux petites cliques sont groupées dans une seule communauté bien qu'elles soient connectées par un seul lien.

Dans une tentative de traitement de ce problème de limite de la résolution, une correction de la fonction de la modularité est proposée dans [Fortunato et Barthélemy, 2007] en ajoutant un paramètre de résolution λ comme suit :

$$Q(P) = \frac{1}{2m} \sum_{c \in P} \sum_{i,j \in c} (A_{ij} - \lambda \frac{d(i)d(j)}{2m}) \tag{1.7}$$

Plus la valeur de λ est élevée, plus les communautés de petite taille seront favorisées par Q puisque la maximisation de Q nécessite la minimisation du terme $\lambda \frac{d(i)d(j)}{2m}$. Inversement, les communautés de grande taille seront favorisées en diminuant λ. Il est à notrer que pour $\lambda = 1$, on obtient la même fonction de modularité initiale. Si cette nouvelle fonction de modularité, appelée modularité multi-résolution, peut être réglée pour explorer de communautés à différentes échelles, elle apporte néanmoins une réponse partielle au problème de la limite de résolution puisque les tailles de communautés dans les réseaux réels sont très hétérogènes et suivent une distribution selon une loi de puissance. D'autre part, on montre dans [Lancichinetti et Fortunato, 2011] que la maximisation de la modularité n'a

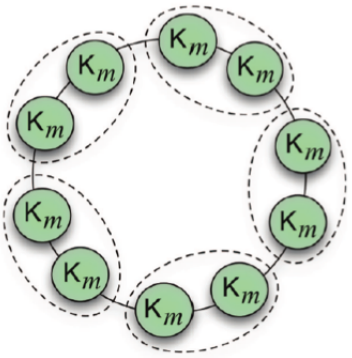


FIGURE 1.5 – Exemple d'un graphe composé de n cliques de taille m. Selon la définition d'une communauté, on devrait identifier chaque clique K_m comme une communauté. Alors que l'optimisation de la modularité les réunit deux par deux, puis trois par trois, etc. Par exemple $n = 10$ cliques de taille $m = 30$ nous avons $Q = 0.650$ si les cliques sont isolées et $Q = 0.675$ si elles sont regroupées deux par deux.

pas seulement tendance à fusionner les petits groupes, mais aussi à éclater des grandes communautés, et il semble impossible d'éviter simultanément les deux problèmes.

Indéterminisme et problème de significativité

L'optimisation de la modularité est un problème NP-difficile [Brandes *et al.*, 2007], et donc on va chercher des solutions proches de l'optimum. Néanmoins, la maximisation de la modularité sur les grands graphes conduit à un grand nombre de maxima locaux qui sont proches de la modularité maximale mais qui correspondent à des partitions très différentes [Good *et al.*, 2010]. Le problème est qu'il n'existe aussi aucun critère pour préférer l'une à l'autre. Cela devient loin d'être pratique dans des grands réseaux. La figure 1.6 illustre un exemple de l'indéterminisme de la maximisation de modularité sur un réseau métabolique.

[Karrer *et al.*, 2008] ont étudié statistiquement la significativité des partitions. Cette significativité est quantifiée en mesurant sa robustesse aux petites perturbations dans la structure du réseau. [Karrer *et al.*, 2008] proposent donc une méthode pour perturber les réseaux et une mesure pour calculer la variation de la structure communautaire obtenue après la perturbation. En utilisant cette méthode, [Aynaud, 2011] ont montré qu'une petite perturbation de graphe peut modifier beaucoup la structure communautaire trouvée par un algorithme basé sur l'optimisation de la modularité.

Modularité élevée dans des réseaux non-modulaires

Les algorithmes de détection de communautés qui maximisent la modularité peuvent identifier des partitions de modularités élevées dans des graphes n'ayant aucune structure communautaire, notamment les graphes aléatoires [Guimera *et al.*, 2004]. Une autre étude [De Montgolfier *et al.*, 2011] montre aussi l'existence des partitions ayant une forte

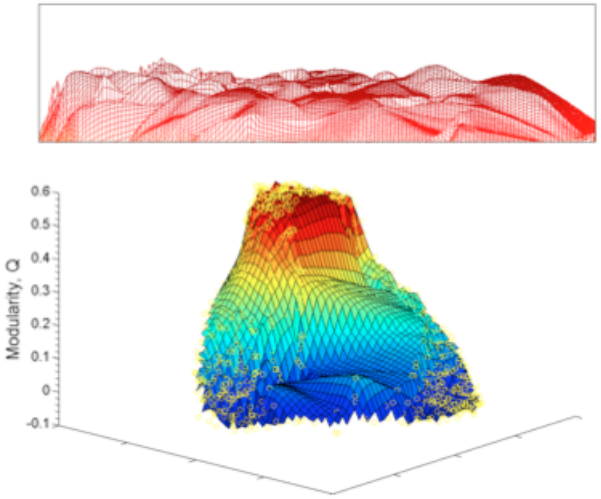


FIGURE 1.6 – La fonction de modularité du réseau métabolique du spirochète Treponema pallidum avec 482 nœuds et 1199 partitions, montrant un plateau de maxima locaux et aucun pic autour de la partition optimale [Good *et al.*, 2010].

modularité dans les graphes qui ne disposent pas de structure communautaire claire.

1.4.4 Approches alternatives

En étudiant les limites des approches fondées sur la maximisation de modularité, il s'avère qu'il est nécessaire d'adopter ou améliorer d'autres concepts pour mieux détecter les communautés. Différentes approches alternatives ont été proposées :

Approches basées sur un modèle dynamique

C'est une catégorie de méthodes qui s'appuient sur un processus dynamique qui se déroule dans le réseau, pour révéler ses communautés. On trouve principalement deux méthodes :

InfoMap : C'est une méthode proposée par [Rosvall et Bergstrom, 2008]. Comme Walktrap, InfoMap exploite le fait qu'un marcheur suivant aléatoirement les liens du graphe, a tendance à rester bloqué dans les communautés. Si on décrit un parcours aléatoire sur le graphe comme une séquence de numéros, qui peut être le numéro du nœud courant ou de la communauté courante, un bon partitionnement consiste à compresser au mieux cette séquence. La qualité du partitionnement représente la quantité d'informations utilisées pour le codage du graphe. Les auteurs optimisent ce critère en utilisant une méthode similaire à celle de Louvain [Blondel *et al.*, 2008]. La complexité de cette méthode est à l'ordre de $O(n \log n)$.

LPA : L'Algorithme de Propagation de Labels (LPA) est une approche basée sur la diffusion [Raghavan *et al.*, 2007]. Elle s'adresse au problème de détection de communautés en utilisant un paradigme de communication qui s'appuie sur l'hypothèse que l'information est échangée de façon plus efficace entre les nœuds d'une même communauté.

L'idée du LPA est simple : un label spécifique l_v est assigné à chaque nœud $v \in V$. Tous les nœuds mettent à jour de façon synchrone leurs labels en sélectionnant le label majoritaire chez les voisins directs. En cas où on a un choix multiple, un label est sélectionné aléatoirement. L'algorithme itère jusqu'à ce qu'il atteigne un état stable où les nœuds ne modifient plus leurs labels. A la fin, les nœuds qui ont la même étiquette sont considérés comme une communauté. La complexité de chaque itération est $\mathcal{O}(m)$, où m est le nombre de liens, si la propagation se fait de façon synchrone. Un avantage majeur de cette approche est son aspect parallélisme massif qui permet à cette méthode de passer à l'échelle. Cependant , il n'y a aucune preuve sur sa convergence. Aussi, en fonction de la topologie locale du réseau, certains nœuds peuvent avoir un problème d'oscillation entre les labels.

Afin d'éviter ce problème, les auteurs proposent également une version asynchrone de LPA [Raghavan *et al.*, 2007]. Dans cette version, les nœuds sont sélectionnés aléatoirement, un à un, pour mettre à jour leurs labels. Malgré que ce modèle réduit le phénomène d'oscillation, il présente quelques limites : premièrement, il durcit l'aspect parallélisme de l'algorithme en créant des dépendances entre les nœuds. Deuxièmement, la sélection aléatoire des nœuds rend l'approche instable où deux exécutions de l'algorithme sur le même graphe donnent deux structures communautaires différentes. Selon [Leung *et al.*, 2009], il semble que cette version asynchrone favorise la détection d'une grande communauté et de plusieurs petites communautés isolées.

Une version semi-synchrone qui tente d'avoir les avantages de ces deux versions a été proposée dans [Cordasco et Gargano, 2012]. L'approche est structurée en deux phases :

— Phase de coloration : chaque deux nœuds voisins auront deux couleurs différentes. La complexité de calcul de cette étape est de l'ordre de : $O(d(G)$.

— Phase de propagation : chaque étape de propagation de label est divisée en k sous-étapes, où k est le nombre de couleurs. À chaque sous-étape c, les labels sont propagés simultanément sur les nœuds qui ont été coloré par la couleur c dans la phase de coloration. Le nombre de sous-étapes par propagation correspond au nombre de couleurs nécessaire pour colorer le réseau, soit $d(G) + 1$.

Le temps de calcul nécessaire pour la convergence est estimé à $O((d(G) + 1)$

Bien que cette version LPA semi-synchrone a traité le problème d'oscillation tout en profitant de l'aspect parallélisme, le nombre de communauté qu'elle génère reste très élevé. La diminution de ce nombre pourrait se faire en propageant seulement le label d'un ensemble de nœuds spécifique qui peuvent représenter les futures communautés. C'est l'idée que les approches centrées graines l'introduisent.

Approches centrées graines

Les approches centrées graines permettent de combiner toutes les approches globales qui décomposent le graphe en des communautés présentées ci-haut, et les ap-

proches fondées sur le calcul de communautés locales centrées sur un nœud cible [Clauset, 2005, Chen *et al.*, 2009a, Kanawati, 2014]. Elles portent sur le calcul de communautés autour d'un ensemble de nœuds spéciaux, dits *graines*, préalablement identifiés. Dans la plupart des cas, les graines cherchées sont des nœuds jouant un rôle important dans les communautés qui vont être détectées, principalement le cas des *approches fondées sur les leaders*. C'est dans ce cadre d'approches, que notre travail se situe.

Dans la section suivante, nous présentons une revue des travaux concernant les approches de détection de communautés fondées sur les leaders.

1.5 Détection de communautés fondée sur les leaders

Les approches fondées sur les leaders constituent une nouvelle tendance dans les approches de détection de communautés. Leur principe est de se concentrer sur la manière dont les relations sont établies et non pas sur la quantification des connexions dans le réseau. Comme la plupart des graphes de terrain sont des graphes-sans-échelle, cela signifie qu'un petit nombre de nœuds possède une multitude de connexions alors qu'un nombre élevé de nœuds possède peu de liens avec les autres nœuds du réseau (Figure 1.1), certains nœuds jouent des rôles plus importants dans le réseau que d'autres. Intuitivement, les approches fondées sur les leaders s'appuyent sur l'idée qu'une communauté se construit autour d'un ensemble de *nœuds leaders*. L'algorithme 1 présente les instructions générales d'une approche fondée sur les leaders. Trois principales étapes ont été identifiées :

1. Identification des leaders
2. Calcul des communautés locales centrées leaders
3. Calcul des communautés globales à partir des communautés locales trouvées dans l'étape 2.

Algorithme 1 Algorithme général d'une approche fondée sur les leaders

1: **Entrée :** un graphe connexe $G =< V, E >$
2: **Sortie :** une partition $\mathcal{C}$
3: **Début**
4: $\mathcal{C} \leftarrow \emptyset$
5: $\mathcal{L} \leftarrow$ **calcul_leaders**(**G**)
6: **pour** $l \in \mathcal{L}$ **faire**
7: $C_l \leftarrow$ **calcul_com_locale**(**l**, **G**)
8: $\mathcal{C} \leftarrow \mathcal{C} + C_l$
9: **fin pour**
10: $\mathcal{C} \leftarrow$ **calcul_com_globale**($\mathcal{C}$)
11: **Retourne** $\mathcal{C}$
12: **Fin**

Chacune des étapes mentionnées ci-haut sont déterminées par différentes techniques dans l'état de l'art. Ces techniques sont classées selon cinq principaux critères. Ces derniers sont détaillés dans la section suivante.

1.5.1 Critères de classification

Cinq critères ont été identifiés pour classifier les méthodes de détection de communautés fondées sur les leaders. Les trois premiers critères concernent l'identification de leaders, le quatrième critère porte sur le calcul des communautés locales, et le dernier critère concerne le calcul des communautés globales. Ces critères sont :

1. **Nature de leaders :** d'un point de vue purement topologique, un leader peut être un seul nœud, un ensemble de nœuds ou un sous-graphe densément interconnecté. La plupart des algorithmes fondés sur les leaders, considère que le leader est le cœur d'une communauté.
2. **Nombre de leaders :** le nombre de nœuds leaders est donné en entrée de l'algorithme. C'est une méthode classique similaire à l'utilisation de l'algorithme de classification non-supervisée k-means [Aggarwal et Reddy, 2014]. Il est généralement difficile de connaître le nombre de communautés à l'avance. La principale heuristique proposée pour calculer automatiquement l'ensemble des leaders est les mesures de centralité ; les nœuds ayant une centralité plus élevée que celle de ses voisins directs, sont considérés comme leaders.
3. **Stratégie de sélection des leaders :** deux principales méthodes ont été utilisées pour la sélection des leaders : la sélection aléatoire et la sélection informée. La sélection aléatoire consiste à choisir des nœuds au hasard comme étant des nœuds leaders dans l'étape initiale puis à les mettre à jour en fonction de leur importance dans la communauté [Reihaneh *et al.*, 2010]. Dans le cas de sélection informée, les leaders peuvent être donnés en entrée de l'algorithme, comme ils peuvent être identifiés par des heuristiques comme les mesures de centralité.
4. **Calcul de communautés locales centrées leaders :** deux approches ont été proposées pour créer les communautés locales autour des leaders : approche d'expansion et approche agglomérative. L'approche d'expansion permet d'élargir les communautés autour des learders. L'inconvénient majeur de l'expansion est qu'elle ne garantit pas de couvrir l'ensemble des nœuds du réseau dans les communautés détectées. Pour remédier à ce problème, les nœuds "outliers" sont ajoutés à la communauté la plus proche. L'approche agglomérative permet à chaque nœud dans le réseau de joindre la communauté du plus proche ensemble de leaders.
5. **Calcul de communautés globales :** une fois que toutes les communautés locales de tous les leaders sont identifiées, une décomposition globale du graphe en communautés est appliquée. Dans la plupart des approches existantes, les communautés globales sont représentées par les communautés locales.

Dans la section suivante, nous détaillons les mesures utilisées dans les deux premières étapes des approches fondées sur les leaders : (1) identification de leaders, et (2) calcul de communautés locales centrées leaders.

1.5.2 État de l'art sur le choix des leaders

L'identification des nœuds leaders est une étape primordiale dans ce genre d'approches de détection de communautés parce que leur performance dépend de la qualité des leaders

trouvés.

La plupart des algorithmes fondés sur les leaders adoptent une sélection informée en utilisant principalement le concept de centralité. Ce concept considère qu'un nœud est leader s'il est le plus central dans le réseau.

Afin de quantifier la notion d'importance d'un nœud dans un graphe, les chercheurs ont proposé plusieurs définitions connues sous le nom de *mesures de centralité* [Koschützki *et al.*, 2005]. Une mesure de centralité est une fonction qui attribue à chaque nœud une valeur positive indiquant à quel point il est "central". La signification de la centralité dépend de la mesure utilisée. Il existe trois catégories de mesures de centralité :

— Centralité locale : elle calcule l'importance d'un nœud en se basant uniquement sur sa connexion avec les voisins directs (ex. centralité de degré).
— Centralité semi-locale : dans cette catégorie, l'importance d'un nœud ne dépend qu'une partie du graphe (ex. le travail de [Chen *et al.*, 2012]).
— Centralité globale : ce sont des centralités qui quantifient l'importance d'un nœud par rapport à tout le graphe. Trois principales mesures ont été réalisées pour cette mesure : la centralité de proximité, la centralité d'intermédiarité et la centralité de vecteurs propres.

Nous décrivons ensuite les différentes mesures de centralité, et nous calculons leurs valeurs sur le graphe exemple présenté dans la figure 1.7.

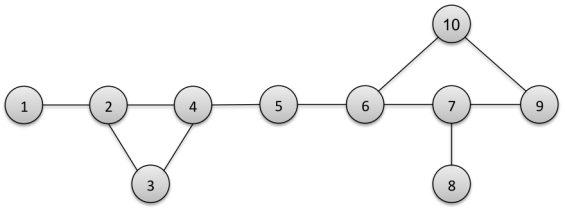


FIGURE 1.7 – Exemple d'un graphe pour l'illustration des mesures de centralités

Soient G un graphe non-dirigé non-pondéré et n le nombre nœuds :

Centralité de degré $C_d(v)$

Elle représente la forme la plus simple et la plus intuitive de la notion de centralité. L'idée de base est que l'importance d'un nœud au sein d'un graphe dépend du nombre de nœuds avec lesquels il est connecté directement. Le calcul de cette mesure se fait d'une manière locale, elle est définie comme la fraction des nœuds incidents au nœud v :

$$C_d(v) = d(v) \tag{1.8}$$

La centralité de degré permet de classer les nœuds d'une manière simple et efficace avec une complexité $\mathcal{O}(n)$, cependant, elle devient moins pertinente dans le cas où un nœud avec peu de voisins importants a un degré d'importance plus élevé que celui d'un nœud ayant plusieurs voisins importants. Le tableau 1.3 montre les valeurs de C_d pour tous les nœuds du graphe exemple.

Centralité de degré										
v	1	2	3	4	5	6	7	8	9	10
$C_d(v)$	1	3	2	3	2	3	3	1	2	2
Rang	9	1	5	1	5	1	1	9	5	5

TABLE 1.3 – Résultat de la centralité de degré sur le graphe exemple

Centralité de proximité $C_c(v)$

La centralité de proximité est connue aussi sous le nom de *closeness*. C'est une mesure de centralité globale fondée sur l'intuition qu'un nœud occupe une position stratégique dans un graphe s'il est globalement proche des autres nœuds du graphe. Par exemple, dans un réseau social, cette mesure correspond à l'idée qu'un acteur est important s'il est capable de contacter facilement un grand nombre d'acteurs avec un minimum d'efforts (l'effort est ici relatif à la longueur des chemins). Formellement, la centralité de proximité est l'inverse de la moyenne des distances géodésiques (i.e. taille du chemin le plus court) vers tous les autres nœuds :

$$C_c(v) = \frac{n-1}{\sum_{u \in E, u \neq v} d_g(u,v)} \tag{1.9}$$

où $d_g(u,v)$ est la distance géodésique entre deux nœuds u et v. La complexité de calcul de cette centralité est : $\mathcal{O}(nlog(n)+m)$, où m est le nombre de liens [Okamoto *et al.*, 2008]. Plus la centralité de proximité est élevée, plus le nœud est proche de l'ensemble des autres nœuds et plus il est central. Le tableau 1.4 montre les scores des nœuds du graphe exemple en terme de C_c.

Centralité de proximité										
v	1	2	3	4	5	6	7	8	9	10
$C_c(v)$	1/34	1/26	1/27	1/21	1/19	1/19	1/23	1/31	1/29	1/25
Rang	10	6	7	3	1	1	4	9	8	5

TABLE 1.4 – Résultat de la centralité de proximité sur le graphe exemple

Centralité d'intermédiarité $C_b(v)$

La centralité d'intermédiarité (ou *betweenness en anglais*) est une mesure de centralité globale. Elle mesure l'utilité d'un nœud dans la transmission de l'information au sein d'un réseau. Un nœud est d'autant plus central qu'il est situé sur beaucoup de plus courts chemins entre d'autres paires de nœuds. Elle est définie formellement par :

$$C_b(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)} \tag{1.10}$$

où $\sigma(s,t)$ est le nombre des plus courts chemins liant s à t, et $\sigma(s,t|v)$ est le nombre de ces chemins passant par v. La complexité de cette centralité est $\mathcal{O}(n.m + (n)^2 log(n))$

[Brandes, 2001]. Le tableau 1.5 montre la centralité d'intermédiarité des nœuds du graphe exemple ainsi que leur classement.

Centralité d'intermédiarité										
v	1	2	3	4	5	6	7	8	9	10
$C_b(v)$	0	8	0	18	20	21	11	0	1	6
Rang	8	5	8	3	2	1	4	8	7	6

TABLE 1.5 – Résultat de la centralité d'intermédiarité sur le graphe exemple

Centralité des vecteurs propres C_{ev}

La centralité des vecteurs propres est nommée aussi centralité spectrale. L'idée est que la centralité d'un nœud dépend de la centralité des nœuds voisins. Il s'agit d'une extension de la centralité de degré dans laquelle on ne donne pas le même poids aux nœuds voisins. Formellement, la centralité spectrale d'un nœud est considérée comme étant dépendante de la combinaison linéaire des centralités de ses nœuds voisins :

$$x_v = \frac{1}{\lambda} \sum_{t \in \Gamma(v)} x_t = \frac{1}{\lambda} \sum_{t \in V} a_{v,t} x_t \tag{1.11}$$

Ici, λ est un réel strictement positif. L'équation 1.11 peut être réécrite sous forme vectorielle comme suit : $x = \frac{1}{\lambda} Ax$ qui est équivalent à $\lambda x = Ax$ avec $x = \{x_{v_1}, x_{v_2}, ..., x_{v_n}\}$ est le vecteur de centralité du vecteur propre de tous les nœuds. Il existe en général, plusieurs valeurs propres pour lesquelles une solution du vecteur propre existe. Cependant, l'exigence que toutes les entrées du vecteur propre soient positives, implique que seule la valeur propre la plus élevée soit retenue, qui est λ. La complexité de la centralité des vecteurs propres est $\mathcal{O}(n^2)$. Dans le reste de ce document, nous notons cette centralité par C_{ev}. Le PageRank de Google est considéré comme une variante de cette centralité. Les résultats de cette centralité sur le graphe exemple sont présentés dans le tableau 1.6.

Centralité de vecteurs propres										
v	1	2	3	4	5	6	7	8	9	10
$C_{ev}(v)$	0.171	0.413	0.363	0.463	0.342	0.363	0.292	0.121	0.221	0.242
Rang	9	2	3	1	5	3	6	10	8	7

TABLE 1.6 – Résultat de la centralité de vecteurs propres sur le graphe exemple

Centralité de degré des voisins $C_l(v)$

C'est une centralité semi-locale qui a été proposée par [Chen *et al.*, 2012]. L'idée est que l'importance d'un nœud dépend de l'importance des nœuds voisins. Au contraire de la centralité des vecteurs propres, le calcul de la centralité n'est pas itératif. Il suffit que

chaque nœud calcule la somme de degré des voisins. Cette centralité esr formulée comme suit :

$$C_l(v) = \sum_{t \in \Gamma(v)} d(t) \tag{1.12}$$

Pour la tâche d'identification de leaders, la centralité de degré des voisins est plus pertinente que la centralité de degré puisqu'elle couvre plus de nœuds, et moins coûteuse que les centralités globales comme la centralité de proximité et d'intermédiarité puisqu'elle a une complexité de l'ordre de $\mathcal{O}(n(k)^2)$ où k est le degré moyen du réseau. Nous donnons les valeurs de cette centralité ainsi que le classement des nœuds dans le tableau 1.7.

Centralité de degré des voisins										
v	1	2	3	4	5	6	7	8	9	10
$C_l(v)$	3	6	6	7	6	7	6	3	5	5
Rang	4	2	2	1	2	1	2	4	3	3

TABLE 1.7 – Résultat de la centralité de degré des voisins sur le graphe exemple

Le tableau 1.8 résume les classements des nœuds du graphe exemple (voir figure 1.7) selon les cinq mesures de centralités expliquées ci-haut. Il est bien évident que pour toutes les mesures de centralité, les nœuds 1 et 8 ont les plus mauvais scores. Il existe une différence significative entre les différents classements. Par exemple, le nœud 3 est considéré comme faiblement inter-connecté lors de l'application de la centralité de proximité et de la centralité d'intermédiarité, alors qu'il se situe au milieu du classement pour la centralité de degré, et bien classé selon les deux centralités vecteurs propres et degré des voisins. En outre, on peut remarquer que la centralité de degré et d'intermédiarité ne différencient pas généralement l'interdépendance des nœuds. Ainsi, par exemple la centralité de degré fournit la même valeur pour les nœuds 2, 4, 6 et 7 mais selon les autres centralités, les nœuds 4 et 6 sont plus centraux que le nœud 7. Ceci est dû au fait que la centralité de degré ne considère que le nombre de voisins directs. Cette limite a été légèrement rectifiée par la centralité des degrés des voisins en plaçant seulement les nœuds 4 et 6 en premier. La centralité d'intermédiarité ne différencie pas entre les nœuds qui ont un seul voisin et les nœuds qui ont des voisins complètement interconnectés (ex. les nœuds 1, 3 et 8).

Malgré quelques corrélations qu'on peut remarquer que ce soit entre la centralité de proximité et la centralité d'intérmédiarité ou entre la centralité de degré, la centralité de degré des voisins et la centralité de vecteurs propres, la centralité d'un nœud reste toujours dépendante de la mesure usée.

Dans le but d'appliquer les mesures de centralité dans l'identification des nœuds leaders, il faut prendre en compte de leurs complexités (cf. Tableau 1.9). Toutes les centralités globales (centralité de proximité, centralité d'intermédiarité et centralité de vecteurs propres) sont coûteuses en temps de calcul ce qui les rend inapplicables sur les graphes de terrain de grande taille. Vue la faiblesse de la pertinence des mesures locales (ex. la centralité de degré), les mesures de centralité semi-locales sont un compromis entre la pertinence et la complexité des mesures locales et globales.

Dans la section suivante, nous étudions les principaux travaux qui abordent la problématique de détection de communautés fondée sur les leaders, en précisant les mesures de

Rang	$C_d(v)$	$C_c(v)$	$C_b(v)$	$C_{ev}(v)$	$C_l(v)$
1	2, 4, 6, 7	5, 6	6	4	4, 6
2			5	2	
3		4	4	3, 6	2, 3, 5, 7
4		7	7		
5	3, 5, 9, 10	10	2	5	
6		2	10	7	
7		3	9	10	9, 10
8		9	1, 3, 8	9	
9	1, 8	8		1	1, 8
10		1		8	

TABLE 1.8 – Classement des nœuds du graphe exemple selon les différentes mesures de centralité

	$C_d(v)$	$C_c(v)$	$C_b(v)$	$C_{ev}(v)$	$C_l(v)$
Complexité	$\mathcal{O}(n)$	$\mathcal{O}(nlog(n)+m)$	$\mathcal{O}(n.m+(n)^2log(n))$	$\mathcal{O}(n^2)$	$\mathcal{O}(n(k)^2)$

TABLE 1.9 – Complexité des mesures de centralités

centralité appliquées.

1.5.3 Composition des communautés

Une fois les nœuds leaders trouvés ou sélectionnés, les communautés doivent se former autour de ces leaders. Les approches qui ont été proposées dans l'état de l'art se divisent en deux grandes familles selon la stratégie adaptée pour créer les communautés : *approches basées sur l'expansion* et *approches agglomératives*.

1.5.3.1 Approches basées sur l'expansion

Les approches basées sur l'expansion considèrent les nœuds leaders comme des noyaux de communautés. Le principe de l'expansion est d'élargir la communauté le plus possible. Ce principe est largement utilisé pour les approches de détection de communautés locales [Xie *et al.*, 2013]. Le but de ce processus est d'itérer sur l'ensemble des nœuds leaders afin de trouver la structure communautaire du graphe. La plupart des approches optimisent une fonction locale qui caractérise la qualité d'une communauté.

L'approche RankRemoval-IS

[Baumes *et al.*, 2005] ont proposé une approche *RankRemoval-IS* composée de deux étapes. Premièrement, un algorithme "RankRemoval" est utilisé pour identifier les nœuds les plus centraux selon la centralité de PageRank, et de les enlever afin de décomposer le graphe en des composantes connexes. Ceci est répété jusqu'à l'obtention des composantes

connexes avec une taille fixée au départ. Les composantes connexes trouvées représentent les leaders ou les cœurs des communautés. Deuxièmement, un processus itératif d'expansion "Iterative Scan (IS)" commence à élargir les communautés autour des leaders trouvés. IS ajoute et supprime des nœuds tout en maximisant la fonction de densité locale définie par [Baumes *et al.*, 2005] comme suit :

$$f(c) = \frac{w_{in}^c}{w_{in}^c + w_{out}^c} \tag{1.13}$$

où $w_{in}^c =$ et w_{out}^c sont respectivement l'intensité interne et externe des liens de la communauté c. Sa complexité est de $\mathcal{O}(n^2)$. La qualité des communautés trouvées dépend de la qualité des cœurs. La limite de cette méthode est que la suppression des nœuds durant la phase d'expansion construit des communautés non-connexes. Pour cette raison, une version modifiée, appelé CIS, a été proposée dans [Kelley, 2009] où la connectivité est vérifiée après chaque itération. Dans le cas où la communauté est divisée en plus d'une partie, seulement celle qui a la plus grande densité est maintenue.

L'approche LCE

[Clauset, 2005] a proposé une autre fonction de qualité appelée *modularité locale* R, qui a fait objet de plusieurs travaux dans ce contexte [Chen *et al.*, 2009b, Chen et Fang, 2012, Pan *et al.*, 2012]. L'expansion basée sur R cherche à ajouter les nœuds qui se trouvent à l'extrémité de la communauté et qui ont plus de connexions avec des nœuds à l'intérieur de la communauté qu'à l'extérieur. La modularité locale quantifie en quelque sorte la netteté de la limite d'une communauté. Soit B l'ensemble des nœuds frontière de la communauté c (cf. Figure 1.8), la modularité locale R est calculée comme suit :

$$R = \frac{B_{in\_edge}}{B_{out\_edge} + B_{in\_edge}} \tag{1.14}$$

où B_{in_edge} est le nombre de liens qui connectent les nœuds frontière à d'autre nœuds de la communauté c, et B_{out_edge} est le nombre de liens connectant les nœuds frontière aux nœuds voisins de la communauté C. Les valeurs de R sont comprises dans l'intervalle $0 < R < 1$. Elle est aussi indépendante de la taille de C.

L'approche LCE [Chen *et al.*, 2009b] est l'une des approches récentes qui adopte la modularité locale R pour l'identification de communautés. Dans cette approche, un leader est un nœud dont le degré est un maximum local. La méthode est composée de quatre étapes : premièrement, on identifie les nœuds leaders grâce à la centralité de degré. Deuxièmement, pour chacun de ces nœuds leaders, on calcule sa communauté en se basant sur le mécanisme d'expansion. Rappelons que ce mécanisme consiste à rajouter un par un et d'une manière itérative à la communauté, les voisins de cette communauté qui maximisent la fonction R. Troisièmement, si les communautés identifiées ne couvrent pas tout le réseau, l'algorithme reprend les deux premières étapes sur le sous-graphe non couvert. Dans le cas où tous les nœuds font partie des communautés trouvées, une étape de fusion unit deux communautés si l'une d'elles contient plus de la moitié de l'autre communauté.

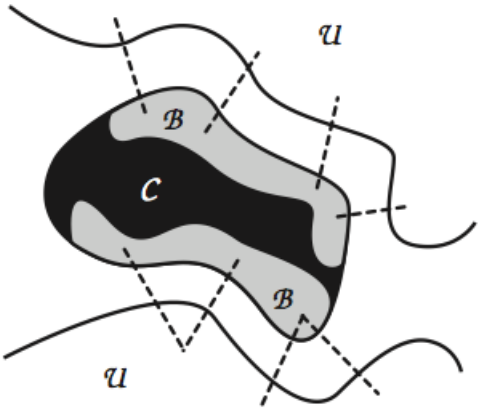


FIGURE 1.8 – Illustration de la division d'un graphe abstrait en une communauté à élargir C, ensemble des nœuds de frontière $\mathcal{B}$ et les liens qui connectent $\mathcal{B}$ au reste du graphe inconnu $\mathcal{U}$ [Clauset, 2005].

Cette approche a une complexité de l'ordre de $O(nd)$, où n est le nombre de nœuds du graphe G et d_G le degré moyen. Par contre, des études ont montré que dans les grands réseaux, la centralité de degré devient moins pertinente parce qu'elle est très locale.

L'approche CPM

La méthode Percolation de Cliques (CPM) [Bollobás et Riordan, 2009] considère qu'une communauté est un chevauchement d'un ensemble de sous-graphes complets, et ainsi, elle détecte les communautés en cherchant les cliques adjacents. CPM commence par identifier toutes les cliques de taille k, puis il construit un nouveau graphe où chaque nœud représente un de ces k-cliques. Deux nœuds sont connectés si les deux k-cliques qu'ils représentent, partagent $k-1$ membres. Les composantes connexes dans le nouveau graphe identifient les cliques qui composent les communautés. La méthode CPM est adaptée aux réseaux contenant des parties très denses. Empiriquement, les bons résultats sont trouvés avec des petites valeurs de k (entre 3 et 6) [Palla *et al.*, 2005, Fortunato et Lancichinetti, 2009]. CFinder[2] est l'implémentation de CPM, sa complexité de calcul est polynomiale dans plusieurs applications [Palla *et al.*, 2005]. Cependant, elle a échoué d'achever la tâche de détection de communautés dans plusieurs grands réseaux.

L'approche HGC

L'approche "Hybrid Graph Clustering" (HGC) [Papadopoulos *et al.*, 2010] est basée sur la notion de $(\mu, \epsilon)-$cœur introduite dans [Xu *et al.*, 2007]. Le cœur d'une communauté est un groupe de nœuds fortement interconnecté. La définition de $(\mu, \epsilon)-$cœur est basée sur

2. `http://www.cfinder.org`

les trois concepts suivants : similarité structurelle, ϵ-voisinage, et l'accessibilité structurelle directe.

La similarité structurelle *sigma* **:** *sigma* calcule la similarité structurelle entre deux nœuds v et w. Soit $G = (V, E)$ un graphe :

$$\sigma(v, w) = \frac{|\Gamma'(v) \cap \Gamma'(w)|}{\sqrt{|\Gamma'(v)| * |\Gamma'(w)|}} \tag{1.15}$$

où $\Gamma'(v)$ est la structure du nœud v, $\Gamma'(v) = \Gamma(v) \cup \{v\}$.

L'ϵ-voisinage : L'ϵ-voisinage d'un nœud v est le sous-ensemble de sa structure $\Gamma'(v)$ qui contient seulement les nœuds qui sont au moins ϵ-similaire au nœud v : $N_\epsilon(v) = \{w \in \Gamma'(v) | \sigma(v, w) \geq \epsilon\}$.

Un nœud v est appelé un $(\mu, \epsilon)-$ cœur si ses $\epsilon-$voisinage contiennent au moins μ nœuds : $\text{Cœur}_{\mu,\epsilon}(v) \Leftrightarrow |N_\epsilon(v)| \geq \mu$.

L'accessibilité structurelle directe : Un nœud v est directement et structurellement accessible à partir d'un (μ, ϵ)-cœur s'il est au moins ϵ-similaire à lui : $\text{DirAcces}_{\mu,\epsilon}(v, w) \Leftrightarrow \text{Cœur}_{\mu,\epsilon}(v) \wedge w \in N_\epsilon(v)$.

Une fois que les (μ, ϵ)-cœurs ont été détectés, il est possible de rattacher les nœuds adjacents aux cœurs à condition qu'ils sont accessibles via une chaîne de nœuds structurellement accessibles entre eux. Le rattachement des nœuds aux cœurs donne comme un résultat un ensemble de graines de communautés. L'expansion des communautés est réalisée via l'optimisation de la fonction locale : modularité de sous-graphe, introduite dans [Luo *et al.*, 2008]. La modularité d'un sous-graphe $S \in V$ est définit par le rapport entre le nombre de liens intra-communautaires $e(S)$ sur le nombre de liens sortant de S :

$$M(S) = \frac{e(S)}{|\{(u, w) \in Ev \in S \wedge w \in V - S\}|} \tag{1.16}$$

Bien que cette approche a introduit un nouveau concept pour la détection de communautés, sa performance reste très liée à la difficulté de l'exploration de l'espace des paramètres (μ, ϵ).

L'approche LICDA

Dans le même type d'approches, [Pan *et al.*, 2012] proposent également une méthode qui part d'un ensemble de leaders et optimise itérativement une fonction objective afin de trouver la structure communautaire. La méthode s'appuie sur la centralité de PageRank pour localiser les nœuds leaders. Les auteurs considèrent qu'une communauté, est représentée par un seul leader. La sélection de ces leaders se fait d'une manière itérative qui suit chaque composition de communauté. L'expansion des communautés est faite via une variante de la modularité locale présentée dans l'equation 1.14. [Pan *et al.*, 2012] introduisent un paramètre pour contrôler la taille des communautés. Après la construction de chaque communauté, une étape de fusion est appliquée sur chaque paire de communautés partageant plusieurs voisins. La méthode itère sur ces trois étapes ; sélection de leader, expansion et fusion, jusqu'à couvrir tout le graphe.

La limite majeure de cette approche est la sélection itérative des leaders qui la rend non-déterministe. En effet, le choix du premier leader candidat à construire sa communauté influe sur la structure communautaire globale trouvée. Également, l'ordre des communautés à fusionner pose le même problème.

Les approches fondées sur l'expansion ont deux limites : la première limite est liée au principe d'expansion lui même. En effet, faire élargir les communautés en optimisant une fonction objective, demande des calculs et des étapes supplémentaire. Souvent, ce genre d'approches introduit une étape de fusion qui sert à regrouper les communautés qui partagent plusieurs nœuds, ce qui augmente la complexité de l'algorithme. Alors que la deuxième limite est le non-déterminisme dû à la sensibilité de ces approches à l'étape d'initialisation.

1.5.3.2 Approches agglomératives

À l'opposé des approches citées dans la section précédente, les approches agglomératives construisent les communautés en partant des nœuds non-leaders, ou nœuds suiveurs, et non des nœuds leaders. L'idée est que chaque nœud non-leader doit calculer séparément à quelle communauté il appartient. Dans ce qui suit, nous détaillons des travaux adoptant cette stratégie.

Approche TopLeaders

TopLeaders [Reihaneh *et al.*, 2010] est une approche inspirée de l'algorithme de clustering k-means. Elle consiste à déterminer les k-nœuds leaders puis associer le reste des nœuds du réseau à ces leaders afin de créer les communautés. Un processus itératif aura lieu ensuite, jusqu'à la convergence. À chaque itération, un nouveau leader est élu pour chaque communauté et réaffecte les autres nœuds aux leaders pour construire des nouvelles communautés. La convergence est atteinte quand l'ensemble de leaders ne change plus, et chaque nœud est affecté à un leader approprié.
Dans l'étape initiale, [Reihaneh *et al.*, 2010] ont évalué différentes heuristiques pour la sélection des nœuds leaders. Une méthode qui a donné des bons résultats, est la suivante : un nœud est un leader s'il a une valeur de centralité dans les top k centralités globales et s'il partage moins de cinq voisins avec les autres leaders. Dans le processus itératif, l'association des nœuds aux leaders se fait via un seuillage sur le nombre de voisins communs entre le nœud et le leader. Les nœuds qui n'ont pas suffisamment de voisins communs avec les leaders, sont considérés comme des "outliers". La réélection d'un nouveau leader d'une communauté consiste à choisir le nœud qui a le maximum de centralités de degré.
Même si cette approche semble simple mais elle possède des limites qui sont liées essentiellement au choix des paramètres notamment le nombre de communautés k. Les experts du domaine sont les seuls qui peuvent estimer ce paramètre. L'approche est également très sensible au nombre de voisins que doivent partager deux leaders. Le nombre utilisé est 5 mais cela pourrait ne pas être toujours une valeur convenable. Ces inconvénients empêchent cette approche, *TopLeaders*, d'être appliquée sur les grands réseaux.

Algorithme Leader-Follower

C'est un algorithme proposé par [Shah et Zaman, 2010]. Il est basé sur l'hypothèse qu'une communauté est une clique. La signification des nœuds leaders et suiveurs (followers) est différente des autres approches : les leaders sont ceux qui connectent les différentes communautés alors que les suiveurs sont les nœuds qui possèdent seulement des voisins d'une même communauté. Pour distinguer les deux ensembles, l'algorithme commence par calculer pour chaque nœud la centralité des plus courts chemins (la somme des plus courts chemins liant ce nœud aux autres nœuds). Les suiveurs sont ceux qui correspondent au maximum local de cette centralité, et les leaders sont les autres nœuds. L'hypothèse est que les suiveurs doivent passer par les leaders pour atteindre les autres nœuds en dehors de la communauté.

Ensuite, l'algorithme passe à l'étape d'affectation de chaque nœud suiveur à un leader. Il trie d'une manière croissante le vecteur des nœuds leaders et il commence à affecter tous les suiveurs au premier leader qui sont voisins avec lui. Il répète cette étape pour l'ensemble des leaders. Dans le cas où un nœud leader n'a aucun suiveur, il sera affecté à un autre leader majoritairement suivi par ses voisins. Ceci permet de réduire le nombre de communautés. La complexité de Leader-Follower est à l'ordre de $O(mn)$. La convergence de l'algorithme Leader-Follower n'est pas garantie car il exige qu'une communauté doit être une clique et ce qui n'est pas toujours le cas dans les graphes de terrain.

1.5.3.3 Synthèse

Nous résumons dans le tableau 1.10 les approches décrites ci-haut en fonction des quatre critères : nature de leaders, nombre de leaders, sélection de leaders et méthode de calcul de communautés. Nous n'avons pas mentionné le critère de calcul de communautés globales car toutes les approches considèrent que ces communautés sont tout simplement les communautés locales trouvées.

L'étude des différentes approches a montré que les méthodes fondées sur l'expansion ne sont pas déterministes puisqu'il n'y a aucune preuve de rattacher tous les nœuds du réseau aux communautés détectées. Quand aux approches agglomératives, l'affectation des nœuds aux communautés est faite d'une manière singulière, c.à.d, qu'elle ignore l'aspect de voisinage entre les nœuds. En effet, comme cela a été montré dans les approches de propagation de labels, les voisins d'un nœud quelconque influent sur son appartenance communautaire. Ainsi, il est nécessaire d'ajouter une étape supplémentaire dans ce genre d'approches afin d'avoir plus d'homogénéité locale.

En outre, la plupart des méthodes sélectionne les leaders d'une manière informée. Cela augmente la vitesse de convergence des méthodes vers les communautés finales.

En pratique, il est difficile d'avoir un nombre de leaders prédéterminé, surtout quand le réseau est de grande taille. C'est pour cette raison que la majorité des approches identifie les leaders d'une manière automatique.

Quant à la nature des leaders et particulièrement dans les approches agglomératives, l'utilisation d'un seul nœud est moins efficace par rapport à un sous-graphe, parce que le nœud leader identifié ne peut pas être forcément au centre de la future communauté. En

effet, le sous-graphe leader, appelé généralement le cœur de la communauté, est plus proche au reste du réseau, ce qui mène à mieux partitionner le réseau. Cependant, les leaders d'une communauté ne sont pas forcément regroupés dans un sous-graphe, ils peuvent occuper des positions éloignées, surtout quand il s'agit des grands réseaux. Il est donc nécessaire de considérer une autre nature de leaders qui fait face à toutes ces limites.

Méthode	**Nature de leaders**	**Nombre de leaders**	**Sélection de leaders**	**Calcul de com.**
RankRemoval-IS	Sous-graphe	Calculé	Informée	Expansion
LCE	Un nœud	Calculé	Informée	Expansion
CPM	Sous-graphe	Calculé	Informée	Expansion
HGC	Sous-graphe	Calculé	Informée	Expansion
LICDA	Un nœud	Calculé	Informée	Expansion
TopLeaders	Un nœud	Entrée	Aléatoire	Agglomérative
Leader-Follower	Un nœud	Calculé	Informée	Agglomérative

TABLE 1.10 – Caractéristiques des méthodes fondées sur les leaders

1.6 Conclusion

Nous avons présenté dans ce chapitre que le problème de détection de communautés est très intéressant pour différentes applications. Nous avons montré que le principe d'optimisation de la modularité utilisé par les méthodes existantes, possède plusieurs limites notamment le problème de significativité des résultats. Nous avons cité les deux principales approches alternatives pour la détection de communautés : les approches basées sur modèle dynamique et les approches centrées graines. Nous avons exposé les limites des méthodes de la première approche particulièrement l'instabilité du LPA, puis nous avons montré notre intérêt pour les approches centrées graines et plus précisément les approches fondées sur les leaders.

L'état de l'art des approches fondées sur les leaders a montré qu'il existe plusieurs stratégies (voir tableau 1.10) et approches intéressantes à exploiter au niveau de la nature des leaders, de la sélection de ces leaders, et de l'agglomération des nœuds autour des leaders. Cela fait l'objet de nos contributions présentées dans les chapitres 3 et 4.

Chapitre 2

Évaluation de la qualité des algorithmes de détection de communautés

Sommaire

2.1 Introduction

Les algorithmes de détection de communautés permettent souvent de calculer des structures communautaires différentes pour un même graphe. Si ces différents algorithmes sont comparés en terme de leur complexité de calcul et d'espace mémoire requis, la qualité des communautés retrouvées reste un indicateur important de la performance de ces algorithmes. Or, l'évaluation de la qualité des communautés est encore une question ouverte malgré le nombre important de travaux dans ce domaine. Trois grandes familles d'approches sont proposées dans l'état de l'art :

— Évaluation orientée communautés
— Évaluation orientée partition
— Évaluation par rapport à une partition de référence (vérité de terrain)

Dans ce chapitre, nous présentons les indices d'évaluation dans ces trois familles d'approches en décrivant leurs caractéristiques. Pour la catégorie d'évaluation par rapport à la vérité de terrain, nous détaillons quatre réseaux réels ayant une partition de référence ainsi que des générateurs de graphes artificiels.

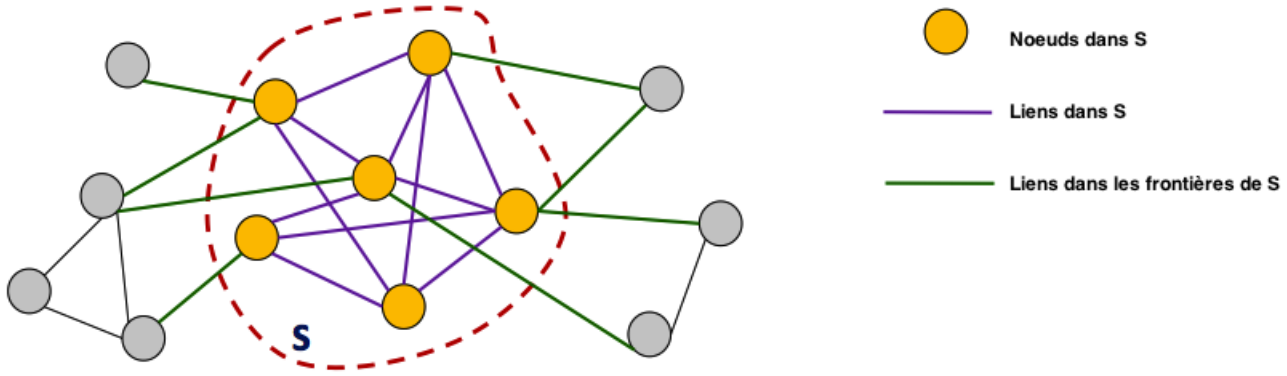


FIGURE 2.1 – Exemple d'un ensemble de nœuds S estimé comme une communauté : $n_s = 6, m_s = 11$ et $c_s = 8$

2.2 Évaluation orientée communautés

La méthode d'évaluation orientée communautés consiste à analyser chaque communauté par une fonction de score puis à calculer la moyenne des scores des communautés pour déterminer la qualité de la partition globale. Différentes fonctions de qualité sont utilisées pour évaluer la structure de la connectivité d'un ensemble de nœuds. Toutes les fonctions de qualité se fondent sur l'hypothèse que les communautés sont des ensembles de nœuds avec de nombreux liens entre eux et peu de liens avec le reste du réseau. Certaines fonctions se focalisent sur la connectivité interne, d'autres se concentrent sur la connectivité externe et d'autres fonctions s'appuient sur la connectivité interne et externe en même temps.

Soit $G(V, E)$ un graphe non-dirigé avec n et m liens. Soit S l'ensemble des nœuds à évaluer comme une communauté, avec n_S le nombre de nœuds dans S, $n_S = |S|$, et m_S le nombre de liens dans S : $m_S = e(S)$ et c_S le nombre de liens dans les frontières de S : $c_S = |\{(u, v) \in E : u \in S, v \notin S\}|$ (voir figure 2.1).

Soit $f(S)$ la fonction de qualité qui mesure la qualité d'une communauté S dans le graphe. Nous exposons dans ce qui suit, l'ensemble des fonctions de qualité selon le type de connectivité traitée.

2.2.1 Évaluation basée sur la connectivité interne

Les fonctions de qualité qui évaluent la connectivité interne d'une communauté S du graphe G sont :

— **Densité interne :** la densité interne représente la concentration de la connectivité interne de l'ensemble de nœuds S [Radicchi *et al.*, 2004] :

$$f(S) = \frac{m_S}{n_S(n_S - 1)/2} \tag{2.1}$$

— **Nombre de liens :** le nombre de liens entre les nœuds de S est représenté comme suit : $f(S) = m_S$ [Radicchi *et al.*, 2004].

— **Degré interne moyen :** le degré moyen interne des nœuds de S [Radicchi *et al.*, 2004] est défini comme suit :

$$f(S) = \frac{2m_S}{n_S} \tag{2.2}$$

— **Fraction au dessus du degré moyen (FDDM) :** la FDDM représente la fraction des nœuds de S qui ont un degré interne plus élevé que d_m, avec d_m la valeur moyenne de $d(v)$ dans V [Yang et Leskovec, 2012]. Formellement, cette fraction est définie comme suit :

$$f(S) = \frac{|\{u : u \in S, |\{(u,v) : v \in S\}\}| > d_m|}{n_S} \tag{2.3}$$

— **Taux de participation à un triangle (TPT) :** le TPT est la fraction des nœuds dans S qui appartiennent à un triangle [Yang et Leskovec, 2012]. Il est formellement défini comme suit :

$$f(S) = \frac{|\{u : u \in S, \{(v,w) : v,w \in S, (u,v) \in E, (u,w) \in E, (v,w) \in E\} \neq \emptyset\}|}{n_S} \tag{2.4}$$

Le tableau 2.1 montre les valeurs de ces fonctions sur l'exemple présenté dans la figure 2.1.

	$f(S)$
Densité interne	0.66
m_s	11
Degré interne moyen	3.33
FDDM	0.33
TPT	1

TABLE 2.1 – Connectivité interne de S pour l'exemple de la figure 2.1

2.2.2 Évaluation basée sur la connectivité externe

Les fonctions de qualité qui permettent d'évaluer la connectivité externe d'une communauté S du graphe G sont les suivantes :

— **Expansion :** elle mesure le nombre de liens par nœud pointant vers l'extérieur de S [Radicchi *et al.*, 2004] :

$$f(S) = \frac{c_S}{n_S} \tag{2.5}$$

— **Rapport de coupe :** la fonction rapport de coupe représente la fraction des liens existants externes (relativement aux liens possibles) [Fortunato, 2010] :

$$f(S) = \frac{c_S}{n_S(n - n_S)} \tag{2.6}$$

La figure 2.2 présente le résultat de ces deux fonctions sur l'exemple de la figure 2.1.

	$f(S)$
Expansion	1.33
Rapport de coupe	0.19

TABLE 2.2 – Connectivité externe de S dans l'exemple affiché dans la figure 2.1

2.2.3 Évaluation basée sur la connectivité interne et externe

Il existe d'autres fonctions qui permettent de combiner la connectivité interne et externe afin d'évaluer une communauté S. Ces fonctions sont représentées comme suit :

— **Conductance :** elle mesure la fraction de tous les liens qui sont externes de S [Shi et Malik, 2000] :

$$f(S) = \frac{c_S}{2m_S + c_S} \tag{2.7}$$

— **Fraction maximale de degré sortant (Max-FDS) :** la Max-FDS permet de mesurer la fraction maximale des liens d'un nœud de S qui pointent à l'extérieur de S [Flake *et al.*, 2000] :

$$f(S) = \max_{u \in S} \frac{|\{(u,v) \in E : v \notin S\}|}{d(u)} \tag{2.8}$$

— **Fraction moyenne de degré sortant (Moy-FDS) :** la Moy-FDS mesure la moyenne de la fraction des liens d'un nœud de S qui pointent à l'extérieur de S [Flake *et al.*, 2000] :

$$f(S) = \frac{1}{n_S} \sum_{u \in S} \frac{|\{(u,v) \in E : v \notin S\}|}{d(u)} \tag{2.9}$$

— **Flake-FDS** : elle calcule la fraction des nœuds dans S qui ont moins de liens pointant vers l'intérieur de S que vers l'extérieur[Flake *et al.*, 2000] :

$$f(S) = \frac{|\{u : u \in S, |\{(u,v) \in E : v \in S\}| < d(u)/2\}|}{n_S} \tag{2.10}$$

Le tableau 2.3 montre le résultat de ces fonctions de qualité sur l'ensemble de nœuds S présenté dans la figure 2.1.

	$f(S)$
Conductance	0.28
Max-FDS	0.4
Moy-FDS	0.26
Flake-FDS	0

TABLE 2.3 – Connectivité interne et externe de S dans l'exemple affiché dans la figure 2.1

2.3 Évaluation orientée partition

L'évaluation orientée partition consiste à donner un score à toute une partition. De nombreuses fonctions de qualité ont été proposées pour mesurer la qualité d'une partition [Mancoridis *et al.*, 1998]. La fonction de modularité [Girvan et Newman, 2002, Newman, 2006b], que nous avons présentée dans la section 1.4.1 du chapitre 1, reste la fonction la plus acceptée. La modularité compare la partition d'un réseau par rapport à un modèle aléatoire. Plus la modularité est élevée, plus la partition est bonne. Néanmoins, les études réalisées dans [Aynaud, 2011] soulève le problème de significativité de ces résultats (cf. chap. 1 section 1.4.3). Dans [Guimera *et al.*, 2004, De Montgolfier *et al.*, 2011], les auteurs montrent également que la modularité est élevée dans des graphes n'ayant aucune structure communautaire. Ces inconvénients remettent en cause l'utilisation de la modularité pour l'évaluation des partitions.

Une étude récente [Yang et Leskovec, 2012] a montré que la majorité des fonctions de qualité que nous avons présentée jusqu'à maintenant sont corrélées. En appliquant un seuillage de 0.6 sur les corrélations, les auteurs ont aperçu qu'il existe 4 groupes (voir Figure 2.2). Cela signifie que les fonctions de qualité du même groupe retournent des valeurs fortement corrélées et quantifient le même aspect de la structure de connectivité.

Par contre, la modularité est faiblement corrélée avec les autres fonctions de score, notamment celles qui évaluent la connectivité interne (la corrélation maximale est trouvée avec le degré interne moyen : 0.05). Ceci est expliqué par le fait que la modularité n'évalue pas la connectivité interne d'un ensemble de nœuds en tant que tel mais elle l'évalue en le comparant à un modèle aléatoire.

2.4 Évaluation par rapport à la vérité de terrain

Le principe de l'évaluation par rapport à la vérité de terrain est de calculer la similarité entre la partition trouvée et la classification issue de la réalité de terrain, connu par "la vérité de terrain". Dans les réseaux réels, cette classification est généralement faite par un expert du domaine, comme elle peut être calculée en fonction de certaines informations sémantiques décrivant les nœuds et/ou les liens du réseau. La disponibilité de la vérité de terrain permet d'utiliser les différentes mesures de distance entre clusters développées pour l'évaluation des approches de classification non-supervisé [Aggarwal et Reddy, 2014]. Plus la partition ressemble à la vérité de terrain plus cette partition est meilleure.

Dans ce qui suit, nous décrivons les principaux critères utilisant ce type d'évaluation. Puis, nous présentons quelques benchmarks réels disposent d'une vérité de terrain. Ensuite, nous exposons des travaux qui portent sur la construction de benchmarks à base d'hypothèses. Enfin, nous décrivons les deux principaux générateurs de graphe.

2.4.1 Mesures d'évaluation

De nombreux critères d'évaluation sous forme de mesure de similarité entre deux partitions du même ensemble d'objets, ont été proposés [Pfitzner *et al.*, 2009] dont une cinquantaine a été étudiée et comparée. Le choix de la "bonne" mesure demeure difficile. En

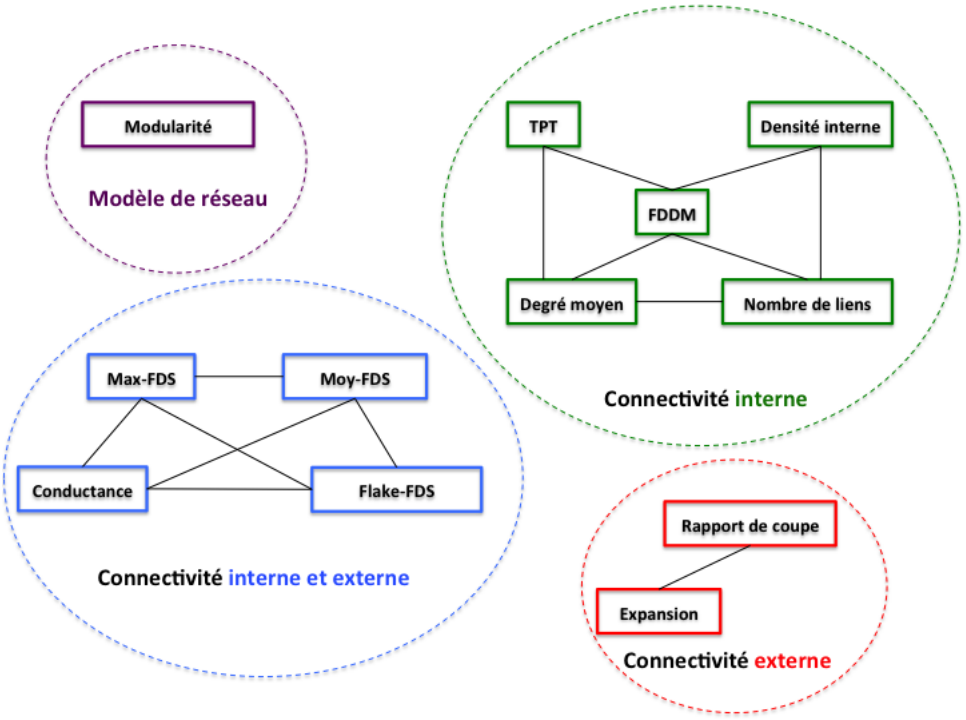


FIGURE 2.2 – Fonctions de qualité formant 4 groupes selon leur corrélation

revanche, on peut classifier ces critères en deux catégories : (1) les mesures basées sur le compte de paires, et (2) les mesures basées sur la théorie de l'information. Nous détaillons dans ce qui suit ces deux catégories et nous décrivons les mesures utilisées par chacune d'elles.

Mesures basées sur la correspondance entre les ensembles

Cette famille de mesures a été définie dans [Meilă, 2005]. Elle s'appuie sur les mesures classiques, la précision et le rappel, de la recherche d'information. La mesure la plus populaire est la pureté [Manning *et al.*, 2008]. Considérons deux partitions $U = \{U_1, U_2, ..., U_R\}$, $V = \{V_1, V_2, ..., V_C\}$ deux partitions de l'ensemble $X = \{x_1, x_2, ..., x_N\}$ où N est le nombre d'objets, la pureté d'une partie U_i par rapport à la partition V est :

$$Pur(U_i, V) = \max_j \frac{|U_i \cap V_j|}{|U_i|} \tag{2.11}$$

Cela permet de chercher parmi toutes les parties de V, la partie majoritaire dans U_i et la proportion des éléments de U_i qu'elle représente. Plus l'intersection est importante, plus leur correspondance est forte et plus la pureté est élevée. La pureté totale de la partition U par rapport à la partition V est obtenue en calculant la somme des puretés de chaque U_i pondérée par leur proportion dans l'ensemble traité :

$$Pur(U, V) = \sum_i \frac{U_i}{N} Pur(U_i, V) \tag{2.12}$$

La pureté, comme beaucoup d'autres mesures d'évaluation issues de la fouille de données, considère la structure communautaire comme une partition de l'ensemble des nœuds et ignore la topologie des réseaux. Dans [Labatut, 2012], l'auteur montre que suite à la topologie, les erreurs de classification n'ont pas forcément la même importance. Il a donc proposé une nouvelle version de pureté qui tient compte de l'information topologique du réseau. La pureté d'un nœud x pour une partition U par rapport à une partition V est :

$$Pur(x, U, V) = \delta(\arg\max_j |U_\alpha \cap V_j|, \beta) \tag{2.13}$$

Avec $x \in U_\alpha$ et $x \in V_\beta$; et où δ est le symbole de Kronecker (i.e. $\delta(a, b) = 1$ si $a = b$, et 0 sinon). La fonction est binaire : 1 si V_β est la partie de V majoritaire dans U_α et 0 sinon.

Après une reformulation de la fonction de pureté présentée dans l'équation 2.12, la nouvelle fonction de pureté est la suivante :

$$Pur\prime(U, V) = \sum_i \sum_{x \in U_i} \frac{w_x}{\sum_y w_y} Pur(x, U, V) \tag{2.14}$$

w est le poids représentant l'importance topologique d'un nœud, il égale au rapport entre le degré interne du nœud $d_{int}(u)$ et le degré maximal du réseau $max_v d(v)$: $w_u = \frac{d_{int}(u)}{max_v d(v)}$. Rappelons que le degré interne correspond au nombre de ses voisins directs situés dans la même communauté.

Cette nouvelle fonction de pureté pénalise plus fortement les erreurs pour les nœuds les plus importants topologiquement.

Mesures basées sur le compte de paires

Soient $X = \{x_1, x_2, ..., x_N\}$ un ensemble de N objets et $U = \{U_1, U_2, ..., U_R\}$, $V = \{V_1, V_2, ..., V_C\}$ deux partitions de X. Les objets communs entre les deux partitions U et V sont décrits sous la forme d'un tableau de contingences $M = [n_{ij}]_{R \times C}$ où n_{ij} désigne le nombre d'objets communs à U_i et V_j (voir tableau 2.4).

$U \setminus V$	V_1	V_2	$\cdots$	V_C	somme
U_1	n_{11}	n_{12}	$\cdots$	n_{1C}	a_1
U_2	n_{21}	n_{22}	$\cdots$	n_{2C}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
U_R	n_{R1}	n_{R2}	$\cdots$	n_{RC}	a_R
somme	b_1	b_2	$\cdots$	b_C	$\sum_{ij} n_{ij} = N$

TABLE 2.4 – Tableau de contingence entre deux partitions U et V d'un même ensemble de N objets. Les sommes a_i et b_j sont égales au nombre d'éléments dans les parties correspondantes U_i et V_j.

Plusieurs indices de similarité des partitions sont calculées en fonction de ce tableau de contingences. Le but est de déterminer si les paires d'objets sont classées de la même manière dans les deux partitions. Plus formellement, soit $S = \{(x_i, x_j) \in X \times X | i \neq j\}$ l'ensemble de N^2 paires possibles d'objets de X. On mesure :

— N_{11} (accord positif), le nombre de paires d'objets qui sont dans la même classe dans U et V

— N_{10} (désaccord), le nombre de paires d'objets qui sont dans la même classe dans U mais dans différentes classes dans V

— N_{01} (désaccord), le nombre de paires d'objets qui sont dans différentes classes dans U mais dans la même classe dans V

— N_{00} (accord négatif), le nombre de paires d'objets qui sont dans des classes différentes dans U et V.

Intuitivement, N_{11} et N_{00} sont des indicateurs d'accord entre U et V, tandis que N_{01} et N_{10} sont des indicateurs de désaccord. En se basant sur cette théorie, [Rand, 1971] a proposé un indice, appelé indice de Rand (*Rand Index* en anglais), qui est défini comme suit :

$$RI(U,V) = \frac{N_{00} + N_{11}}{N^2} \tag{2.15}$$

Le RI prend des valeurs entre 0 et 1. $RI = 1$ quand les deux partitions sont identiques et $R = 0$ si aucune paire de points n'apparaissent soit dans le même cluster ou en différents clusters dans les deux partitions U et V,i.e. $N_{00} = N_{11} = 0$. Ce cas aura lieu uniquement quand un regroupement contient un seul cluster tandis que l'autre ne se compose que de clusters ne contenant qu'un seul nœud. La limite de cet indice est qu'il ne donne pas une valeur constante entre deux partitions aléatoires. Pour obtenir une valeur constante, Huber et Arabie [Hubert et Arabie, 1985] ont proposé une version ajustée de l'indice de Rand (ou *Adjusted RAnd Index (ARI)* en anglais) :

$$ARI(U,V) = \frac{2(N_{00}N_{11} - N_{01}N_{10})}{(N_{00} + N_{01})(N_{01} + N_{11}) + (N_{00} + N_{10})(N_{10} + N_{11})} \tag{2.16}$$

L'indice de Rand ajusté demeure la mesure la plus acceptée par rapport aux autres mesures basées sur le compte des paires [Steinley, 2004].

Mesures basées sur la théorie de l'information

D'après la théorie des probabilités et la théorie de l'information, l'information mutuelle de deux variables aléatoires est une quantité mesurant la dépendance statistique de ces variables. Deux variables sont dites indépendantes si la réalisation de l'une n'apporte aucune information sur la réalisation de l'autre. L'information mutuelle dans ce cas est nulle. Elle augmente si la dépendance augmente.

Soit X une variable aléatoire qui peut prendre n_x valeurs parmi $\{x_1, x_2, \cdots, x_{n_x}\}$ avec les probabilités $\{P(x_1), P(x_2), \cdots, P(x_{n_x})\}$. L'entropie de Shannon de la variable X est définie par :

$$H(X) = \sum_{i=1}^{n_x} P(x_i) log(\frac{1}{P(x_i)}) \tag{2.17}$$

et l'entropie conjointe de X et Y :

$$H(X,Y) = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} P(x_i, x_j) log(\frac{1}{P(x_i, x_j)}) \tag{2.18}$$

L'information mutuelle entre deux variables X et Y est alors définie par :

$$I(X,Y) = H(X) + H(Y) - H(X,Y) \tag{2.19}$$

La valeur de l'information mutuelle est plus importante quand les variables sont liées. La valeur 0 indique que les variables sont indépendantes.

Intuitivement, l'information mutuelle peut être vue comme la quantité d'information qu'apporte la variable U pour prédire la valeur de V. Cette propriété suggère que l'information mutuelle peut être utilisée pour mesurer les informations partagées par deux partitions et évaluer leur similarité [Banerjee *et al.*, 2005].

Supposons que nous choisissons un objet au hasard, la probabilité que cet objet soit dans le cluster U_i est $P(i) = \frac{|U_i|}{N}$. L'entropie associé à U est :

$$H(U) = -\sum_{i=1}^{R} P(i)log(P(i)) = -\sum_{i=1}^{R} \frac{a_i}{N} log(\frac{a_i}{N}), \tag{2.20}$$

L'entropie conjointe de U et V se définit par :

$$H(U,V) = -\sum_{i=1}^{R}\sum_{j=1}^{C} P(i,j)log(P(i,j)) = -\sum_{i=1}^{R}\sum_{j=1}^{C} \frac{n_{ij}}{N} log(\frac{n_{ij}}{N}). \tag{2.21}$$

Alors l'information mutuelle entre deux partitions U et V est :

$$I(U,V) = \sum_{i=1}^{R}\sum_{j=1}^{C} P(i,j)log(\frac{P(i,j)}{P(i)P'(j)}) = \sum_{i=1}^{R}\sum_{j=1}^{C} \frac{n_{ij}}{N} log(\frac{n_{ij}N}{a_i b_j}) \tag{2.22}$$

où $P'(j) = \frac{|V_j|}{N}$ est l'entropie du cluster V et $P(i,j)$ représente la probabilité qu'un nœud appartienne au cluster U_i de U et au cluster V_j de V : $P(i,j) = \frac{|U_i \cap V_j|}{N}$.

Dans [Meilă, 2005], une mesure similaire est introduite, appelée variation de l'information (IV) et est définie comme suit :

$$VI(U,V) = H(U) + H(V) - 2I(U,V) \tag{2.23}$$

Une version normalisée de l'information mutuelle est introduite dans [Strehl et Ghosh, 2003] afin d'obtenir une mesure entre 0 et 1. L'information mutuelle normalisée (NMI) est définie comme suit :

$$NMI(U,V) = \frac{I(U,V)}{\sqrt{H(U)H(V)}} \tag{2.24}$$

Le NMI est égale à 0 pour deux partitions indépendantes et est égale à 1 quand elles sont égales.

Une version ajustée de la mesure NMI est récemment introduite dans [Pfitzner *et al.*, 2009]. Or, l'ensemble des mesures de comparaison de paritions basées sur la théorie de l'information sont assez corrélées entre elles. En pratique, la mesure NMI reste la mesure la plus utilisée dans la littérature scientifique.

2.4.2 Benchmarks d'évaluation

La disponibilité d'une partition de référence pour un graphe G peut être le résultat d'un des trois processus suivant :

— Annotation par un expert
— Définition implicite à base d'hypothèses
— Génération par un modèle artificiel

2.4.2.1 Annotation par un expert

Les graphes pour lesquels des experts ont défini des partitions de référence ne sont pas nombreux et souvent de très petite taille. Nous décrivons ci-dessous les quatre principaux réseaux.

Club de karaté du Zachary

C'est un réseau social qui décrit les relations entre 34 membres d'un club de karaté qui ont été observés sur une période de deux ans [Zachary, 1977]. À cause d'un désaccord entre l'administrateur du club et l'instructeur, celui-ci a fondé un nouveau club et la moitié des membres du club d'origine est partie avec lui. Le réseau se compose donc en deux groupes (Figure 2.3(a)).

Football américain

C'est un réseau qui décrit les rencontres entre 115 équipes de football américain. La compétition est composée de 12 "conférences" régionales correspondant aux différentes ligues dans lesquelles les équipes se rencontrent plus fréquemment [Girvan et Newman, 2002] (Figure 2.3(b)).

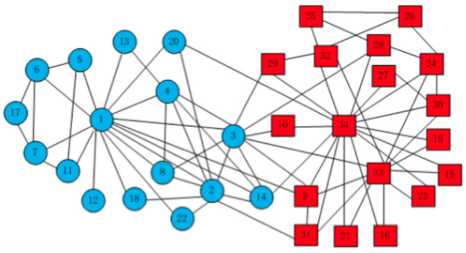
Livres sur la politique américaine

C'est un réseau qui contient des livres sur la politique américaine vendus par la librairie en ligne Amazone.com. Les liens entre les livres représentent l'achat des livres par des acheteurs ayant utilisés l'option du site Amazon indiquant "les clients ayant acheté ce livre ont également acheté ces autres livres". Les livres ont aussi été classés manuellement en trois groupes : "libérals", "neutres", ou "conservateurs". Cette classification est faite par Mark Newman d'après les descriptions et les commentaires sur les livres sur Amazon[1] (Figure 2.3(c)).

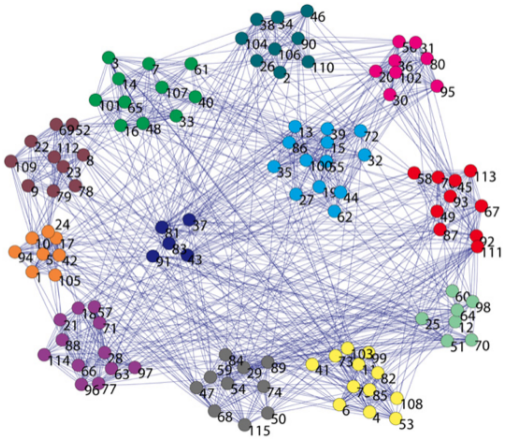
Dauphins

C'est un réseau social qui décrit des associations fréquentes entre 62 dauphins dans une communauté vivant en Nouvelle-Zélande [Lusseau *et al.*, 2003] (Figure 2.3(d)).

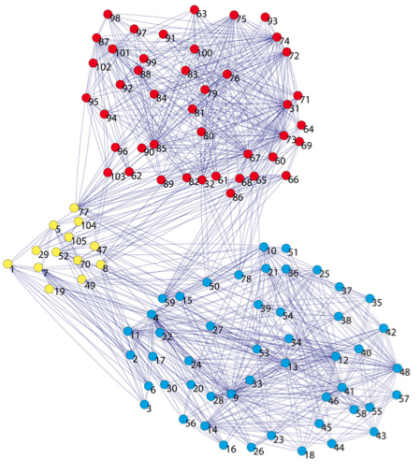
1. La classification est disponible en ligne : `http://www-personnal.umich.edu/~mejn/netdata`



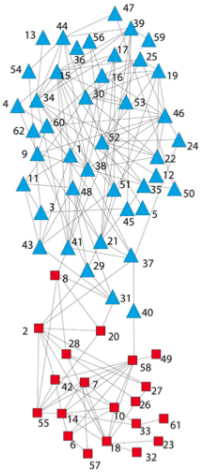
(a) Réseau du club de karaté de Zachary : 34 membres connectés par 78 liens et classés en 2 groupes



(b) Réseau de football américain : 115 équipes connectées par 616 liens et classées en 11 groupes



(c) Réseau des livres sur la politique américaine : 100 livres connectés par 441 liens et classés en 3 groupes



(d) Réseau de 62 dauphins connectés par 78 liens et classés en 2 groupes

FIGURE 2.3 – Structure des réseaux réels étudiés

2.4.2.2 Définition implicite à base d'hypothèses

Ces approches basées sur l'inférence de communautés en fonction de certaines informations sémantiques décrivant les nœuds etou les liens d'un réseau. C'est l'approche adoptée dans le travail récent de Jure Leskovec et al. [Yang et Leskovec, 2012] :

Méthode de Yang et Leskovec

Cette méthode [Yang et Leskovec, 2012] introduit une nouvelle méthode pour extraire la vérité de terrain d'un réseau complexe non-orienté statique. En se basant sur l'hypothèse que l'identification de communautés permet de trouver des groupes de nœuds qui ont la même fonction dans le réseau, les auteurs proposent d'utiliser ces fonctions communes pour définir la vérité de terrain des communautés. Une fonction commune peut être un rôle, une affiliation ou un attribut. Ayant cette information pour chaque nœud du réseau, permet de déterminer directement une structure communautaire chevauchante qui sera considérée comme la vérité de terrain.

Cette méthode a été appliquée sur des réseaux appartenant à trois domaines différents. La fonction qui sera utilisée comme label pour chaque nœud dépend du type de réseau étudié. Nous décrivons ci-dessous pour chacun de ces réseaux, la fonction qui a été retenue comme label :

— Réseaux sociaux en ligne : la fonction qui est prise en compte dans ce type de réseaux, est l'appartenance à des groupes. Ces groupes sont généralement créés avec des sujets, centres d'intérêts, des loisirs et régions géographiques bien déterminées. Les utilisateurs peuvent rejoindre les groupes qui les intéressent et partager des contenus. Cette appartenance identifie la communauté dans la vérité de terrain.

— Réseaux d'informations : le réseau étudié est celui de co-achat des produits sur Amazon [2]. Les nœuds représentent les produits et les liens connectent les produits co-achetés. Puisque chaque produit appartient à une ou plusieurs catégories organisées hiérarchiquement, les auteurs proposent d'utiliser ces catégories comme la vérité de terrain.

— Réseaux de collaborations : Les nœuds sont les auteurs et les liens connectent les auteurs qui ont co-publié ensemble. Afin d'identifier la communauté scientifique des auteurs, Yang et Leskovec adoptent l'hypothèse que les auteurs qui ont publié dans le même lieu (e.g. conférences) appartiennent à la même communauté scientifique. Ainsi, les lieux de publications représentent la vérité de terrain des communautés.

Cette méthode va sans doute enrichir l'évaluation des algorithmes de détection de communautés. La mise en place de vérité de terrain pour 230 réseaux qui contiennent des millions de nœuds, constitue un nouveau challenge pour les différentes méthodes de détection de communautés. Cependant, nous constatons que les règles suivies pour déterminer le label des nœuds sont insuffisantes. Dans les réseaux sociaux par exemple, le fait de rejoindre un groupe n'est pas une information assez pertinente pour qu'elle soit utilisée comme label de communauté. Il se peut que l'appartenance de l'utilisateur à un tel groupe soit liée à

2. `http://www.amazon.com`

une recherche d'informations momentanée. Il existe une marge d'erreur pour ce type de vérité de terrain.

En outre, le travail de [Lee et Cunningham, 2013] a démontré que l'évaluation des algorithmes de détection de communautés sur ce type de benchmark, est confronté à deux principaux problèmes :

— Les méta-données utilisées pour décrire l'appartenance communautaire des nœuds peuvent êtres incomplètes : l'attribut utilisé comme indicateur de la vérité de terrain peut trouver, par manque d'informations sur certains nœuds, un sous-ensemble de la partition réelle du réseau. La tâche d'évaluation des communautés trouvées devient ainsi très difficile à réaliser.

— Dans certains cas, les vraies communautés du réseau ne sont pas identifiables par un seul attribut : les auteurs ont montré par des exemples réels que les communautés réelles ne peuvent être trouvées que par l'emploi de plusieurs attributs.

Pour faire face à ces deux problèmes, [Lee et Cunningham, 2013] ont proposé une méthode d'évaluation appropriée pour ce genre de réseau. Cette méthode est basée sur un algorithme d'apprentissage supervisé. L'idée est qu'au lieu de faire une correspondance une-à-une entre les classes des attributs et les communautés, on utilise un classifieur, disposant d'une méthode de sélection, de variables et d'un espace d'hypothèses assez expressif pour tenir compte du fait que les communautés peuvent être un sous-ensemble ou un sur-ensemble des classes définies par l'attribut. Le classifieur permet d'apprendre de manière flexible une correspondance plus complexe entre les deux.

À son tour, ce cadre d'évaluation a quelques limites. En effet, il ne garantit pas de découvrir les relations complexes entre les attributs et les communautés, et pour ne pas échouer dans l'évaluation, il faut des bonnes connaissances à priori sur le fait que la structure communautaire est reliée à l'attribut en question. De plus, puisque cette méthode est complexe, dans le cas où la précision est mauvaise, il n'est pas facile de savoir si cela dû à l'algorithme de détection de communauté ou à l'algorithme d'apprentissage utilisé.

2.4.2.3 Génération par un modèle artificiel

Le principe des générateurs artificiels est de produire des graphes avec des structures communautaires paramétrables. Deux principaux modèles ont été proposés : le modèle de Girvan et Newman et le modèle LFR.

Modèle de Girvan et Newman

Ce modèle a été proposé par [Girvan et Newman, 2002]. Il génère un graphe de 128 nœuds, divisé en 4 groupes de 32 nœuds chacun. Le degré moyen du réseau est 16 et les nœuds ont des degrés proches, comme dans un graphe aléatoire. À la différence d'un graphe aléatoire, les nœuds ont tendance à être reliés de manière préférentielle à des nœuds de leur groupe : un paramètre k_{out} indique le nombre de liens qui connecte chaque nœud à un autre nœud d'un groupe différent. Quand $k_{out} < 8$, chaque nœud partage plus de liens avec son groupe qu'avec le reste du réseau. Dans ce cas, les quatre groupes sont des communautés

bien définies et un bon algorithme devrait être en mesure de les identifier. La figure 2.4 montre la structure du graphe de ce modèle.

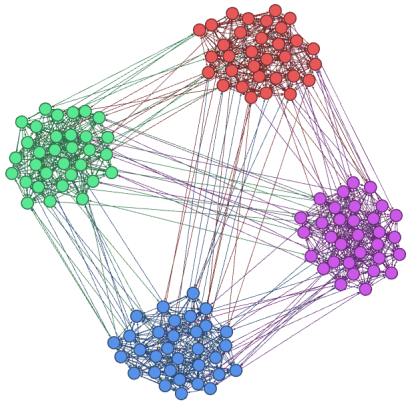


FIGURE 2.4 – Graphe généré par le modèle Girvan et Newman

Ce benchmark a été utilisé fréquemment pour tester des algorithmes. Cependant, le fait que les nœuds ont le même degré, que les communautés ont la même taille et que le réseau est de petite taille, indiquent que le benchmark de Girvan et Newman ne peut pas être un benchmark de référence pour évaluer les algorithmes car les réseaux réels sont caractérisés par une distribution hétérogène de degré ainsi que de la taille des communautés. Il est également inapproprié de tester des algorithmes, conçu pour traiter des réseaux de millions de nœuds, sur ces petits réseaux.

Modèle LFR

Le modèle Lancichinetti-Fortunato-Radicchi (LFR) est un générateur de graphe introduit par [Lancichinetti *et al.*, 2008]. Il garantit d'obtenir des valeurs considérées comme réalistes pour les propriétés suivantes : taille du réseau, distribution en loi de puissance des degrés et taille de communautés. Ceci est fait via le contrôle des paramètres présentés dans le tableau 2.5. La configuration des trois premiers paramètres est indispensable tandis que le reste peut être configuré automatiquement par le modèle. Le coefficient de mélange représente la proportion moyenne souhaitée de liens entre un nœud et les nœuds situés en dehors de sa communauté (liens inter-communautaires).

Le processus de génération est comme suit :

1. Premièrement, LFR utilise le modèle de configuration proposé par [Molloy et Reed, 1995] pour générer un réseau avec un degré moyen k, un degré maximale k_{max} et une distribution de degré d'exposant γ.
2. Deuxièmement, des communautés virtuelles sont définies de sorte que leurs tailles suivent une distribution en loi de puissance avec un exposant β. Chaque nœud est

Paramètre	Signification
N	Nombre de nœuds
k	Degré moyen
μ	Coefficient de mélange
k_{max}	Degré maximal
γ	Paramètre de la distribution de degré
β	Paramètre de la distribution de la taille des communautés

TABLE 2.5 – Principaux paramètres du générateur LFR

aléatoirement affecté à une communauté à condition que la taille de la communauté soit supérieure ou égale au degré interne du nœud.

3. Troisièmement, un processus itératif prend place pour reconnecter certains liens afin d'approximer μ tout en conservant la distribution des degrés. Pour chaque nœud, le degré total n'est pas modifié, mais le ratio de liens internes et externes est modifié de sorte que la proportion résultante se rapproche de μ.

Un exemple de graphe généré par ce modèle est présenté dans la figure 2.5.

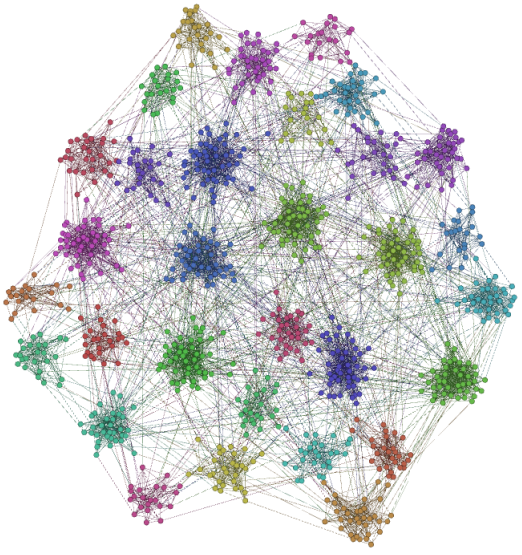


FIGURE 2.5 – Graphe généré par le modèle LFR :$N = 1000, k = 15, maxk = 50, \mu = 0.1$

Les principales propriétés des réseaux réels ont été étudiés empiriquement sur ce modèle [Orman et Labatut, 2009, Orman *et al.*, 2012]. Il génère des réseaux de faible densité et ayant une distribution de degré proche de la loi de puissance. Cependant, malgré la flexibilité du modèle de configuration [Molloy et Reed, 1995] utilisé dans la première étape de LFR, il est connu pour générer des réseaux avec une corrélation de degré nulle [Serrano et Boguñá, 2005] et un coefficient de clustering faible quand les degrés sont distribués selon la loi de puissance [Newman, 2003]. Rappelons que dans [Newman, 2003],

l'auteur a montré que la plupart des réseaux complexes ont une corrélation de degré non-nulle ; elle est positive dans les réseaux sociaux et négative dans les autres types de réseaux complexes.

Pour faire face à ces limites, [Orman *et al.*, 2013] proposent de remplacer le modèle de configuration de [Molloy et Reed, 1995] utilisé par LFR dans la première étape, par un autre modèle basé sur l'attachement préférentiel et l'évolution. Les auteurs ont expérimenté deux modèles : (1) le modèle d'attachement préférentiel de Barabási-Albert (BA) [Barabasi et Albert, 1999], et une de ses variantes (2) le modèle d'attachement préférentiel évolutif (EV)[Poncela *et al.*, 2008]. L'étude des réseaux générés par LFR-BA et LFR-EV a montré qu'ils ont la plupart des caractéristiques des réseaux réels connus, notamment la corrélation non-nulle des degrés. En conséquence, la détection de communautés est devenue plus au moins difficile sur LFR en utilisant BA ou EV [Orman *et al.*, 2013].

En revanche, rien ne prouve que les graphes générés par ces modèles sont bien similaires dans leurs mécanismes de formation aux graphes de terrain. L'historique de l'évolution de nos connaissances sur la caractérisation des graphes de terrain ne permet pas de dire qu'on connait déjà la liste ultime de leurs caractéristiques topologiques.

2.5 Conclusion

Dans ce chapitre, nous avons étudié les trois méthodes d'évaluation des algorithmes de détection de communautés dans les réseaux complexes. Les deux premières méthodes topologiques, évaluation orientée-communautés et évaluation orientée-partition, sont presque toutes corrélées. La fonction de qualité topologique la plus utilisée est la modularité. Cette dernière a plusieurs limites notamment le problème de significativité de ses résultats. La troisième méthode d'évaluation consiste à comparer la partition trouvée par rapport à la vérité de terrain du réseau. En pratique, cette vérité de terrain est disponible que dans des petits réseaux benchmarks, qui sont généralement utilisés pour estimer la qualité de solution des algorithmes. Le travail de Yang et Leskovec, qui utilise des métadonnées pour estimer la partition réelle, est une bonne initiative. Toutefois, il est bien difficile de définir ces métadonnées sans une analyse approfondie, mais les communautés définies sont souvent très nombreuses, de petite taille par rapport à la taille du réseau. De même, les générateurs artificiels des graphes, notamment le modèle LFR, ne garantissent pas d'un côté d'avoir toutes les caractéristiques des réseaux réels, et d'un autre côté, les communautés des réseaux générés, sont construites pour qu'elles soient bien détectées.

Dans l'ensemble, la rareté des réseaux de taille importante pour lesquelles une partition de référence est connue, les limitations des critères topologiques d'évaluation des communautés, et les limitations des modèles générateurs de réseaux artificiels de benchmark, sont quelques facteurs qui ont motivé la recherche de nouvelles approches pour l'évaluation des partitions détectées par les différents algorithmes de détection de communautés. L'évaluation guidée par une tâche semble être une alternative prometteuse.

Chapitre 3

L'approche LICOD

Sommaire

3.1 Introduction

Les approches fondées sur les leaders font partie de la famille d'approches : détection de communautés centrée graines, avec la spécificité "leader" pour les nœuds graines. Comme on a décrit dans le chapitre 1, les approches fondées sur les leaders sont structurées en trois étapes principales : (1) l'identification des leaders, (2) le calcul de communautés locales centrées leaders, et (3) le calcul de communautés globales. Le leader peut être un seul nœud comme il peut être un sous-graphe. Les méthodes existantes utilisent principalement les mesures de centralité pour la sélection de leaders et considèrent les communautés locales trouvées comme des communautés globales. Quand à la deuxième étape, les deux techniques utilisées sont l'expansion et l'agglomération. À l'opposé des méthodes adoptant l'expansion, les méthodes agglomératives sont généralement déterministes. Or, le rattachement des nœuds aux communautés dans ces approches reste singulier. De plus, la multitude des mesures applicables dans les différentes étapes des approches agglomératives nécessite une étude approfondie sur leur pertinence.

L'approche LICOD [Kanawati, 2011] constitue un framework pour évaluer les différentes alternatives. Elle introduit une nouvelle nature de leaders : un ensemble de nœuds qui ne sont pas forcément interconnectés. Leur nombre est calculé automatiquement en utilisant les mesures de centralité. LICOD adopte la technique d'agglomération pour construire les

communautés locales et en considérant que celles-ci sont les communautés finales. Contrairement aux méthodes existantes, LICOD intègre une étape de "lissage" où le choix d'appartenance à une communauté dépend de la communauté des voisins directs.

Dans ce chapitre, nous détaillons l'approche LICOD. Pour chacune de ses étapes, nous montrons l'ensemble des mesures que nous avons utilisées, y compris celles qui n'ont pas été intégrées dans la première version de LICOD. Ensuite, nous proposons une extension itérative it-LICOD de LICOD qui s'occupe plus précisément de la qualité des leaders identifiés. Dans LICOD et it-LICOD, nous expérimentons les différentes configurations sur un ensemble de réseaux réels et artificiels. Nous comparons également la performance de it-LICOD par rapport à LICOD, et aussi par rapport aux autres algorithmes de détection de communautés qui appartiennent à différentes familles d'approches.

3.2 L'approche LICOD

3.2.1 Description de l'approche

Nous présentons ci-dessous le pseudo-code de LICOD (voir algorithme 2). Étant donné un graphe connexe $G = (V, E)$:

1. LICOD commence par identifier les nœuds qui semblent être des leaders. Plusieurs heuristiques peuvent être utilisées pour estimer le rôle d'un nœud. Soit $\mathcal{L}$ l'ensemble des leaders identifiés. Cette étape est effectuée par la fonction $EstLeader()$ (voir algorithme 2, ligne 6).
2. Afin de faire face à un éventuel grand nombre de leaders, ce qui va affecter directement le nombre de communautés final, LICOD intègre une étape de regroupement des leaders estimés être dans la même communauté. Cela est réalisé via la fonction *LeaderCommunautés()* (voir algorithme 2, ligne 10). Chaque groupe de leaders représente une communauté. Notons $\mathcal{C}$ l'ensemble de communautés identifiées.
3. LICOD adopte une construction agglomérative des communautés. Chaque nœud dans le réseau, qu'il soit leader ou non, calcule son vecteur d'appartenance aux communautés dans $\mathcal{C}$. Puis, selon ce vecteur, il trie les communautés dans l'ordre décroissant. Cette étape est réalisée par la fonction $TrierEtClasser$ (voir algorithme 2, lignes 11-16).
4. Une fois chaque nœud a son vecteur d'appartenance, on commence une phase d'intégration ou le vecteur d'appartenance d'un nœud est fusionné avec ceux de ses voisins directe. Ceci permet de favoriser la classe dominante dans l'ensemble des nœuds voisins. Des algorithmes issus de la théorie de choix social sont utilisés dans cette phase de fusion de votes. Ce processus est répété jusqu'à la stabilisation du vecteur d'appartenance de chaque nœud. Le temps de convergence dépend de la méthode de fusion de votes appliquée. Cette étape est effectuée par la fonction $Fusion\_$ (voir algorithme 2, lignes 17-22).
5. À la stabilisation, chaque nœud est affecté à la communauté placée en tête de vecteur d'appartenance (voir algorithme 2, lignes 23-30).

Par la suite, nous énumérons les mesures appropriées pour chacune de ces étapes.

Algorithme 2 LICOD

Entrée : Un graphe connexe $G =< V, E >$
Sortie : Une partition $\mathcal{C}$
Début
$\mathcal{L} \leftarrow \emptyset$ {un ensemble de leaders}
pour $v \in V$ **faire**
  si $EstLeader(v)$ **alors**
    $\mathcal{L} \leftarrow \mathcal{L} \cup \{v\}$
  fin si
fin pour
$\mathcal{C} \leftarrow LeaderCommunautes(\mathcal{L})$
pour $v \in V$ **faire**
  pour $c \in \mathcal{C}$ **faire**
    $M[v, c] \leftarrow Appartenance(v, c)$
  fin pour
  $P[v] =$ **TrierEtClasser**$(M[v])$
fin pour
repeat
  pour $v \in V$ **faire**
    $P^*[v] \leftarrow$ **Fusion**$_{\mathbf{x}\in\{\mathbf{v}\}\cap\mathbf{\Gamma(v)}}\mathbf{P[x]}$
    $P[v] \leftarrow P^*[v]$
  fin pour
until Stabilisation de $P^*[v] \forall v$
pour $v \in V$ **faire**
  / assignant v aux communautés */*
  pour $c \in P[v]$ **faire**
    si $|M[v, c] - M[v, P[0]]| \leq \epsilon$ **alors**
      $COM(c) \leftarrow COM(c) \cup \{v\}$
    fin si
  fin pour
fin pour
Retourne $\mathcal{C}$
Fin

Identification de leaders

Une idée simple pour distinguer les leaders des autres nœuds est de comparer leurs centralités. Les nœuds leaders sont censés avoir des centralités plus élevés que les nœuds non-leaders. Nous proposons d'appliquer les centralités qui ont été présentées dans la section 1.5 du chapitre 1. Nous montrons leurs principales caractéristiques dans le tableau 3.1. Le type d'une centralité indique si le calcul de celle-ci est fait par rapport aux voisins directs seulement (locale), ou par rapport à k niveaux de voisins (semi-locale), ou par rapport à tout le réseau (globale).

Nom	Symbole	Formule	Complexité	Type
Degré	$C_d(v)$	$\frac{d(v)}{n-1}$	$\mathcal{O}(n)$	Locale
Degré des voisins	$C_l(v)$	$\sum_{t\in\Gamma(v)} d(t)$	$\mathcal{O}(n(k)^2)$	Semi-locale
Proximité	$C_c(v)$	$\frac{n-1}{\sum_{u\in E,u\neq V} d_g(u,v)}$	$\mathcal{O}(nlog(n)+m)$	Globale
Intermédiarité	$C_b(v)$	$\sum_{s,t\in V} \frac{\sigma(s,t\|v)}{\sigma(s,t)}$	$\mathcal{O}(n.m+(n)^2log(n))$	Globale
Vecteurs propres	C_{ev}	$\frac{1}{\lambda}\sum t \in V a_{v,t}x_t$	$\mathcal{O}(n^2)$	Globale

TABLE 3.1 – LICOD : les mesures de centralité utilisées

Un nœud est désigné comme leader si sa centralité est supérieure ou égale à $\sigma\%$ de la centralité de ses voisins, $\sigma \in [0,1]$. Plus formellement, soient $C(v)$ une centralité quelconque d'un nœud $v \in V$ et $F_v = \{u \in \Gamma(v) : c(v) > c(u)\}$. Le nœud v est leader si $\frac{|F_v|}{|\Gamma(v)|} > \sigma \in [0,1]$.

Le rôle du paramètre σ est de récupérer les leaders connectés à d'autres leaders et de contrôler leur nombre. Plus σ est élevé, moins les leaders sont présents.

Regroupement de leaders

Dans cette phase, nous nous intéressons à regrouper les leaders qui partagent le plus de voisins. Chaque groupe de leaders définira une communauté. Soient l_i et l_j deux leaders, ils seront regroupés dans la même communauté si le taux de voisins communs par rapport au nombre total de voisins est supérieur au sœuil $\delta \in [0,1]$: $\frac{|\Gamma(l_i)\cap\Gamma(l_j)|}{|\Gamma(l_i)\cup\Gamma(l_j)|} > \delta \in [0,1]$.

Le regroupement de leaders permet d'éviter un éventuel grand nombre de communautés, surtout quand on utilise des mesures de centralité locales.

Calcul de vecteur d'appartenance

Après l'identification des nœuds leaders de chaque communauté, chaque nœud $v \in V$ calcule son degré d'appartenance à chaque communauté c. Dans LICOD, nous proposons d'utiliser deux mesures basées sur la distance entre les nœuds : le plus court chemin (SP) et la mesure de Katz tronquée (TKatz).

La mesure de Katz a été introduite dans [Katz, 1953]. Elle calcule le nombre total de chemins qui relie deux nœuds, en pondérant chaque chemin de longueur l par β^l, β est un

paramètre positif qui favorise les chemins les plus courts. La mesure de Katz entre deux nœuds u et v est calculée comme suit :

$$Katz(u,v) = \sum_{l=1}^{\infty} \beta^l |paths_{u,v}^l|, \tag{3.1}$$

où $|paths_{u,v}^l|$ est l'ensemble de tous les chemins de taille l entre u et v. A étant la matrice d'adjacence du graphe, Katz peut s'écrire en fonction de A :

$$Katz(u,v) = \sum_{l=1}^{\infty} \beta^l A^l = (I - \beta A)^{-1} - I, \tag{3.2}$$

où I est la matrice d'identité de A. Cette fonction est très coûteuse. On peut choisir de s'arrêter à des chemins de longueur l_{max}, pour obtenir le score tronqué de Katz, d'où :

$$TKatz(u,v) = \sum_{l=1}^{l_{max}} \beta^l A^l \tag{3.3}$$

Dans toutes les expérimentations, la configuration des deux paramètres est l_{max} et β est : $l_{max} = 5$ et $\beta = 0.005$

Soit $Dist()$ la mesure utilisée, qui peut être SP ou TKatz, le degré d'appartenance d'un nœud v à la communauté associée au leader c est calculé comme suit :

$$Appartenance(v,c) = \frac{1}{(min_{x \in COM(c)} Dist(v,x)) + 1} \tag{3.4}$$

A la fin de cette étape, chaque nœud aura un vecteur d'appartenance aux communautés trié d'une manière décroissante, i.e. la communauté la plus proche au nœud est placée en tête (voir figure 3.1). Dans le cas où plus qu'une communauté est classée première dans le vecteur d'appartenance, le nœud est affecté à l'une d'elle aléatoirement.

Fusion de votes

L'étape de fusion de votes consiste à fusionner le vecteur d'appartenance de chaque nœud avec les vecteurs de ses voisins. L'influence des voisins permet de favoriser la communauté majoritaire et d'avoir plus d'homogénéité locale.

Différentes méthodes de fusion peuvent être appliquées. Dans LICOD, nous avons choisi d'utiliser des algorithmes issus de la théorie du choix social [Chevaleyre *et al.*, 2007]. Plus précisément, nous appliquons les trois méthodes suivantes : la méthode de majorité, la méthode de Borda [Sculley, 2007] et la méthode Kemeny [Dwork *et al.*, 2001].

Soit $L = \{L_1, L_2, ..., L_n\}$ l'ensemble des n vecteurs d'appartenance. Chacun d'eux contient les communautés triées d'une manière décroissante selon le degré d'appartenance $L_i = [c1, c_2, ..., c_k]$.

Méthode de majorité : est la méthode de fusion la plus simple. Le calcul se fonde sur la position individuelle des candidats (dans notre cas, les candidats sont les communautés) dans tous les votes. Pour chaque rang $r \in [1, 2, ..., k]$, on retient le candidat qui a été le plus classé dans ce rang.

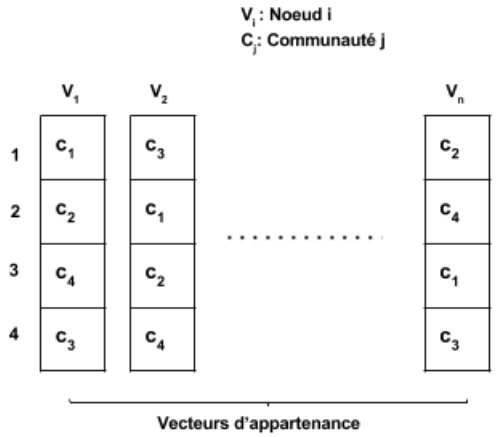


FIGURE 3.1 – LICOD : vecteurs d'appartenance des nœuds avec un exemple de 4 communautés

Méthode de Borda : la méthode de Borda est une méthode fondée sur le positionnement absolu des communautés classées plutôt que de leur classement respectif. Un score de Borda est calculé pour chaque communauté c dans les vecteurs d'appartenance et selon ce score, les communautés seront classées dans le vecteur agrégé. Pour l'ensemble des vecteurs d'appartenance L, le score d'une communauté c pour un vecteur L_i est donné par :

$$B_{L_i}(c) = \{count(c')|L_i(c) > L_i(c')\&c' \in L_i\} \tag{3.5}$$

Le score de Borda total de c est donné par :

$$B(c) = \sum_{t=1}^{n} B_{L_i}(c) \tag{3.6}$$

Méthode de Kemeny : est une méthode fondée sur l'ordre entre le rang des communautés. Elle garantit que le résultat de la fusion satisfait le principe de Condorcet [Young et Levenglick, 1978]. Ce principe stipule que : s'il y a une partition C, D de la liste totale des candidats (i.e. les communautés) C telle que $\forall x \in C$, $\forall y \in D$, si la majorité des électeurs (i.e les votes des nœuds voisins) préfère x à y, alors x est classé au-dessus de y.

Pour mieux comprendre la différence entre ces trois méthodes de fusion de votes, nous prenons cet exemple : soient cinq nœuds $\{v_0, v_1, v_2, v_3, v_4\}$ d'un graphe G, $P[v_i]$ le vecteur d'appartenance du nœud i, et $\mathcal{P}$ une partition, $\mathcal{P} = \{a, b, c, d\}$. Sachant que $\Gamma(v_0) = \{v_1, v_2, v_3, v_4\}$, on veut fusionner le vecteur d'appartenance de v_0 avec ceux de ses voisins.

- $P[v_0] = a > b > c > d$
- $P[v_1] = a > b > c > d$
- $P[v_2] = b > c > a > d$

- $P[v_3] = d > a > b > c$
- $P[v_4] = b > d > a > c$

Le résultat de fusion est :

- Majorité $\rightarrow P[v_0] = a > b > c > d$
- Borda $\rightarrow P[v_0] = b > a > d > c$
- Kemeny $\rightarrow P[v_0] = a > b > c > d$

Affectation des nœuds aux communautés A la fin de LICOD, chaque nœud v est associé à la communauté triée en tête du vecteur d'appartenance agrégé P_v^*. Le seuil ϵ contrôle le degré de chevauchement souhaité dans les communautés identifiées. Rappelons que dans les communautés chevauchantes, les nœuds peuvent appartenir à plusieurs communautés, alors que dans les communautés disjointes, chaque nœud est affecté à une seule communauté. Comme on s'intéresse dans cette thèse seulement aux communautés disjointes, on a fixé ϵ à 0.

3.2.2 Expérimentation

L'approche LICOD a été implémentée en Python en utilisant igraph [1] et Numpy [2].

LICOD a 5 paramètres : σ, δ, la mesure de centralité, la mesure utilisée pour le calcul d'appartenance $Dist()$ et la méthode de fusion des votes. Nous avons expérimenté les différentes stratégies dans LICOD sur des réseaux réels et artificiels. Nous avons remarqué durant l'expérimentation que le paramètre δ a un effet négligeable sur les résultats de LICOD. Nous nous sommes donc concentrés sur la variation des autres paramètres comme suit :

1. $\sigma = [0.5, 0.6, 0.7, 0.8, 0.9, 1]$
2. Mesure de centralité=[Centralité de degré (D), centralité de degré des voisins (SD), centralité d'intermédiarité (B), centralité de proximité (C), centralité de vecteurs propres (EV)]
3. Mesure de calcul d'appartenance=[Plus court chemin (SP), Katz tronquée (T. Katz)]
4. Méthode de fusion de votes=[Majorité, Borda, Kemeny]

La variation de chaque paramètre nécessite une configuration par défaut des paramètres restants. Pour ce fait, nous nous appuyions sur les hypothèses suivantes pour déterminer cette configuration :

— σ **:** Fixer σ à 1 pour sélectionner les leaders, empêche de détecter les leaders interconnectés, et faire baisser σ permet de générer beaucoup de communautés. Ainsi, nous choisissons la valeur 0.9 comme valeur par défaut.

— **Mesure de centralité :** Les mesures de centralité globales sont les plus pertinentes. Nous fixons ce paramètre par la centalité d'intermédiarité. Ce choix est motivé par le fait que cette dernière est basée sur l'idée de contrôle exercé par le nœud sur les

1. `http://igraph.sourceforge.net`
2. `http://www.numpy.org/`

interactions entre les autres nœuds ; deux nœuds non-adjacents dépendent d'un autre nœuds pour s'interagir.

— **Calcul d'appartenance :** Pour ce paramètre, on a deux mesures basées sur la distance entre les nœuds : le plus court chemin et Katz tronquée. Nous choisissons la distance la plus basique qui est le plus court chemin.

— **Fusion de votes :** La fusion des vecteurs d'appartenance par les deux méthodes Borda et Majorité, est basée sur le classement individuel des communautés. À l'inverse, Kemeny suit un calcul plus complexe et prend en compte le bon et le mauvais classement des communautés. Par ailleurs, l'appartenance des nœuds se trouvant aux extrémités des communautés sera probablement perturbée par l'appartenance des nœuds voisins si nous utilisons Borda ou Majorité. Nous choisissons donc Kemeny comme méthode par défaut.

Ainsi, la configuration par défaut de LICOD est : $\sigma = 0.9$, Mesure de centralité= B, Calcul d'appartenance= SP, Fusion de votes=Kemeny

Nous appliquons protocole expérimental décrit ci-haut sur un ensemble de réseaux réels et artificiels. Les indices de performance calculés sont le NMI et l'ARI.

3.2.2.1 Benchmarks utilisés

Réseaux réels

Nous avons testé LICOD sur 4 réseaux réels dont on connait la vérité de terrain (voir section 2.4.2 du chapitre 2 pour plus de détails). Le tableau 3.2 décrit leurs principales caractéristiques.

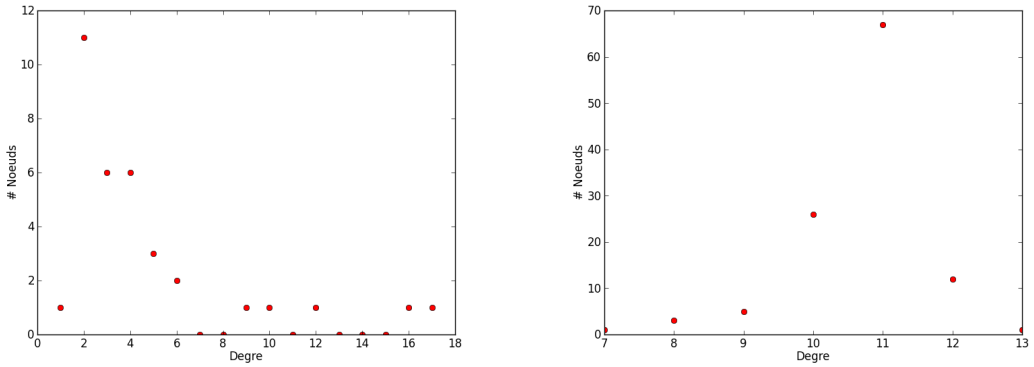
Réseau	**n**	**m**	**#communautés**
Club de Karaté de Zachary	34	78	2
Football	115	616	11
Livres politiques	100	411	3
Dauphins	62	159	2

TABLE 3.2 – Caractéristiques principales des réseaux

Nous représentons également les caractéristiques topologiques de ces réseaux dans le tableau 3.3. Nous remarquons que la densité est faible et le coefficient de clustering plutôt élevé. Le paramètre γ est celui de la loi de puissance estimée pour la distribution degré des réseaux. On aperçoit clairement dans la figure 3.2 la présence de plusieurs nœuds ayant un degré faible et peu de nœuds ayant un degré élevé dans ces deux réseaux. Toutefois, dans les réseaux Football et Dauphins, la distribution est différente, avec plusieurs nœuds ayants des degrés proches du degré moyen.

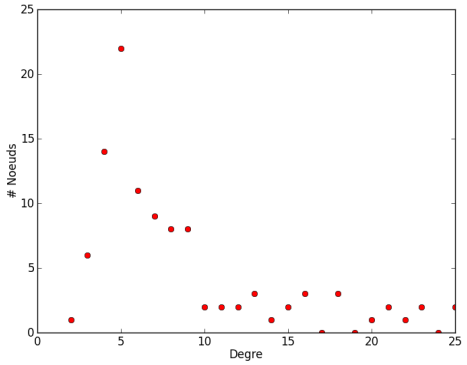
	Zachary	Football	Livres politiques	Dauphins
Diamètre	5	4	7	8
Densité	0.139	0.093	0.081	0.084
Coefficient de clustering	0.588	0.403	0.487	0.302
γ loi de puissance	2.12	22.46	2.62	7.70

TABLE 3.3 – Caractéristiques topologiques des réseaux

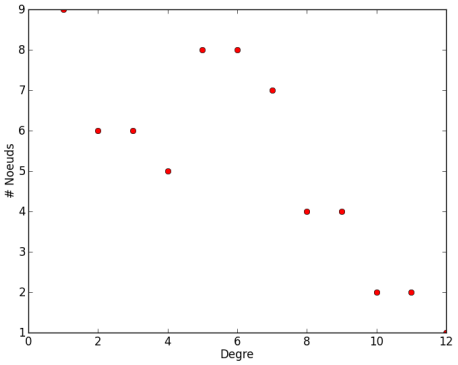


(a) Zachary

(b) Football



(c) Livres politiques



(d) Dauphins

FIGURE 3.2 – Distribution de degré des réseaux réels

Générateur de graphes artificiels utilisé : le modèle LFR

En suivant le même protocole expérimental, nous expérimentons LICOD sur le générateur de graphes LFR [Lancichinetti *et al.*, 2008]. Nous présentons la configuration de base de ce générateur dans le tableau 3.4.

Paramètre	Signfication	valeur
N	Nombre de nœuds	1000
k	Degré moyen des nœuds	20
k_{max}	Degré maximal des nœuds	70
μ	Coefficient de mélange	0.1

TABLE 3.4 – Configuration des principaux paramètres du générateur de graphe LFR

3.2.2.2 Comparaison des différentes configurations de LICOD

Résultats obtenus sur les réseaux réels

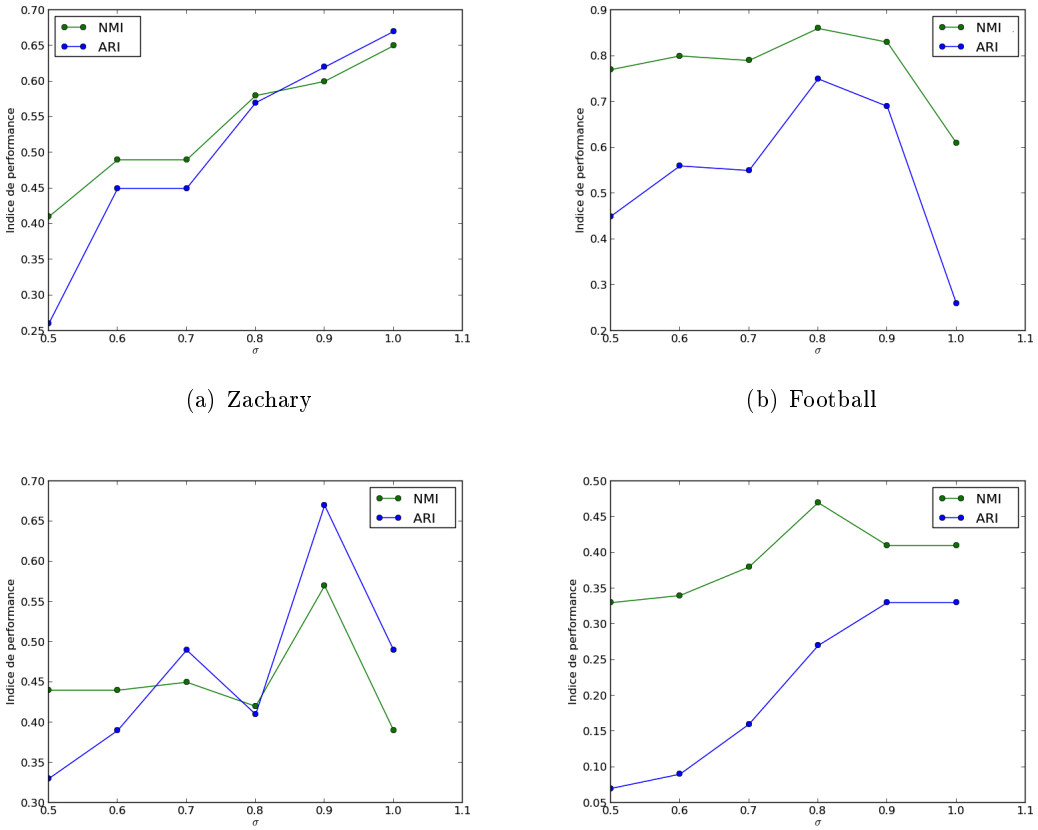
Nous discutons ci-dessous les résultats de variations des quatre paramètres de LICOD sur l'ensemble des réseaux réels.

Variation de σ

La figure 3.3 montre l'évolution des indices de performance, NMI et ARI, en fonction du paramètre σ. Pour le réseau Zachary, le maximum est atteint quand $\sigma = 1$ (c.à.d : le leader est celui qui a le maximum de centralité par rapport à ses voisins). Ceci est expliqué par la domination des deux nœuds 1 et 34 (voir figure 2.3(a)) en terme de centralité d'intermédiarité. Dans le réseau Livres politiques, le maximum de performance est obtenu à $\sigma = 0.9$, par contre cette performance se dégrade quand $\sigma = 1$. On peut voir cette chute aussi dans le réseau Football quand σ passe de 0.9 à 1. Ces résultats montrent que le choix des nœuds leaders diffère d'un réseau à un autre. Toutefois, LICOD trouve de bons résultats pour des valeurs de σ comprises entre 0.8 et 0.9. Ceci valide l'idée d'introduire le paramètre σ et ne pas considérer les cas extrêmes où un nœud est qualifié comme un leader si il a le maximum de centralité dans son voisinage direct.

Variation de la mesure de centralité :

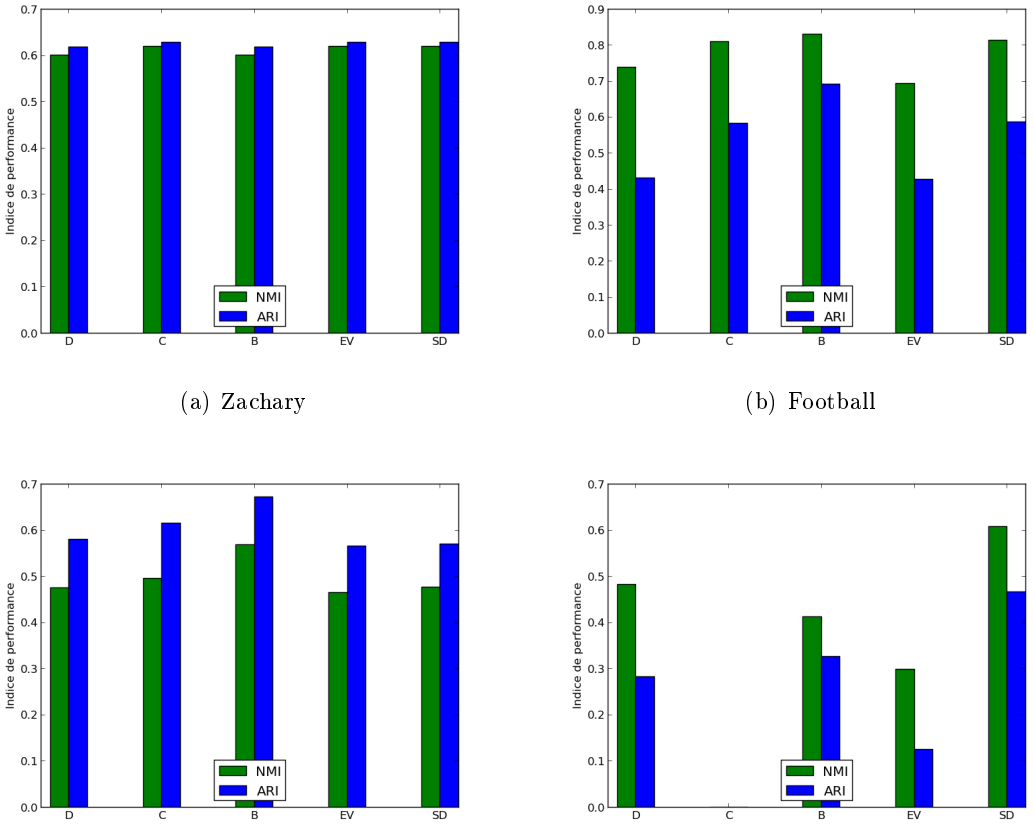
Nous montrons dans la figure 3.4 le niveau de performance de LICOD, en terme de NMI et ARI, avec les différentes mesures de centralité. Les résultats obtenus sur le réseau Zachary sont identiques, ceci est expliqué par la forte présence des nœuds leaders (voir figure 2.3(a)). Dans les deux réseaux Football et Livres politiques, la centralité d'intermédiarité dépasse les autres centralités en terme de NMI et ARI. On remarque aussi que la centralité semi-locale SD a donné un bon résultat dans le réseau Dauphin. Ceci est dû à la distribution de degré un peu particulière de ce réseau par rapport aux autres réseaux (voir figure 3.2(d)), où la majorité des degrés sont proches du degré moyen. En contrepartie, LICOD-centralité de proximité a échoué de trouver une partition dans ce réseau. L'absence de leaders est dû principalement au seuil utilisé pour σ, 0.9, et aux valeurs très proches de la centralité de proximité des nœuds. Ce qui confirme que la nature de distribution des degrés a un effet important sur la pertinence des mesures de centralité. Dans l'ensemble, nous constatons que LICOD-centralité d'intermédiarité est plus performant en terme de NMI et ARI qu'avec les autres mesures de centralité.



(a) Zachary (b) Football

(c) Livres politiques (d) Dauphin

FIGURE 3.3 – LICOD : Évolution de NMI et ARI en fonction de σ sur les réseaux réels



(a) Zachary

(b) Football

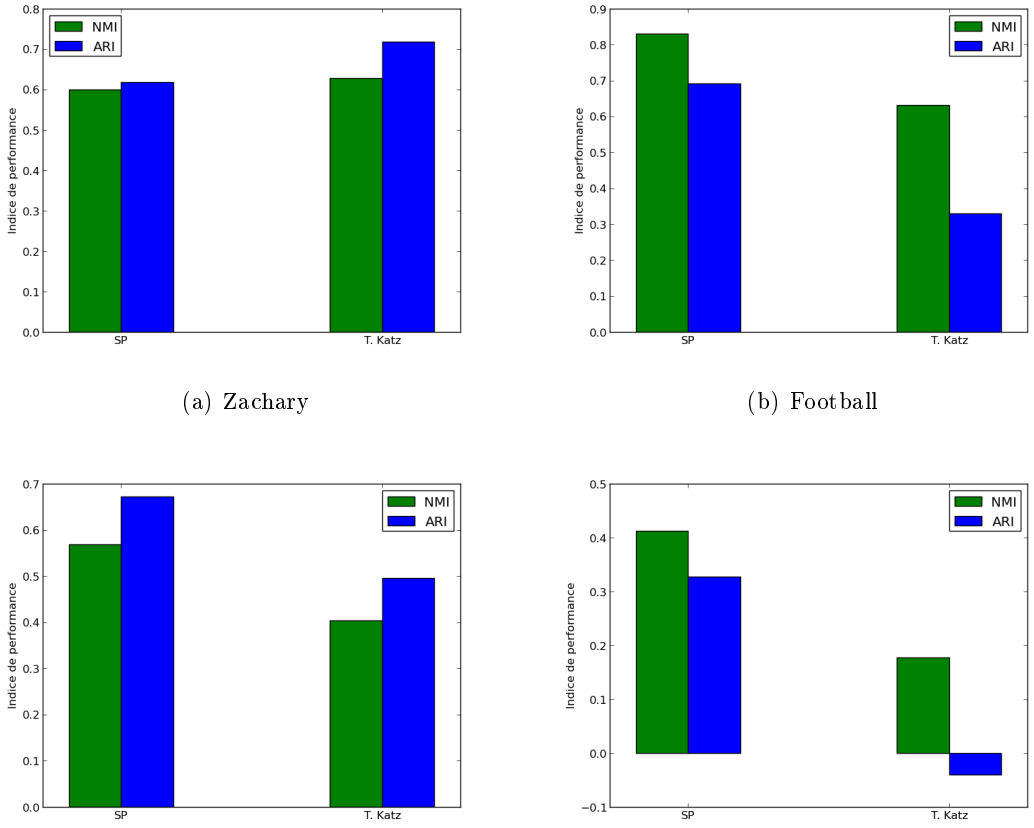
(c) Livres politiques

(d) Dauphin

FIGURE 3.4 – LICOD : Variation de NMI et ARI en fonction de la mesure de centralité (D : centralité de degré, SD : centralité de degré des voisins, B : centralité d'intermédiarité, C : centralité de proximité, EV : centralité de vecteurs propres)

Variation de la fonction de calcul du degré d'appartenance *Dist*()

Nous présentons dans la figure 3.5 le résultat de LICOD en terme de NMI et ARI avec les deux méthodes de calcul d'appartenance : le plus court chemin (SP) et la mesure de Katz tronquée (TKatz). Cette figure montre que, sur l'ensemble des quatre réseaux, LICOD-SP est plus performant que LICOD-TKatz. Ce résultat est expliqué par le fait que le calcul d'appartenance communautaire en s'appuyant seulement sur les plus courts chemins, est plus efficace que si on le calcule à base de tous les chemins reliant les nœuds aux leaders.



(a) Zachary (b) Football

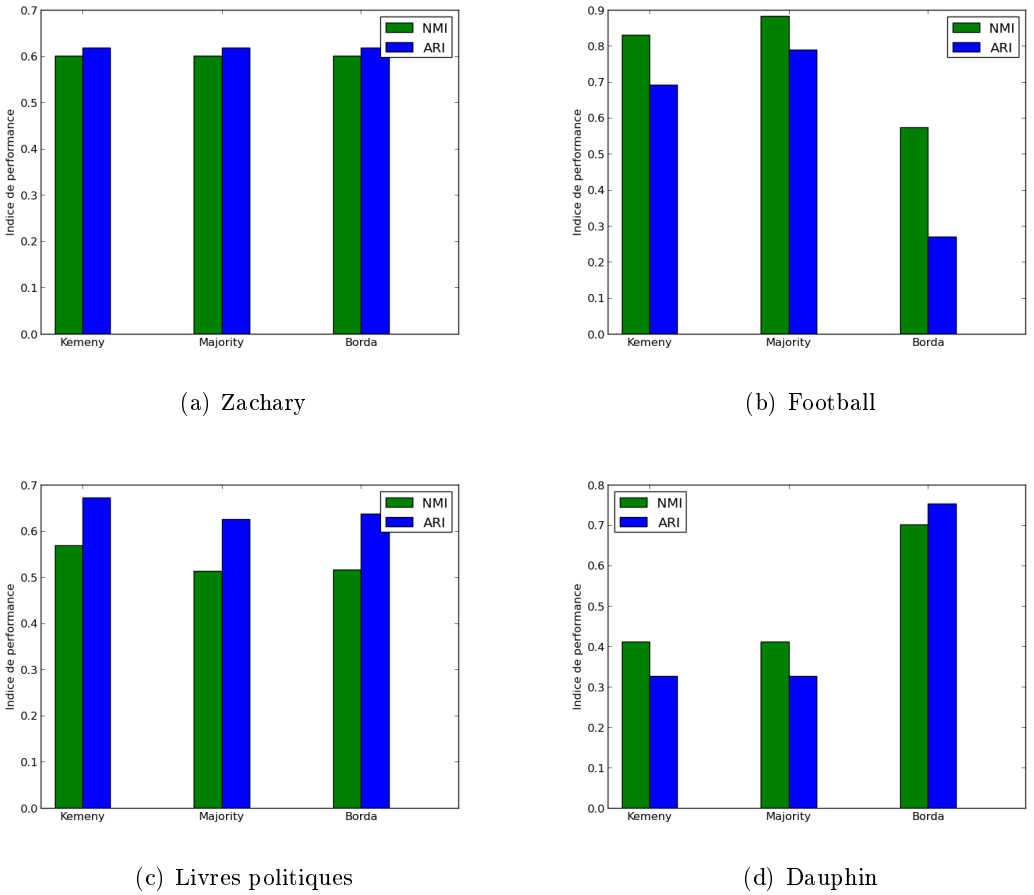
(c) Livres politiques (d) Dauphin

FIGURE 3.5 – LICOD : Variation de NMI et ARI en fonction de la mesure de *Dist*() (SP : plus court chemin, T. Katz : Katz tronquée)

Variation de la méthode de fusion des votes

Le résultat de l'étape de fusion des votes des voisins concernant l'appartenance communautaire des nœud est présenté dans la figure 3.6. Mis à part le réseau Zachary, où les trois méthodes ont donné les même scores de NMI et ARI, le résultat différent d'un réseau

à un autre ; dans le réseau Football, LICOD obtient des bons résultats seulement avec Kemeny et Majorité, alors que c'est l'inverse dans le réseau Dauphin. Dans Livres politiques, Kemeny dépasse légèrement les deux autres méthodes. D'après ces résultats, on constate que la méthode parfaite n'existe pas, et que les caractéristiques topologiques notamment la distribution de degré rend le réseau étudié un cas particulier.



(a) Zachary (b) Football

(c) Livres politiques (d) Dauphin

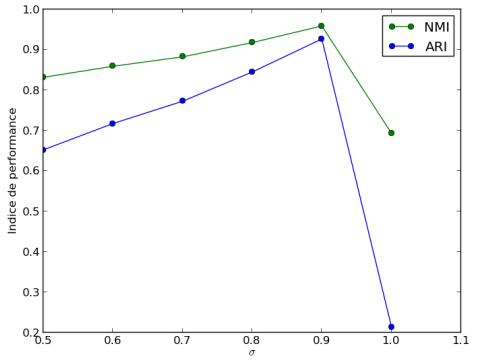
FIGURE 3.6 – LICOD : Variation de NMI et ARI en fonction de la méthode de vote

Résultats obtenus sur LFR

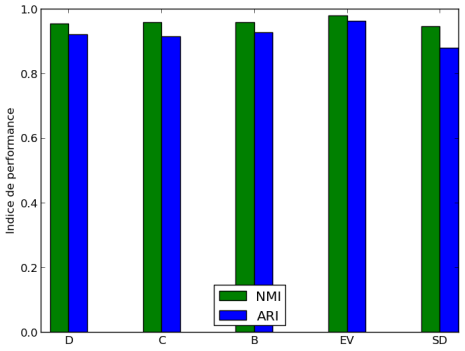
Dans la figure 3.7, nous présentons les résultats du protocole expérimental décrit ci-haut sur un graphe généré par le modèle LFR. La configuration de ce dernier est la suivante : $N = 1000$, $k = 20$, $k_{max} = 70$ et $\mu = 0.1$.

Le résultat de variation du σ confirme que LICOD atteint son maximum de performance à $\sigma = 0.9$. De même, LICOD avec le plus court chemin est plus pertinent qu'avec la mesure TKatz. Concernant les méthodes de fusion de vote, l'utilisation de Kemeny et Majorité donne des scores de NMI et ARI très élevés par rapport à ceux trouvés par la méthode

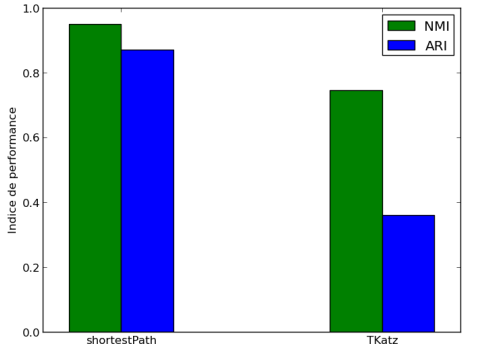
Borda. Quand à la variation des mesures de centralité, l'expérimentation a donné des scores très proches.



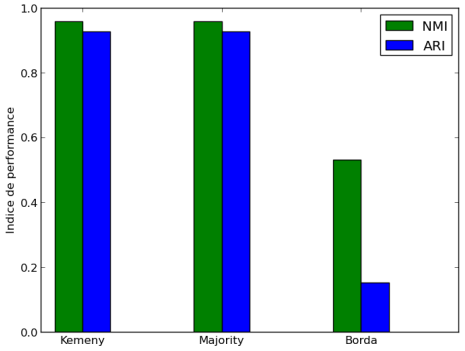
(a) NMI, ARI en fonction de σ



(b) NMI, ARI en fonction de centralité



(c) NMI, ARI en fonction de $Dist()$



(d) NMI, ARI en fonction de la méthode de fusion de votes

FIGURE 3.7 – Comparaison des différentes configurations de LICOD sur le graphe LFR ($N = 1000$, $k = 20$, $k_{max} = 70$ et $\mu = 0.1$)

3.2.3 Discussion

Nous avons montré les résultats de LICOD en faisant varier séparément ses quatre paramètres. Sur l'ensemble des réseaux réels et artificiels, nous avons constaté que le seuil 0.9 est une bonne valeur de σ. De la même façon, nous avons démontré que, dans la majorité des cas, la centralité d'intermédiarité (B) rend LICOD plus compétitif et plus stable. Pour le calcul d'appartenance des nœuds aux leaders, la distance géodésique est plus efficace que T. Katz. Le choix de la meilleure méthode de fusion de votes était une étape non triviale à cause des résultats contradictoires obtenus dans les deux réseaux Football et Dauphin. L'étape de fusion de votes dépend de la nature de la distribution des degrés du réseau, et puisque le modèle LFR permet de générer des graphes ayant une distribution de degré qui

suit la loi de puissance (qui représente une des caractéristiques de la plupart des réseaux complexes du monde réel), nous l'avons choisi comme repère et ainsi, nous considérons que les deux méthode Kemeny et le vote majoritaire sont les meilleures.

La performance de LICOD dépend fortement des leaders identifiés, cependant, l'algorithme ne vérifie pas si les leaders sélectionnés sont vraiment les leaders de leurs communautés. Autrement dit, il manque à LICOD une procédure d'auto-évaluation des leaders identifiés. Dans la suite, nous proposons une extension itérative de LICOD qui prend en compte cette limite.

3.3 Extension itérative de LICOD : it-LICOD

Dans cette section, nous introduisons une extension itérative de LICOD, appelée it-LICOD, qui traite le problème de validation des leaders. Notre hypothèse est qu'au lieu d'identifier les leaders en appliquant seulement les mesures de centralité, on ajoute une étape itérative qui calcule les leaders à l'échelle des communautés identifiées dans l'étape initiale. Cette étape est considérée comme une étape d'auto-évaluation du choix des leaders.

3.3.1 L'approche it-LICOD

Nous présentons ci-dessous le pseudo-code de it-LICOD. La version algorithmique est présentée dans l'algorithme 3.

L'approche est composée de quatre principales étapes : soit $G = (V, E)$ un graphe non-dirigé :

1. Appliquer LICOD sur G pour trouver la partition initiale. Soit $\mathcal{L}_t$ et $\mathcal{C}_t$ l'ensemble de leaders et la partition trouvés (ligne 4) ;
2. Considérer que chaque communauté $c \in \mathcal{C}_t$ est un graphe et identifier ses leaders via la fonction $nouveauxLeaders()$. Soit $\mathcal{L}_{t+1}$ l'ensemble des leaders de toutes les communautés dans $\mathcal{C}_t$ (lignes 7-9) ;
3. Appliquer les quatre dernières étapes de LICOD : regroupement de leaders, calcul d'appartenance communautaire, fusion de votes et affectation aux communautés finales, soit $\mathcal{C}_{t+1}$ la nouvelle partition trouvées (lignes 14-33) ;
4. it-LICOD itère sur les étapes 2-4 jusqu'à la stabilisation de l'ensemble de leaders. Pour que it-LICOD converge rapidement, nous calculons également la pureté entre chaque deux partitions. it-LICOD s'arrête quand $\mathcal{L}_{t+1} = \mathcal{L}_t$ ou $PUR(\mathcal{C}_t, \mathcal{C}_{t+1}) >= \beta$. À la fin, it-LICOD retourne $\mathcal{C}_{t+1}$.

3.3.2 Expérimentation

L'expérimentation de it-LICOD est réalisée sur les mêmes réseaux benchmarks utilisés dans l'expérimentation de LICOD. Elle est faite en deux phases : Dans la première phase, nous comparons it-LICOD et LICOD en fonction des différentes configurations des paramètres. Deuxièmement, nous comparons leurs performance par rapport à un ensemble

Algorithme 3 it-LICOD

Entrée : Un graphe connexe $G =< V, E >$
Sortie : Une partition $\mathcal{C}$
Début
$\mathcal{L}_t, \mathcal{C}_t \leftarrow LICOD(G)$
$\mathcal{L}_{t+1}, \mathcal{C}_{t+1} \leftarrow \emptyset$
tant que $TROUVE1 = False$ et $TROUVE2 = False$ **faire**
  pour $c \in C_t$ **faire**
    $\mathcal{L}_{t+1} \leftarrow \mathcal{L}_{t+1} \cup \{nouveauxLeaders(c)\}$
  fin pour
  si $\mathcal{L}_{t+1} = \mathcal{L}_t$ **alors**
    $TROUVE1 = True$
  fin si
  $\mathcal{L}_t \leftarrow \mathcal{L}_{t+1}$
  $\mathcal{C}_{t+1} \leftarrow LeadersCommunautes(\mathcal{L}_{t+1})$
  pour $v \in V$ **faire**
    pour $c \in \mathcal{C}_{t+1}$ **faire**
      $M[v, c] \leftarrow Appartenance(v, c)$
    fin pour
    $P[v] =$ **TrierEtClasser(**$M[v]$**)**
  fin pour
  repeat
    pour $v \in V$ **faire**
      $P^*[v] \leftarrow$ **Fusion**$_{\mathbf{x} \in \{\mathbf{v}\} \cap \mathbf{\Gamma(v)}}$**P[x]**
      $P[v] \leftarrow P^*[v]$
    fin pour
  until Stabilisation de $P^*[v] \forall v$
  pour $v \in V$ **faire**
    pour $c \in P[v]$ **faire**
      si $|M[v, c] - M[v, P[0]]| \leq \epsilon$ **alors**
        $COM(c) \leftarrow COM(c) \cup \{v\}$
      fin si
    fin pour
  fin pour
  si $PUR(\mathcal{C}_t, \mathcal{C}_{t+1}) \geq \beta$ **alors**
    $TROUVE2 = True$
  fin si
  $\mathcal{C}_t \leftarrow \mathcal{C}_{t+1}$
fin tant que
Retourne $\mathcal{C}_{t+1}$
Fin

d'algorithmes de référence. Les indices de performance calculés sont le NMI, l'ARI, et le nombre de communautés (*#com*).

3.3.2.1 it-LICOD vs LICOD

Nous allons effectuer le même protocole expérimental que pour LICOD, sauf que cette fois nous allons comparer, pour les différentes configurations des paramètres, la performance de it-LICOD et LICOD. Pour des raisons d'espace, nous notons les deux algorithmes LICOD et it-LICOD respectivement par L et it-L. Nous adoptons également la même configuration par défaut pour les deux algorithmes :

- $\sigma = \delta = 0.9$
- Mesure de centralité : B
- Calcul d'appartenance : SP
- Fusion de votes : Kemeny
- $\beta = 0.9$ (pour it-LICOD)

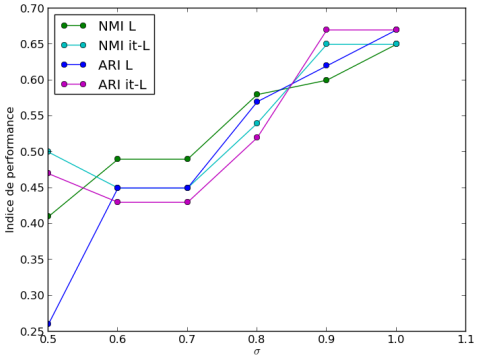
Performance de it-LICOD vs LICOD en fonction de σ

Nous montrons dans la figure 3.8 l'évolution de NMI et ARI de it-LICOD et LICOD en fonction de σ. Pour Zachary, it-LICOD converge vers la performance maximale (σ =0.9) avant LICOD, sinon, il n'y pas différence notable entre la performance des deux algorithmes pour les autres valeurs de σ. Dans le réseau Football, LICOD dépasse légèrement it-LICOD pour les deux valeurs 0.8 et 0.9, mais ce dernier est plus stable et plus performant pour les autres valeurs de σ. Nous remarquons la même observation pour le réseau Livres politiques. Pour le réseau Dauphin, it-LICOD est plus performant pour toutes les valeurs de σ.

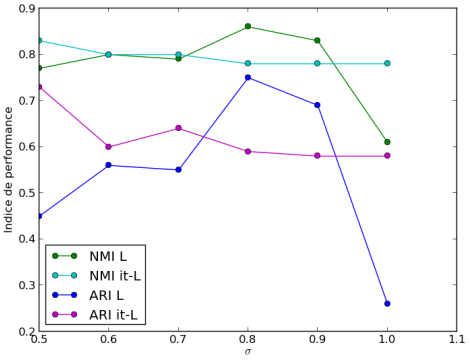
Le résultat obtenu sur le graphe LFR (N=1000, k=0.1, μ = 0.1) montre une bonne performance de it-LICOD par rapport à LICOD pour toute les valeurs de σ entre 0.5 et 0.9. Ce résultat met en évidence l'importance de l'étape d'auto-évaluation des leaders introduite par it-LICOD.

Performance de it-LICOD vs LICOD en fonction de la mesure de centralité

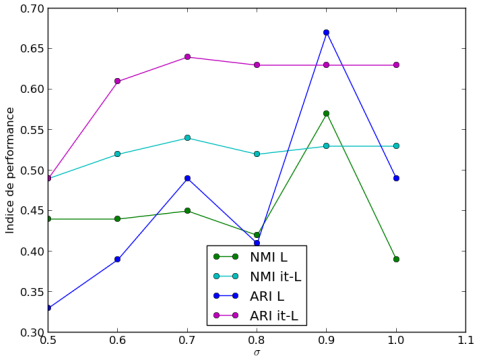
Les scores de NMI et ARI des deux méthodes, obtenus sur les réseaux réels et le graphe LFR, sont affichés dans le tableau 3.5 On remarque que dans les trois réseaux Football, Livres politiques et Dauphins, it-LICOD est généralement plus performant avec la mesure de centralité locale (D) et la mesure de centralité semi-locale (SD) que LICOD. Cela explique que ce genre de mesures plus au moins locale captent mieux les leaders lorsqu'on les applique seulement aux réseaux qui représentent les communautés. Quand aux mesures globales (B et EV), on remarque qu'elles sont plus efficaces avec it-LICOD dans le réseau Dauphin, sachant que ce réseau possède en vérité de terrain seulement 2 communautés. L'utilisation de la centralité de proximité (C) n'a pas permis it-LICOD de partitionner le réseau Livres politiques et Dauphins. Ceci est dû à l'absence de leaders avec le seuil de σ choisi (0.9).



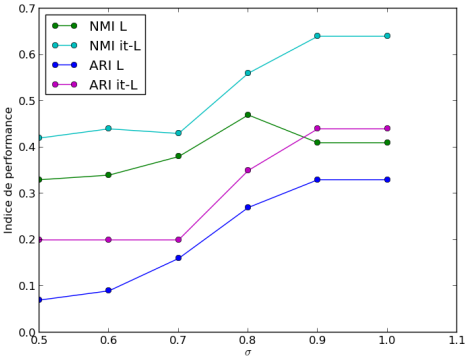
(a) Zachary



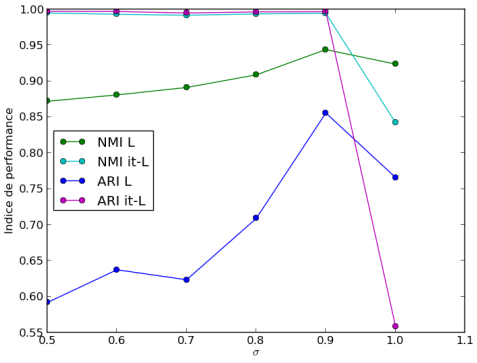
(b) Football



(c) Livres politiques



(d) Dauphin



(e) Graphe LFR : N=1000, k=20, $\mu = 0.1$

FIGURE 3.8 – it-LICOD (it-L) vs LICOD (L) : Évolution de NMI et ARI en fonction de σ

Le choix du type de la mesure de centralité, locale/semi-locale ou globale, à appliquer dans it-LICOD n'est pas simple puisqu'il est difficile d'estimer la taille des communautés.

Globalement, et comme dans l'expérimentation précédente, la variation de mesure de centralité pour LICOD et it-LICOD sur les 4 réseaux réels a montré que, mis à part le réseau de Football, it-LICOD obtient des bons résultats par rapport à LICOD, sur Zachary, Livres politiques et surtout Dauphins. Sur le graphe LFR, it-LICOD dépasse LICOD et obtient des bons scores de NMI et ARI.

		D		C		B		EV		SD	
Réseau	Score	L	it-LI	L	it-L	L	it-L	L	it-L	L	it-L
Zachary	NMI	0.60	**0.65**	0.62	**0.65**	0.60	**0.65**	0.62	**0.65**	0.62	**0.65**
	ARI	0.62	**0.67**	0.63	**0.67**	0.62	**0.67**	0.63	**0.67**	0.63	**0.67**
	#com	3	**2**	3	**2**	3	**2**	3	**2**	3	**2**
Football	NMI	0.74	0.74	**0.81**	0.78	**0.83**	0.77	**0.83**	0.61	0.69	**0.79**
	ARI	0.43	**0.50**	0.58	**0.59**	**0.69**	0.58	**0.69**	0.30	0.59	0.59
	#com	52	34	14	29	16	21	8	4	15	36
Livres Politiques	NMI	0.47	**0.57**	**0.49**	0	**0.57**	0.53	0.47	**0.57**	0.48	**0.57**
	ARI	0.58	**0.64**	**0.62**	0	**0.67**	0.63	0.58	**0.64**	0.57	**0.64**
	#com	9	2	5	1	6	3	7	2	6	2
Dauphin	NMI	0.48	0.48	0	0	0.41	**0.64**	0.30	**0.64**	**0.60**	0.58
	ARI	0.28	**0.69**	0	0	0.33	**0.44**	0.13	**0.44**	0.47	**0.54**
	#com	7	6	1	1	2	4	3	4	4	3
Graphe LFR	NMI	0.95	**0.99**	0.94	**0.99**	0.95	**0.99**	0.94	**0.96**	0.93	**0.99**
N=1000, k=20, $\sigma = 0.1$	ARI	0.90	**0.99**	0.86	**0.96**	0.91	**0.99**	0.82	**0.84**	0.78	**0.98**

TABLE 3.5 – Comparaison des résultats de it-LICOD et LICOD en fonction de la mesure de centraltié

Performance de it-LICOD vs LICOD en fonction de *Dist*()

Les résultats de variation de la mesure de calcul d'appartenance *Dist*() a montré que comme dans LICOD, it-LICOD est plus efficace avec SP qu'avec T. Katz (voir tableau 3.6). Par ailleurs, it-LICOD obtient des bons résultats par rapport à LICOD dans Zachary, Dauphins et le graphe LFR, ce que nous avons déjà trouvé dans les deux expérimentations précédentes.

Performance de it-LICOD vs LICOD en fonction de la méthode de vote

Pour l'étape de fusion de votes des voisins sur l'appartenance communautaire des nœuds, la performance de it-LICOD par rapport à LICOD dans Zachary, Dauphins et le graphe LFR se confirme encore (voir tableau 3.7). À l'exception du réseau Dauphin, où it-LICOD à une de performance en baisse avec la méthode Borda, la version itérative de l'algorithme gagne en performance avec toutes les méthodes de fusion de votes dans ces trois réseaux. Dans les deux réseaux Football et Livres politiques, les deux algorithmes

		SP		T. Katz	
Réseau	Score	L	it-LI	L	it-L
Zachary	NMI	0.60	**0.65**	**0.63**	0.10
	ARI	0.62	**0.67**	**0.71**	0.03
#com	3	2	3	4	
Football	NMI	**0.83**	0.77	0.63	**0.68**
	ARI	**0.69**	0.58	0.33	**0.41**
	#com	16	21	16	34
Livres Politiques	NMI	**0.57**	0.53	0.40	**0.45**
	ARI	**0.67**	0.63	0.49	**0.56**
	#com	6	3	6	3
Dauphin	NMI	0.41	**0.64**	0.17	**0.53**
	ARI	0.33	**0.44**	-0.04	**0.63**
	#com	2	4	2	4
Graphe LFR	NMI	0.95	**0.99**	**0.91**	0.57
N=1000, k=20, $\sigma = 0.1$	ARI	0.70	**0.99**	**0.33**	0.19

TABLE 3.6 – Comparaison des résultats de it-LICOD et LICOD en fonction de $Dist()$

ont obtenu des résultats proches. Dans le premier réseau, LICOD dépasse légèrement it-LICOD avec la méthode Kemeny et le vote majoritaire, alors que ce dernier dépasse avec la méthode de borda. Quand au réseau Livres politiques, it-LICOD, avec le vote majoritaire et borda, obtient un score de NMI légèrement supérieur à celui de LICOD, alors que ce dernier le dépasse avec la méthode de Kemeny.

En résumé, it-LICOD est plus performant dans les réseaux Zachary, Dauphins et le graphe artificiel de LFR. Il a obtenu également des résultats compétitifs avec ceux de LICOD dans les deux autres réseaux. Ces résultats encourageants valide notre hypothèses sur l'étape d'identification de leaders. Dans la suite nous comparons les deux algorithmes LICOD et it-LICOD aux autres algorithmes de détection de communautés de référence.

3.3.2.2 Validation

Nous adoptons la même configuration de base de LICOD ($\sigma = \delta = 0.9$, mesure de centralité : B, $Dist()$=SP, méthode de vote : Kemeny) pour comparer sa performance avec des célèbres algorithmes de détection de communautés. Ces algorithmes sont :

— Girvan-Newman (GN) [Girvan et Newman, 2002], voir chapitre1 section 1.4.2.

— Walktrap [Pons et Latapy, 2004], voir chapitre 1 section 1.4.2.

— EV [Newman, 2006b], voir chapitre 1 section 1.4.2.

— InfoMap [Rosvall et Bergstrom, 2008], voir chapitre 1 section 1.4.4.

— FastGreedy [Clauset *et al.*, 2004], voir chapitre 1 section 1.4.2.

— Méthode de Louvain [Blondel *et al.*, 2008], voir chapitre 1 section 1.4.2.

		Kemeny		Majorité		Borda	
Réseau	Score	L	it-LI	L	it-L	L	it-L
Zachary	NMI	0.60	**0.65**	0.60	**0.65**	0.60	**0.65**
	ARI	061	**0.67**	0.61	**0.67**	0.61	**0.67**
#com	3	**2**	3	**2**	3	**2**	
Football	NMI	**0.83**	0.77	**0.88**	0.84	0.57	**0.71**
	ARI	**0.69**	0.58	**0.78**	0.70	0.26	**0.46**
	#com	17	21	13	22	**11**	26
Livres Politiques	NMI	**0.56**	0.53	0.51	**0.53**	0.52	**0.53**
	ARI	**0.67**	0.63	**0.63**	**0.63**	**0.64**	0.63
	#com	6	3	6	3	6	3
Dauphin	NMI	0.41	**0.64**	0.41	**0.64**	**0.70**	0.64
	ARI	0.33	**0.44**	0.33	**0.44**	**0.75**	0.44
	#com	**2**	4	**2**	4	**2**	4
Graphe LFR	NMI	0.95	**0.99**	0.94	**0.99**	0.62	**0.91**
N=1000, k=20, $\sigma = 0.1$	ARI	0.91	**0.99**	0.89	**0.99**	0.24	**0.69**

TABLE 3.7 – Comparaison des résultats de it-LICOD et LICOD en fonction de la méthode de vote

La comparaison a été faite sur les 4 réseaux réels et sur des graphes issus du générateur LFR avec différentes configurations.

Validation sur les réseaux réels

Le tableau 3.8 montre les scores obtenus par les différents algorithmes sur les 4 réseaux réels. La compétitivité des algorithmes varie d'un réseau à un autre. LICOD et it-LICOD obtiennent respectivement les meilleurs scores dans Livres politiques et Dauphins. Le résultat obtenu par it-LICOD sur Zachary est compétitif par rapport aux autres algorithmes. Le mauvais résultat trouvé par it-LICOD et LICOD sur Football est dû à la distribution de degré de ce réseau qui ne suit pas une loi de puissance. Alors que les algorithmes basés sur les leaders sont plus efficaces sur des réseaux sans-échelle (peu de nœuds ayant un degré élevé et beaucoup de nœuds ayant un degré faible). Dans la suite, nous présentons les résultats de comparaison sur le modèle LFR.

Validation sur LFR

La deuxième partie de comparaison de it-LICOD aux autres algorithmes de détection de communautés est faite sur des graphes artificiels générés par le modèle LFR. Cette comparaison est effectuée sur trois expérimentations. Nous varions à chaque fois un de ces trois paramètres : N, k, μ. La variation du nombre de nœuds N permet de tester la performance des algorithmes sur des réseaux de taille importante. Alors que le changement du degré moyen k sert à évaluer leurs performances sur des réseaux de densités différentes. Dans la troisième expérimentation, nous varions le coefficient de mélange μ afin d'évaluer les algorithmes sur communautés plus aux moins dégagées.

Réseau	Algo	NMI	ARI	#communautés
Zachary	GN	0.57	0.46	5
	Walktrap	0.50	0.33	5
	EV	0.67	0.51	4
	InfoMap	**0.69**	**0.70**	3
	FastGreedy	**0.69**	0.68	3
	Louvain	0.58	0.46	4
	LICOD	0.60	0.62	3
	it-LICOD	0.65	0.67	2
Football	GN	0.87	0.77	10
	Walktrap	0.88	0.81	10
	EV	0.69	0.46	8
	InfoMap	**0.92**	**0.89**	12
	FastGreedy	0.69	0.47	6
	Louvain	0.89	0.80	10
	LICOD	0.83	0.69	16
	it-LICOD	0.77	0.69	21
Livres Politiques	GN	0.55	**0.68**	5
	Walktrap	0.53	0.65	4
	EV	0.49	0.54	4
	InfoMap	0.53	0.53	6
	FastGreedy	0.51	0.63	4
	Louvain	0.57	0.55	4
	LICOD	**0.58**	**0.67**	6
	it-LICOD	0.53	0.63	3
Dauphins	GN	0.55	0.39	5
	Walktrap	0.53	0.41	4
	EV	0.44	0.28	5
	InfoMap	0.48	0.29	6
	FastGreedy	0.57	**0.45**	4
	Louvain	0.51	0.32	5
	LICOD	0.41	0.32	2
	it-LICOD	**0.64**	**0.45**	4

TABLE 3.8 – Performance de it-LICOD par rapport au LICOD, GN, Walktrap, EV, InfoMap, FastGreedy, et Louvain. La configuration de de LICOD et it-LICOD est celle par défaut : $\sigma = \delta = 0.9$, centralité= B, calcul d'appartenance= SP ,fusion de votes=Kemeny

Les plages de valeurs des trois paramètres sont les suivants :

1. $N = [500, 1000, 1500, 2000, 2500]$.
2. $k = [20, 30, 40, 50, 60]$
3. $\mu = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$

Les résultats des expérimentation n°1, n°2 et n°3 sont présentés respectivement dans les figures suivantes : 3.9, 3.10 et 3.11.

Dans les deux premières expérimentations (figures 3.9 et 3.10), on remarque que l'algorithme EV a une mauvaise performance pour toutes valeurs de N et k. L'algorithme LICOD montre des résultats supérieurs à 0.94 en terme de NMI tout au long de la variation de ces deux paramètres, et il dépasse l'algorithme FastGreedy. Pour les autres algorithmes, y compris it-LICOD, ils ont eu des résultats très proches qui varient entre 0.99 et 1.

Concernant la troisième expérimentation (figure 3.11), it-LICOD possède des résultats compétitifs jusqu'à la valeur 0.3 de μ, après son score de NMI commence à chuter. On fait la même observation pour LICOD.

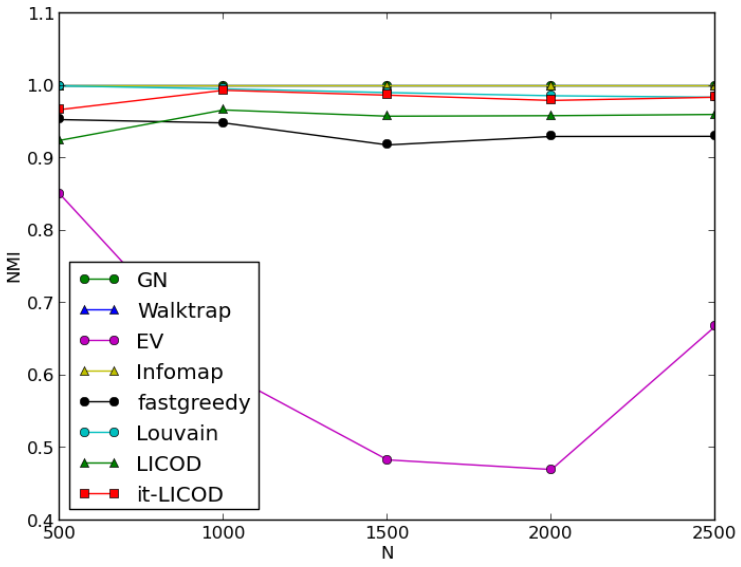


FIGURE 3.9 – Comparaison des résultats de it-LICOD, LICOD, GN, Walktrap, EV, InfoMap, FastGreedy, et Louvain : NMI vs N

3.4 Conclusion

Dans ce chapitre, nous avons proposé deux contributions :

Enrichissement de LICOD : comme nous l'avons montré dans l'état de l'art (chapitre 1) les algorithmes de détection de communautés guidés par l'identification de leaders

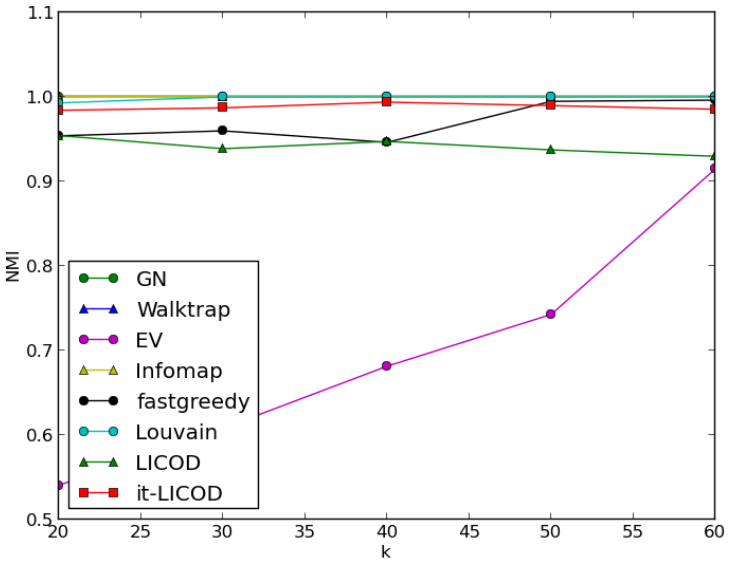


FIGURE 3.10 – Comparaison des résultats de it-LICOD, LICOD, GN, Walktrap, EV, InfoMap, FastGreedy, et Louvain : NMI vs k

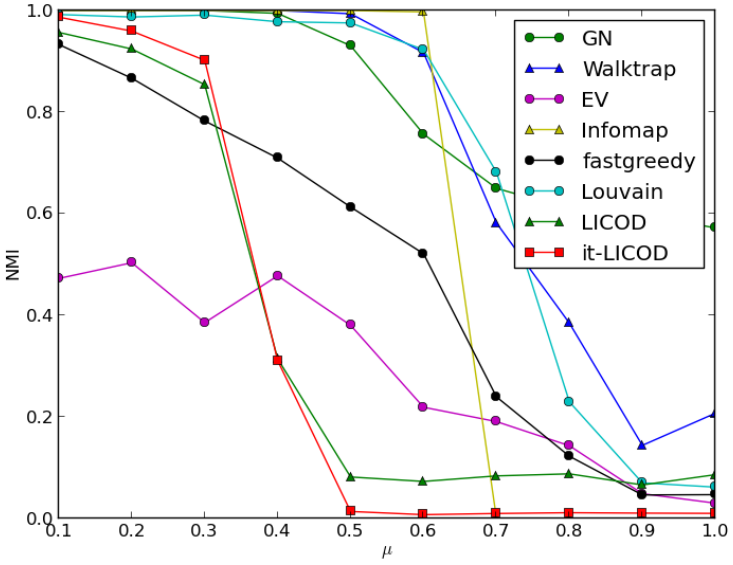


FIGURE 3.11 – Comparaison des résultats de it-LICOD, LICOD, GN, Walktrap, EV, InfoMap, FastGreedy, et Louvain : NMI vs μ

sont composés de deux grandes parties : identification de leaders et construction de communautés, et il existe plusieurs stratégies qui permettent de réaliser ces deux étapes. Dans ce chapitre, nous nous sommes concentrés sur la framework LICOD en introduisant l'intégration de nouvelles mesures pour chaque étape. Nous avons expérimenté sur des réseaux réels et artificiels les différentes stratégies d'utilisation des mesures utilisées dans les 4 étapes de LICOD.

Proposition d'une extension itérative de it-LICOD : . puisque la qualité des leaders a un impact direct sur la qualité des communautés, et que LICOD ne vérifie pas si l'ensemble des nœuds choisis comme leaders peut être amélioré ou pas, it-LICOD a été proposé pour ajouter une étape d'auto-évaluation des leaders identifiés. La comparaison de cet algorithme avec LICOD a montré un gain de performance dans plusieurs cas, surtout dans le cas où les communautés trouvés dans la première itération sont de taille importante. Notre algorithme it-LICOD a donné également des résultats encourageants par rapport aux autres algorithmes de détection de communautés.

Chapitre 4

TopoCent : une nouvelle mesure de centralité semi-locale

Sommaire

4.1 Introduction

L'identification des nœuds leaders est une étape primordiale dans les algorithmes de détection de communautés guidée par les leaders, comme l'approche LICOD. Les mesures utilisées s'appuient généralement sur le positionnement topologique des nœuds, que ce soit par rapport à tout le graphe (mesures globales), par rapport aux voisins directs (mesures locales), ou par rapport au k-ème niveau de voisins (mesures semi-locales). Les algorithmes de détection de communautés sont généralement, plus performants avec les mesures globales qu'avec les mesures locales. Cependant, les mesures globales sont coûteuses en temps de calcul et les mesures locales peuvent être un compromis entre ce coût et la non-efficacité des mesures locales. Nous proposons une mesure semi-locale *TopoCent* qui évalue la position d'un nœud en fonction des connexions pondérées par une similarité topologique. Le but de cette pondération est de donner plus d'influence aux nœuds proches dans le réseau.

4.2 L'approche TopoCent

TopoCent est caractérisée par deux propriétés principales :

— Aspect semi-local : l'importance d'un nœud dépend du nombre des voisins directs qui sont à leur tour évalués en fonction du nombre de leurs voisins. Le choix de se limiter au deuxième voisin (les voisins des voisins) pour quantifier l'importance d'un nœud est motivé par la faible pertinence de la centralité de degré et la complexité élevée des centralités globales.

— Aspect topologique : l'influence entre les voisins est calculée en fonction de leurs similarités topologiques. Plus le voisin est proche structurellement au nœud n_i, plus

son importance influe sur l'importance de n_i. Pour cela, nous avons choisi d'utiliser la fonction de similarité structurelle *sigma* introduite dans [Xu *et al.*, 2007] présentée dans le chapitre 1, section 1.15. Nous rappelons de cette fonction, soit $G = (V, E)$ un graphe, *sigma* est défini comme suit :

$$\sigma(v, w) = \frac{|\Gamma'(v) \cap \Gamma'(w)|}{\sqrt{|\Gamma'(v)| * |\Gamma'(w)|}} \tag{4.1}$$

où $\Gamma'(v)$ est la structure du nœud v, $\Gamma'(v) = \Gamma(v) \cup \{v\}$

L'algorithme 4 décrit le processus de calcul de *TopoCent*. La complexité de calcul est de l'ordre de $O(n(k)^2)$, avec n le nombre de nœuds et k le degré moyen du graphe.

Algorithme 4 TopoCent

Entrée : G
Sortie : $strenght_G$: un dictionnaire de centralités
Début
$strenght_G = \{\}$
pour $v \in V$ **faire**
  $s = 0$
  pour $n1 \in \Gamma(v) \backslash \{v\}$ **faire**
    $s1 = 0$
    pour $n2 \in \Gamma(n1) \backslash \{n1\}$ **faire**
      $s1 = s1 + \sigma(n1, n2)$
    fin pour
    $s = s + s1 * \sigma(v, n1)$
  fin pour
  $strenght_G[v] = s$
fin pour
Retourne $strenght_G$
Fin

4.3 Résultats expérimentaux

Afin de mieux étudier la performance de *TopoCent*, nous avons choisi de l'évaluer dans le cadre de LICOD. Nous comparons les résultats, en termes de NMI, ARI et nombre de communautés *#com*, de LICOD-TopoCent avec LICOD muni des autres mesures de centralités. Nous utilisons la centralité de degré (D), la centralité de degré des voisins (SD), la centralité de proximité (C), la centralité d'intermédiarité (B), la centralité de vecteurs propres (EV) et la centralité de PageRank. La configuration de LICOD est la suivante :

— $\sigma = \delta = 0.9$
— Calcul d'appartenance : le plus court chemin
— Fusion de votes : Kemeny

L'expérimentation est réalisée sur les quatre réseaux réels : Zachary, Football, Livres politiques et Dauphin, et sur des graphes LFR [Lancichinetti *et al.*, 2008].

4.3.1 Résultats sur les réseaux réels

Réseau	Mesure de centralité	NMI	ARI	#com
Zachary	D	0.60	0.61	3
	SD	0.61	0.62	3
	TopoCent	**0.65**	**0.65**	3
	C	0.61	0.62	3
	B	0.60	0.61	3
	EV	0.61	0.62	3
	PageRank	0.60	0.61	3
Football	D	0.74	0.43	52
	SD	0.81	0.59	15
	TopoCent	0.77	0.54	**11**
	C	0.81	0.58	14
	B	**0.83**	**0.69**	16
	EV	0.69	0.43	20
	PageRank	**0.83**	0.68	22
Livres politiques	D	0.47	0.58	9
	SD	0.47	0.57	6
	TopoCent	0.53	0.60	5
	C	0.49	0.61	5
	B	**0.56**	**0.67**	6
	EV	0.46	0.56	7
	PageRank	0.47	0.58	7
Dauphin	D	0.48	0.28	7
	SD	**0.61**	0.47	4
	TopoCent	0.58	**0.51**	4
	C	0	0	1
	B	0.41	0.33	2
	EV	0.30	0.13	3
	PageRank	0.56	0.40	5

TABLE 4.1 – Comparaison des résultats de LICOD-TopoCent avec LICOD-{SD, C, B, EV, PageRank} sur les réseaux réels

Le tableau 4.1 montre le résultat de LICOD-TopoCent et avec les autres mesures de centralité, sur les 4 réseaux Zachary, Football, Livres politiques et Dauphins. On peut remarquer que LICOD-TopoCent est plus performant que LICOD-D dans les 4 réseaux étudiés. Avec TopoCent, nous avons obtenu le meilleur score de LICOD (NMI et ARI). Dans les deux réseaux Livres politiques et Dauphins, le score (NMI et ARI) de LICOD-TopoCent est très proche de celui du LICOD-B et parfois meilleur (ARI =51 dans le réseau Dauphins). Dans le réseau Football, les valeurs de NMI et ARI de LICOD-TopoCent sont proches de celles du LICOD avec les mesures globales, mais avec TopoCent, nous avons obtenu le bon nombre de communauté : 11.

La centralité semi-locale SD est plus efficace que TopoCent en terme de NMI et ARI dans le réseau Football. Dans Livres politique et Zachary, LICOD est plus performant avec TopoCent que SD.

4.3.2 Résultats sur les réseaux artificiels

Deux expérimentations ont été effectuées sur des graphes générés par LFR [Lancichinetti *et al.*, 2008]. Les paramètres à fixer dans LFR sont les suivants : nombre de nœuds N, le degré moyen k, le degré maximale $maxk$ et le paramètre de mélange μ. La configuration de ces paramètres dans les deux expérimentations est comme suit :

1. Variation de k : $k = [20, 30, 40, 50, 60]$, $N = 1000$, $maxk = 70$, $\mu = 0.1$
2. Variation de N : $N = [500, 1000, 1500, 2000, 2500]$, $k = 20$,$maxk = 70$, $\mu = 0.1$

Le score de LICOD avec les différentes mesures de centralités, est calculé en terme de NMI et ARI. Chaque score représente la moyenne de 30 graphes LFR ayant la même configuration. Les figures 4.1 et 4.2 montrent l'effet de la variation de k sur la performance de LICOD avec les différentes mesures de centralité. L'effet de la variation de N est affiché également dans les deux figures 4.3 et 4.4. En analysant les quatre courbes, on peut remarquer qu'à part un léger décalage de LICOD-PageRank en terme de ARI (Figure 4.4), LICOD avec TopoCent surpasse en terme de NMI et ARI et LICOD muni des autres mesures de centralité.

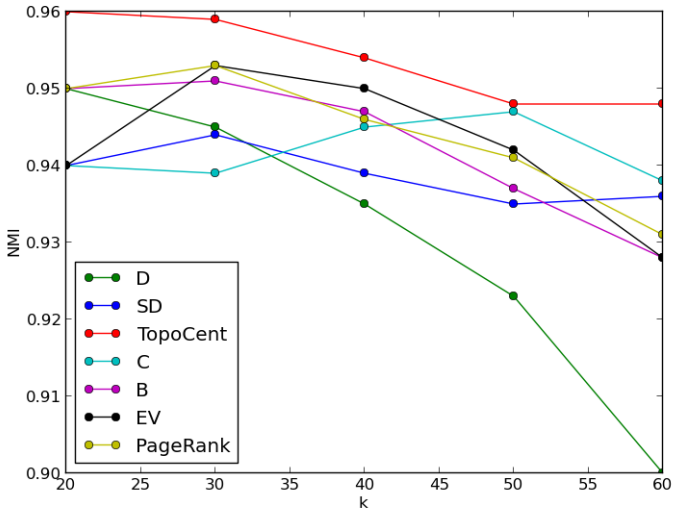


FIGURE 4.1 – Comparaison des résultats de LICOD-TopoCent avec LICOD-{SD, C, B, EV, PageRank} sur LFR (N=1000, $\mu = 0.1$) : NMI vs K

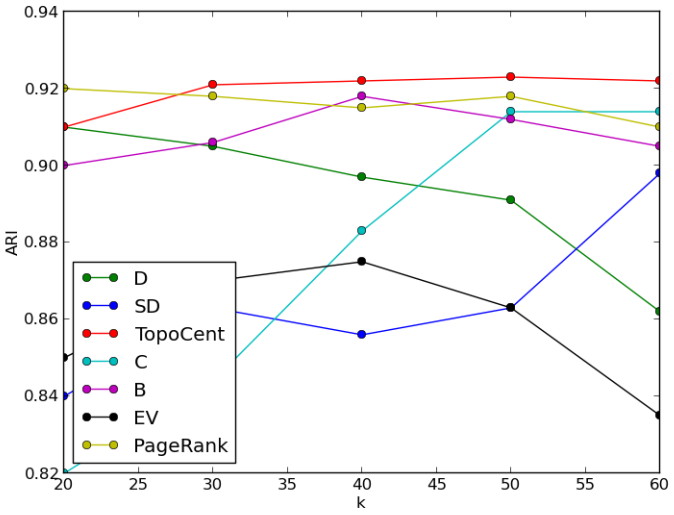


FIGURE 4.2 – Comparaison des résultats de LICOD-TopoCent avec LICOD-{SD, C, B, EV, PageRank} sur LFR (N=1000, $\mu = 0.1$) : ARI vs K

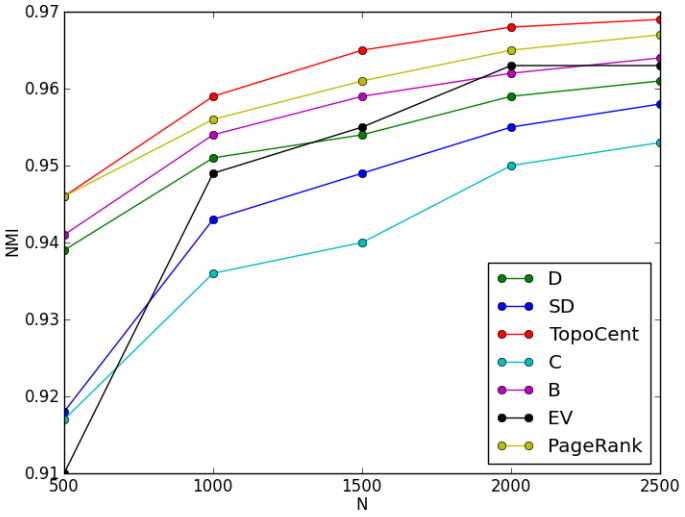


FIGURE 4.3 – Comparaison des résultats de LICOD-TopoCent avec LICOD-{SD, C, B, EV, PageRank} sur LFR (k=20, $\mu = 0.1$) : NMI vs N

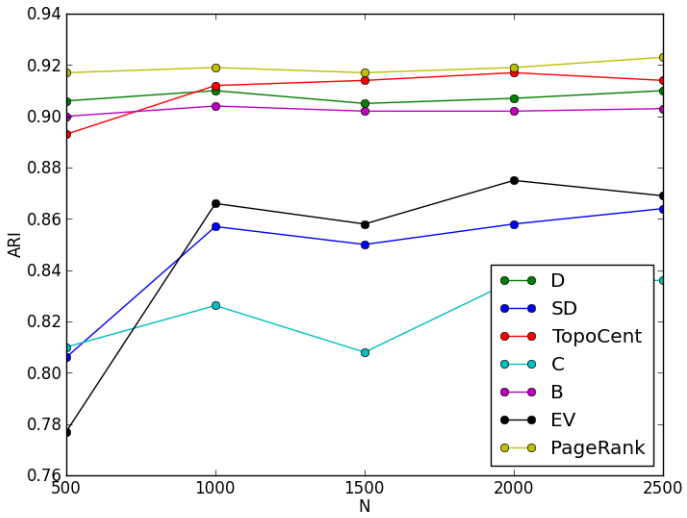


FIGURE 4.4 – Comparaison des résultats de LICOD-TopoCent avec LICOD-{SD, C, B, EV, PageRank} sur LFR (k=20, $\mu = 0.1$) : ARI vs N

Évaluation de TopoCent avec un test de Student

Afin d'étudier plus en détails la performance de LICOD avec TopoCent, nous avons appliqué pour les deux expérimentations 1 et 2, le test de Student [Box *et al.*, 2005] sur le couple de score LICOD-TopoCent et LICOD-{D,SD,C,B,EV,PR}. Nous nous sommes limités aux scores de NMI pour effectuer ce test. Comme affiché dans les deux tableaux 4.2 et 4.3, le test Student est représenté principalement par la p-valeur. Le but de ce test Student est de mesurer si le score moyen diffère significativement entre les échantillons (dans notre cas, les échantillons sont les scores obtenus sur chaque 30 graphes). Dans le cas où p-valeur dépasse 0.05, on ne peut pas rejeter l'hypothèse nulle des scores moyens identiques. Si la p-valeur est plus petite que ce seuil, alors l'hypothèse nulle de l'égalité des moyennes est rejetée. Dans les deux tableaux 4.2 et 4.3, nous présentons les p-valeurs ainsi que la moyenne du score NMI obtenu pour chaque 30 graphes.

Nous remarquons que seulement quand k=20 (tableau 4.2) ou N=500 (tableau 4.3), les p-valeurs avec PR et B (seulement dans le tableau 4.3) sont supérieurs à 0.05. Cela est dû à une faible différence entre les scores de NMI. Pour le reste des deux tableau, la p-valeur est toujours faible. Ainsi, la différence du NMI entre LICOD-TopoCent et LICOD-{D,SD,C,B,EV,PR} est significative.

4.4 Conclusion

Nous avons introduit dans ce chapitre une nouvelle mesure de centralité semi-locale que nous avons appelée TopoCent. Elle a la spécificité de distinguer, parmi les voisins, les

	k=20		k=30		k=40		k=50		k=60	
	NMI	p	NMI	p	NMI	p	NMI	p	NMI	p
D	.950	2.127e-05	.946	1.133e-13	.935	4.217e-16	.924	3.228e-15	.900	1.046e-17
SD	.941	6.728e-09	.945	1.360e-07	.939	3.69e-06	.935	2.989e-05	.936	0.290
C	.936	2.742e-10	.940	1.272e-06	.947	0.23e-03	.946	0.387	.935	14e-03
B	.954	0.003	.951	1.152e-06	.947	7.523e-06	.937	1.073e-08	.928	1.218e-09
EV	.947	4.338e-05	.953	0.064	.950	0.118	.942	0.072	.928	0.020
PR	.958	0.375	.953	8.020e-08	.947	3.110e-08	.941	1.177e-07	.931	4.139e-09
TopoCent	.960		.959		.955		.948		.938	

TABLE 4.2 – Test Student du score NMI de LICOD-TopoCent vs NMI de LICOD-{D,SD,C,B,EV,PR} sur LFR (N=1000, k=[20,30,40,50,60], $\mu = 0.1$)

	N=500		N=1000		N=1500		N=2000		N=2500	
	NMI	p	NMI	p	NMI	p	NMI	p	NMI	p
D	.939	0.015	.951	1.095e-06	.955	1.830e-10	.959	1.948e-11	.961	2.433e-13
SD	.918	6.869e-08	.943	1.980e-07	.949	1.400e-08	.955	5.524e-12	.958	1.375e-12
C	.917	5.987e-07	.936	7.446e-10	.940	2.253e-12	.950	1.534e-12	.953	3.881e-12
B	.941	0.163	.954	0.350e-03	.959	2.128e-05	.962	2.810e-10	.964	8.943e-08
EV	.910	4.700e-06	.949	0.783e-03	.955	2.741e-06	.963	0.003	.963	0.430 e-03
PR	.945	0.973	.956	0.020	.961	0.002	.965	0.315e-03	.967	0.12e-03
TopoCent	.946		.959		.965		.968		.969	

TABLE 4.3 – Test Student du score NMI de LICOD-TopoCent vs NMI de LICOD-{D,SD,C,B,EV,PR} sur LFR (N=[500,1000,1500,2000,2500], k=20, $\mu = 0.1$)

plus centraux des moins centraux. Le calcul de centralité s'arrête au deuxième niveau des voisins. Les bons résultats de LICOD avec TopoCent ont montré que cette centralité couvre suffisamment de nœuds pour qualifier de bons leaders pour les communautés. Cependant, il faut étudier plus en profondeur comment déterminer automatiquement le bon niveau de voisins pour TopoCent.

Chapitre 5

Évaluation des algorithmes de détection de communautés : une méthode orientée tâche

Sommaire

5.1 Introduction

Bien qu'il y ait eu de nombreuses méthodes proposées pour la détection de communautés, très peu de recherches ont été menées pour explorer l'évaluation et la validation de ces méthodes. Comme mentionné dans le deuxième chapitre, les méthodes d'évaluation sont de trois types : (1) évaluation orientée-communautés, (2) évaluation orientée-partition, et (3) évaluation par rapport à la vérité de terrain. Après l'exploration des limites de la modularité (voir chapitre 1), l'évaluation de la performance des algorithmes par rapport à la vérité de terrain est devenue la pratique la plus courante. Néanmoins, comme on l'a expliqué dans le chapitre 2, il existe peu de benchmarks dont on connait la vérité de terrain, tandis que les générateurs de benchmarks synthétiques utilisés actuellement comme LFR, ne garantissent pas la couverture de toutes les caractéristiques des réseaux réels. Ces faits incitent à explorer d'autres types d'approches d'évaluation.

Dans ce chapitre, nous proposons un autre type d'évaluation des algorithmes de détection de communautés : l'évaluation orientée-tâche. L'objectif est de confronter l'algorithme à une tâche, autre que la détection de communautés, pour laquelle on dispose des outils d'évaluation. Nous nous concentrons dans cette thèse sur deux tâches :

— Tâche 1 : classification non-supervisée de donnés

— Tâche 2 : prévision des liens dans les graphes dynamiques

Dans la suite de ce chapitre, nous rappelons le mécanisme de chacune de ces tâches, puis nous expliquons le mode d'emploi de ces deux tâches pour l'évaluation des algorithmes de détection de communautés, et nous présentons les données utilisées ainsi que les résultats obtenus.

5.2 CLE : Evaluation orientée CLassification non-supervisée

La détection de communautés est une tâche similaire à la tâche de classification non-supervisée de données. Les deux tâches cherchent à identifier des groupes d'individus homogènes. Elles se ressemblent encore plus quand les liens du graphe expriment des similarités entre les nœuds. Dans ce cas, le concept de communauté se confond avec celui du cluster et les méthodes de détection de communautés deviennent applicables à la classification non-supervisée.

Étant donné que dans le domaine de classification non-supervisée, on dispose non seulement des bonnes mesures d'évaluation mais aussi d'une variété de benchmarks avec des vérité de terrain connues, nous proposons une approche, que nous appelons CLE, qui évalue les algorithmes de détection de communautés sur une tâche de classification non-supervisée.

5.2.1 Présentation générale de l'approche

L'idée principale est de transformer le problème de partitionnement de données en un problème de détection de communautés. La figure 5.1 représente les principales étapes de notre approche. Premièrement, nous calculons la matrice de similarité entre chaque couple

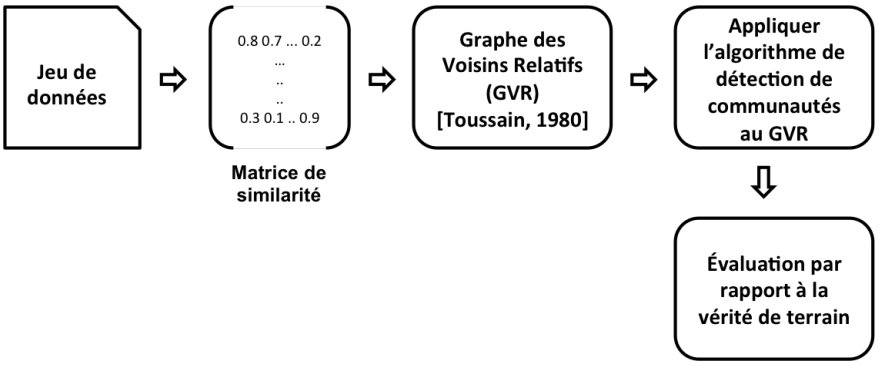


FIGURE 5.1 – L'approche CLE

d'individus du jeu de données. Cette matrice aura une taille $n \times n$ avec n le nombre d'individus dans le jeu de données. Deuxièmement, nous construisons à partir de la matrice de similarité, *le Graphe des Voisins Relatifs* proposé dans [Toussaint, 1980]. Bien évidement, les caractéristiques topologiques des graphes obtenus dépendent de la fonction de similarité utilisée dans la première étape. Troisièmement, nous appliquons l'algorithme de détection de communautés que nous voulons évaluer. Et enfin, nous comparons la performance des algorithmes en fonction de l'information mutuelle normalisée (NMI) et l'indice de Rand

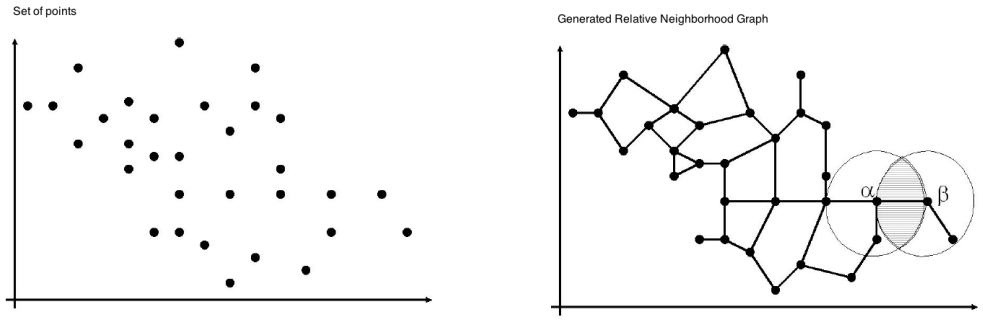
ajusté (ARI).

Les graphes des voisins relatifs

Un Graphe des Voisins Relatifs (GVR) est un graphe connecté symétrique. La règle de construction est simple : deux points x_i et x_j sont connectés par un lien si et seulement si :

$$d(x_i, x_j) \leq \max_l \{d(x_i, x_l), d(x_j, x_l)\}, \forall l \neq i, j \tag{5.1}$$

avec $d(x_i, x_j)$ est une fonction de distance.
La figure 5.2 montre un exemple de construction d'un GVR à partir d'un ensemble de données.



(a) Nuage de données (b) GVR généré

FIGURE 5.2 – Un exemple de génération d'un GVR à partir d'un ensemble de données : α et β sont deux voisins relatifs parce qu'il n'y pas d'autre nœud dans l'intersection des deux cercles centrées respectivement en α et β et de rayon $d(\alpha, \beta)$

Nous avons construit pour chaque jeu de données quatre GVRs en utilisant les matrices de distance générées par ces quatre fonction de distance :

Distance	formule
Euclidienne	$dist_{euc}(x,y) = \sqrt{\sum_{i=1}^{n} \lvert x_i - y_i \rvert^2}$
Chebyshev	$dist_{cheb}(x,y) = \max_i(x_i - y_i)$
Cosinus	$dist_{cos}(x,y) = 1 - \frac{x.y}{\lvert x \rvert \lvert y \rvert}$
Corrélation	$dist_{cor}(x,y) = 1 - \frac{\lvert (x-\bar{x}) \rvert \lvert (y-\bar{y}) \rvert}{\lvert (x-\bar{x}) \rvert \lvert (y-\bar{y}) \rvert}$

TABLE 5.1 – Les fonction de distance utilisées pour la génération des GVRs

5.2.2 Expérimentation

Nous avons testé notre approche sur quatre jeux de données téléchargés du site web UCI[1]. Nous présentons leurs principales caractéristiques dans le tableau 5.2.

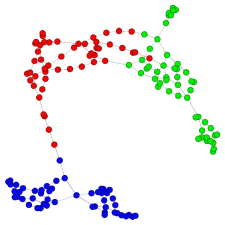
1. http ://archive.ics.uci.edu/ml/datasets.html

Jeux de données	Iris	Glass	Wine	Vehicle	Abalone
#Instances	150	214	178	846	772
#Attributs	4	10	13	18	8
# Classes	3	7	3	4	29

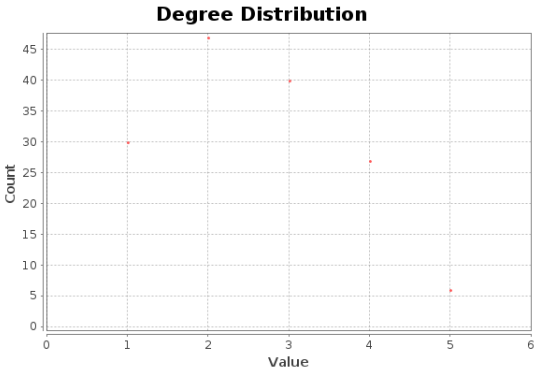
TABLE 5.2 – Caractéristique des jeux de données

Pour chaque jeu de données, nous avons généré quatre GVR en variant à chaque fois la distance. Le tableau 5.3 décrit les caractéristiques topologiques de ces graphes. À part le jeu de donnée Wine, les graphes à base de la distance chebyshev sont les plus denses. Ils possèdent également le plus haut coefficient de clustering (transitivité). Les autres distances donnent des graphes creux, comme dans les réseaux réels, avec des densités qui ne dépassent pas 0.035. En revanche, les valeurs de γ sont relativement élevées par rapport à celles des réseaux réels, et la distribution de degré est loin d'être une loi de puissance.

Dans la suite, nous nous limitons aux GVR générés en fonction de la distance euclidienne pour comparer la performance des algorithmes de détection de communautés. Nous présentons en particulier leurs structures communautaires (chaque couleur indique une communauté) et leurs distributions de degré dans les figures 5.3, 5.4, 5.5 ,5.6 et 5.7. Les figures des GVRs produites à partir des autres distances sont affichées dans l'annexe A.



(a) Structure du graphe



(b) Distribution de degré

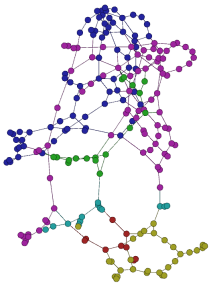
FIGURE 5.3 – Iris : GVR généré en fonction de la distance euclidienne

L'application des algorithmes de détection de communautés GN, Walktrap, EV, Info-Map, FastGreedy, Louvain, LICOD et it-LICOD a donné les résultats présentés dans le tableau 5.4. Les résultats sont en termes de NMI, ARI, la modularité Q et le nombre de communautés (*#com*). Le but de calculer Q est de montrer ses résultats contradictoires par rapport à ceux fournis par NMI et ARI.

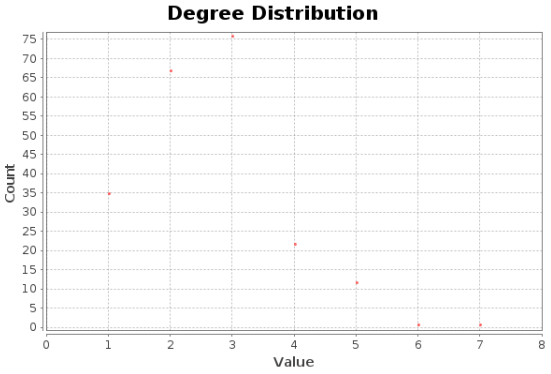
Pour faciliter l'interprétation des résultats, nous avons calculé également pour chaque score le classement des algorithmes (rang). En analysant ces classements, nous remarquons qu'ils varient d'un graphe à un autre. Cependant, si nous nous concentrons sur les valeurs de NMI, il y des fortes ressemblances entre le classement obtenu sur Iris et Wine, et entre

Dataset	Caractéristiques	Euclidienne	Chebyshev	Cosinus	Corrélation
Iris	#Nœuds		150		
	#Liens	382	2468	426	398
	Diamètre	33	14	25	26
	γ loi de puissance	4.51	8.92	4.73	5.68
	Densité	0.034	0.220	0.038	0.035
	Transitivité	0.055	0.340	0.011	0.012
Glass	#Nœuds		214		
	#Liens	558	7786	552	530
	Diamètre	21	8	24	22
	γ loi de puissance	4.07	4.49	10.35	9.49
	Densité	0.024	0.341	0.024	0.023
	Transitivité	0.0139	0.252	0.011	0.010
Wine	#Nœuds		178		
	#Liens	380	514	438	450
	Diamètre	102	84	59	56
	γ loi de puissance	10.31	12.63	7.79	5.09
	Densité	0.024	0.032	0.027	0.028
	Transitivité	0	0.178	0	0
Vehicle	#Nœuds		846		
	#Liens	2598	4072	2764	2794
	Diamètre	63	54	45	37
	γ loi de puissance	11.28	9.46	11.93	10.81
	Densité	0.007	0.011	0.007	0.007
	Transitivité	0.002	0.091	0	0
Abalone	#Nœuds		772		
	#Liens	2542	89338	2158	2038
	Diamètre	38	22	50	83
	γ loi de puissance	11.47	9.50	7.53	8.00
	Densité	0.008	0.30	0.007	0.006
	Transitivité	0	0.49	0	0

TABLE 5.3 – Caractéristiques topologiques des graphes de voisinage

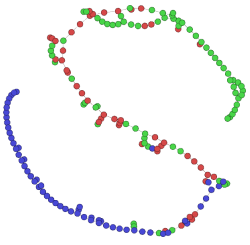


(a) Structure du graphe

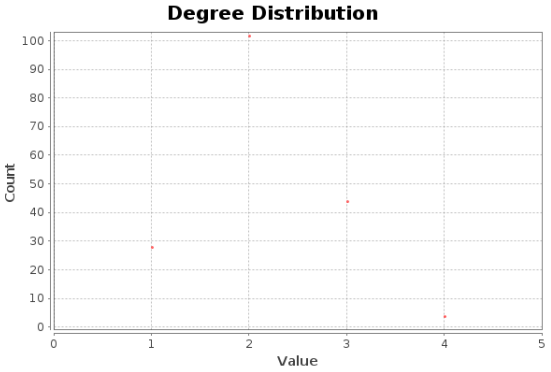


(b) Distribution de degré

FIGURE 5.4 – Glass : GVR généré en fonction de la distance euclidienne

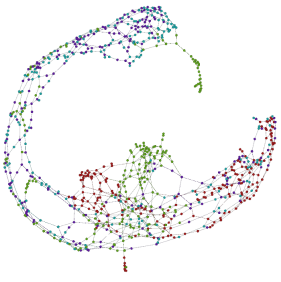


(a) Structure du graphe

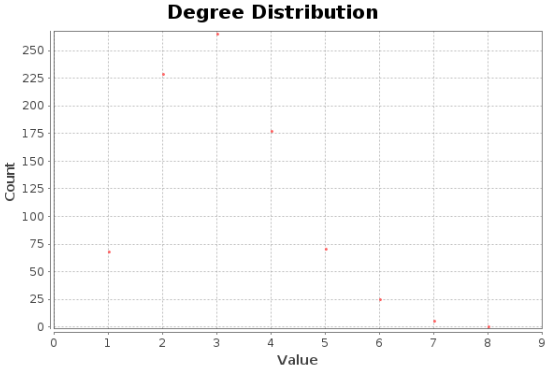


(b) Distribution de degré

FIGURE 5.5 – Wine : GVR généré en fonction de la distance euclidienne

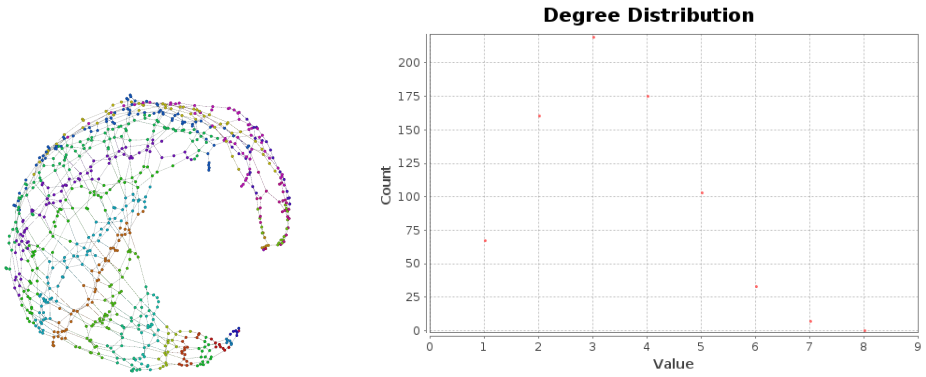


(a) Structure du graphe



(b) Distribution de degré

FIGURE 5.6 – Vehicle : GVR généré en fonction de la distance euclidienne



(a) Structure du graphe

(b) Distribution de degré

FIGURE 5.7 – Abalone : GVR généré en fonction de la distance euclidienne

celui de Glass, Vehicle et Abalone. Dans le premier groupe, on trouve en tête du classement LICOD, Walktrap, Louvain et EV. En bas du classement, it-LICOD, GN, FastGreedy donnent toujours des bons résultats par rapport à l'algorithme InfoMap. Dans le deuxième groupe, et surtout dans Glass et Vehicle, InfoMap et FastGreedy sont les plus performants, suivis par it-LICOD et Louvain. Dans Abalone, InfoMap est aussi classé premier suivi par Louvain, it-LICOD et Walktrap. Cette différence de classement des algorithmes entre les deux premier réseaux (celui de Iris et Wine) et les trois derniers (réseaux de Glass, Vehicle et Abalone) est dû à la nature de la distribution des nœuds des communautés réelles. On peut voir dans les figures 5.3(a), et 5.5(a) que les communautés sont bien dégagées, ce qui n'est pas le cas pour les 3 derniers réseaux (voir figures 5.4(a), 5.5(a) et 5.7(a)). La dispersion des nœuds de la même communauté sur l'ensemble du réseau permet d'avoir plus de petites communautés, et c'est pour cette raison que les algorithmes qui détectent beaucoup de communautés, comme it-LICOD et InfoMap, ont eu des bons résultats.

L'application des algorithmes sur ce genre de réseaux a permis de découvrir quelques propriétés des algorithmes que nous n'avons pas découvertes ni dans les réseaux réels et ni dans les réseaux artificiels. Dans la suite, nous étudions de nouvelles caractéristiques des algorithmes de détection de communautés en les confrontant à une autre tâche.

5.3 PLE : Evaluation orientée Prévision de Liens

5.3.1 Présentation générale de l'approche

Le problème de prévision de nouveaux liens est un problème classique. Il peut être formulé comme suit : étant donné une suite temporelle $< G_0, G_1, ..., G_t >$ qui décrit l'évolution d'un graphe G dans l'intervalle de temps [0,t], le problème de prévision de nouveaux liens consiste à construire un modèle qui peut prévoir l'apparition de liens à l'instant $t+1$ entre les nœuds existants mais qui n'ont été jamais reliés entre eux auparavant.

Les approches classiques de ce problème calculent la similarité topologique de deux

Jeux de données	Algo	NMI	$Rang_{NMI}$	ARI	$Rang_{ARI}$	Q	$Rang_Q$	#com
Iris	GN	0.64	2	0.42	6	079	1	10
	Walktrap	0.68	1	0.48	2	0.78	2	9
	EV	0.63	4	0.43	4	0.73	5	9
	InfoMap	0.51	8	0.16	7	0.73	5	22
	FastGreedy	0.61	6	0.42	6	0.76	4	9
	Louvain	0.63	4	0.43	4	0.77	8	9
	LICOD	0.64	2	0.53	1	0.69	7	11
	it-LICOD	0.58	7	0.48	2	0.65	8	18
Wine	GN	0.31	5	0.11	6	0.84	2	15
	Walktrap	0.33	2	0.14	2	0.84	2	11
	EV	0.33	2	0.14	2	0.84	2	12
	InfoMap	0.29	8	0.05	8	0.81	6	26
	FastGreedy	0.31	5	0.13	4	0.85	1	13
	Louvain	0.33	2	0.13	4	0.84	2	12
	LICOD	0.35	1	0.34	1	0.75	7	14
	it-LICOD	0.31	5	0.07	7	0.73	8	34
Glass	GN	0.46	5	0.16	3	0.74	1	14
	Walktrap	0.46	5	0.13	6	0.70	4	22
	EV	0.45	7	0.16	3	0.69	6	17
	InfoMap	0.47	2	0.12	8	0.70	4	30
	FastGreedy	0.50	1	0.28	1	0.73	3	10
	Louvain	0.47	2	0.22	2	0.74	1	11
	LICOD	0.43	8	0.15	5	0.67	7	16
	it-LICOD	0.47	2	0.13	6	0.67	7	23
Vehicle	GN	0.21	5	0.09	2	0.82	1	17
	Walktrap	0.21	5	0.08	4	0.80	4	25
	EV	0.21	5	0.08	4	0.79	5	19
	InfoMap	0.25	1	0.03	7	0.73	6	75
	FastGreedy	0.24	2	0.11	1	0.81	2	15
	Louvain	0.22	4	0.09	2	0.81	2	16
	LICOD	0.21	5	0.05	6	0.68	7	42
	it-LICOD	0.23	3	0.02	8	0.60	8	96
Abalone	GN	0.49	7	0.19	5	0.78	1	14
	Walktrap	0.54	2	0.22	2	0.75	4	24
	EV	0.48	8	0.20	3	0.75	4	10
	InfoMap	0.60	1	0.15	6	0.70	6	69
	FastGreedy	0.51	5	0.20	3	0.76	3	14
	Louvain	0.54	2	0.23	1	0.78	1	13
	LICOD	0.51	5	0.15	6	0.64	7	35
	it-LICOD	0.53	4	0.13	8	0.60	8	61

TABLE 5.4 – Comparaison des résultats des algorithmes sur les GVRs générés en fonction de la distance euclidienne

nœuds non connectés, et concluent que les nœuds très similaires sont les plus susceptibles d'être reliés à l'instant $t+1$. Récemment, une approche de prévision de liens a été introduit dans [Soundarajan et Hopcroft, 2012], qui a montré que si on inclut l'appartenance communautaire des nœuds dans les mesures de similarités topologiques, la précision du modèle prédictif augmente.

Dans cette section, nous proposons une approche d'évaluation orientée-prévision de liens, appelée PLE, qui s'appuie sur ce travail. Notre objectif est d'évaluer les algorithmes de détection de communautés en fonction de leurs contributions dans la tâche de prévision de liens (voir figure 5.8) .

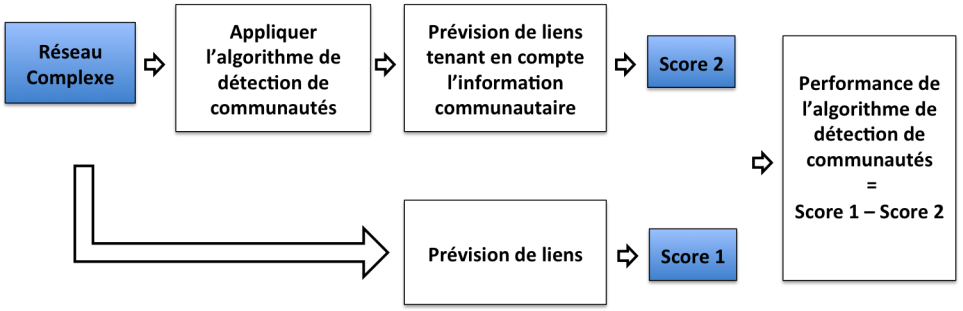


FIGURE 5.8 – L'approche PLE

L'approche de prévision de lien adoptée est une approche non supervisée. Etant donné un graphe $G = (E, V)$ non-dirigé avec, le pseudo-code de l'approche est le suivant :

1. Extraire les couples candidats à partir du G_t. Soit $Candidats = \{cd_1, cd_2, ..., cd_j\}$, avec $cd_i = (u, v)$ est un couple de nœuds.
2. Calculer le score topologique entre chaque couple de nœuds en tenant compte de l'information communautaire. On notera la fonction qui calcule ce score par S.
3. Trier les couples selon leurs scores, et retenir le top k de ces couples, k est égal au nombre de nouveaux liens apparus dans G_{t+1}.
4. Calculer la précision P des top k couples choisis. P est définit comme suit :

$$P = \frac{\text{Nombre de couples positifs}}{k} \tag{5.2}$$

Notre approche se base sur les mesures de similarité topologiques suivantes : le nombre de voisins communs (CN), le coefficient de Jaccard, Adamic Adar (AA) et le plus court chemin (SP).

Afin d'introduire l'information communautaire dans le problème de prévision de liens, nous proposons de modifier ces mesures de similarité. Soit $G = (E, V)$ un graphe non-dirigé avec :

— $\Gamma(u)$ l'ensemble des voisins de u

— $\Gamma(u, v)$ l'ensemble des voisins communs de u et v.

— Soit $P = \{ c_1, c_2, ..., c_m\}$ l'ensemble des communautés de G. $c(u)$ est la communauté de u. $c(u, v)$ est la communauté de u et v.

— $\Gamma'(u,v)$ l'ensemble des voisins communs de u et v appartenant à la même communauté

La définition de ces mesures de similarité ainsi que de leurs versions prenant en compte l'aspect communautés sont présentées dans le tableau 5.5

Mesure de base	Mesure introduisant l'information de la communauté
$CN(u,v) = \lvert\Gamma(u,v)\rvert$	$CN'(u,v) = \lvert\Gamma'(u,v)\rvert$
$JAC(u,v) = \frac{\lvert\Gamma(u,v)\rvert}{\Gamma(u)\cap\Gamma(v)}$	$JAC'(u,v) = \frac{\lvert\Gamma'(u,v)\rvert}{\Gamma(u)\cup\Gamma(v)}$
$AA(u,v) = \sum_{z\in\Gamma(u,v)} \frac{1}{log\lvert\Gamma(z)\rvert}$	$AA'(u,v) = \sum_{z\in\Gamma'(u,v)} \frac{1}{log\lvert\Gamma(z)\rvert}$
$SP(u,v) = \frac{1}{ShortestPath(u,v)+1}$	$SP'(u,v) = \frac{1}{\lvert z\in ShortestPath(u,v), c(z)=c(u,v)\rvert+1}$

TABLE 5.5 – Les mesures de similarité topologiques utilisées pour le calcul de score entre les couples de nœuds

5.3.2 Expérimentation

Description des réseaux

L'expérimentation de notre approche a été faite sur des donnés bibliographiques de DBLP `http://dblp.dagstuhl.de`. Nous nous sommes concentrés sur les graphes de co-publications : deux auteurs sont connectés s'ils ont co-publié ensemble. Nous allons effectué 4 expérimentations sur 4 périodes différentes :

1. La période 1972-1977 : G_t=1972-1975, G_{t+1}=1975-1977
2. La période 1974-1979 : G_t=1974-1977, G_{t+1}=1977-1979
3. La période 1980-1985 : G_t=1980-1983, G_{t+1}=1983-1985
4. La période 1982-1987 : G_t=1982-1985, G_{t+1}=1985-1987

Pour chacune de ces périodes, nous avons utilisé seulement la plus grande composante connexe du graphe. Nous affichons leurs caractéristiques principales dans le tableau 5.3.2. Les réseaux des deux premières périodes sont de petites tailles et ils ont des densités légèrement supérieures par rapport aux deux derniers réseaux.

Résultats

Pour chaque période, nous présentons ci-dessous dans le même tableau les précisions obtenues par les différentes mesures de similarité topologique et celles obtenues avec notre approche PLE, là où on introduit l'information communautaire par les différentes algorithmes étudiés.

Dans le tableau 5.7, on remarque que les scores de précision des mesures topologiques CN, JAC, AA et SP varient d'un graphe à un autre. En effet, dans ce genre de tâche, le score de précision ne dépend pas seulement des caractéristiques globales du graphe, mais aussi des caractéristiques du sous-graphe connectant les deux couples candidats du graphe G_t. Ainsi, il est très difficile d'étudier ces sous-graphes cas par cas pour tous les couples de nœuds.

Période	**Caractéristiques**	**Valeur**
1972-1975	Nœuds	221
	Liens	319
	Densité	0.013122
1974-1977	Nœuds	323
	Liens	451
	Densité	0.008673
1980-1983	Nœuds	1371
	Liens	2463
	Densité	0.002623
1982-1985	Nœuds	2146
	Liens	3749
	Densité	0.001629

TABLE 5.6 – Caractéristiques des graphes de co-publication

Période	k	CN	JAC	AA	SP
G_t=1972-1975, G_{t+1}=1975-1977	49	0.0816	0.0408	0.0612	0.0612
G_t=1974-1977, G_{t+1}=1977-1979	93	0.0215	0.0215	0.0215	0.0323
G_t=1982-1985, G_{t+1}=1985-1987	426	0.0164	0.0305	0.0281	0.0234
G_t=1982-1985, G_{t+1}=1985-1987	733	0.0231	0.0259	0.0272	0.0150

TABLE 5.7 – Résultats obtenus sur les quatre périodes avec les différentes mesures topologiques sans utilisation de l'information communautaire

Avant d'analyser la contribution des algorithmes de détection de communautés dans les scores de précision, nous tenons à mentionner que la précision P de S' peut être :

— dégradée $P(S') < P(S)$: la dégradation de P(S') par rapport à P(S) peut avoir lieu quand deux nœuds candidats ayant la même communauté, mais topologiquement éloignés, reçoivent une récompense sous forme d'information communautaire, et ainsi bien classés par rapport à d'autres couples n'appartenant pas à la même communauté mais ils sont topologiquement proches.

— inchangée $P(S') = P(S)$: dans le cas où les deux nœuds candidats appartiennent à deux communautés différentes, la précision est égale à celle calculée sans l'information communautaire.

— améliorée $P(S') > P(S)$: cela peut avoir lieu quand les deux nœuds candidats partagent la même communauté et sont suffisamment proches à l'intérieur de celle-ci. C'est le cas qui évalue plus l'algorithme de détection de communauté. La différence $P(S') - P(S)$ mesure à quel point la communauté est de bonne qualité.

Afin de faciliter la lecture des résultats, nous avons coloré les valeurs de P(S') appartenant aux premier cas par la couleur rouge, tandis que celles du troisième cas on été mis en gras. Le deuxième cas est en style normal.

À première vue, les tableaux 5.8, 5.9, 5.10 et 5.11 montrent que les scores de précision varient d'une mesure topologique à une autre, et d'une période à une autre. Ce qui rend l'interprétation des résultats encore plus difficile est le changement de performance des algorithmes d'une période à une autre.

Toutefois, on peut tirer quelques observations :

Les deux mesures CN' et SP' aident plus à évaluer les algorithmes : pour l'ensemble des 4 périodes, les deux mesures qui ont évalué le mieux la contribution des algorithmes à la tâche de prévision des liens sont CN' et SP', car au cours des deux premières périodes (voir tableaux 5.8 et 5.9), les scores trouvés avec AA' ne montrent ni dégradation, ni amélioration. On observe le même cas pour JAC' dans la deuxième période (voir tableau 5.9).

FastGreedy est plus performant avec CN' : à part la première période, les scores obtenus par FastGreedy-CN' sont toujours positifs et parfois les meilleurs (voir tableaux 5.9 et 5.10).

Performance instable de l'algorithme EV : algré l'obtention de quelques bon résultats sur l'ensemble des 4 périodes, la performance de l'algorithme EV varie d'un manière très remarquable, jusqu'à qu'elle chute à 0 avec SP' dans la troisième période (voir tableau 5.10).

Louvain est plus performant avec les mesures topologiques basés sur le voisinage : en analysant les scores de précisions obtenus par Louvain, on voit clairement qu'il n'a eu de bons résultats qu'avec CN' ou JAC'.

it-LICOD est LICOD sont plus efficaces dans les réseaux à très faibles densités : les algorithmes basés sur les leaders, it-LICOD est LICOD, ont donné des résultats positifs seulement dans la dernière période (voir tableau 5.11), qui correspond au réseau qui a la densité la plus faible par rapport aux autres. Ces résultats sont obtenus par CN', JAC' ou SP'.

Dans l'ensemble, la performance de chaque algorithme ne dépend pas seulement des caractéristiques du réseau mais aussi de la performance de la mesure topologique utilisée. Ci-dessous, nous avons montré quelques relations particulières entre algorithme et mesure topologique et algorithme et caractéristique topologique. Cela pourrait être plus défendu avec d'autres réseaux dynamiques.

	CN'	JAC'	AA'	SP'
GN	0.0816	0.0408	0.0612	**0.0816**
Walktrap	0.0816	0.0408	0.0612	0.0612
EV	0.0816	**0.0612**	0.0612	0.0408
InfoMap	**0.1020**	0.0408	0.0612	0.0612
FastGreedy	0.0816	0.0408	0.0612	**0.0816**
Louvain	0.0816	**0.0612**	0.0612	0.0612
LICOD	0.0408	0.0204	0.0612	0.0408
it-LICOD	0.0612	0.0204	0.0612	0.0204

TABLE 5.8 – Résultats de l'approche PLE sur la période : G_t=1972-1975, G_{t+1}=1975-1977

	CN'	JAC'	AA'	SP'
GN	0.0215	0.0215	0.0215	0.0215
Walktrap	**0.0322**	0.0215	0.0215	0.0322
EV	**0.0322**	0.0215	0.0215	**0.0430**
InfoMap	0.0215	0.0215	0.0215	0.0215
FastGreedy	**0.0322**	0.0215	0.0215	0.0323
Louvain	**0.0322**	0.0215	0.0215	0.0323
LICOD	0.0215	0.0215	0.0215	0.0215
it-LICOD	0.0215	0.0215	0.0215	0.0215

TABLE 5.9 – Résultats de l'approche PLE sur la période : G_t=1974-1977, G_{t+1}=1977-1979

	CN'	JAC'	AA'	SP'
GN	**0.0117**	0.0305	0.0258	**0.0305**
Walktrap	0.0164	0.0258	0.0281	0.0117
EV	0.0164	0.0258	0.0281	0
InfoMap	0.0164	0.0305	0.0281	0.0211
FastGreedy	**0.0187**	0.0305	0.0258	0.0234
Louvain	0.0164	**0.0328**	0.0281	0.0164
LICOD	0.0164	0.0234	0.0234	0.0234
it-LICOD	0.0164	0.0281	0.0211	**0.0281**

TABLE 5.10 – Résultats de l'approche PLE sur la période : G_t=1980-1983, G_{t+1}1983-1985

	CN'	JAC'	AA'	SP'
GN	0.0218	0.0245	0.0272	**0.0218**
Walktrap	0.0218	0.0218	0.0259	**0.0163**
EV	**0.0245**	0.0218	0.0272	0
InfoMap	**0.0245**	**0.0272**	0.0272	**0.0231**
FastGreedy	**0.0245**	0.0259	0.0272	**0.0163**
Louvain	0.0231	0.0259	0.0259	0.0136
LICOD	**0.0259**	**0.0272**	0.0259	**0.0190**
it-LICOD	0.0218	**0.0272**	0.0259	**0.0177**

TABLE 5.11 – Résultats de l'approche PLE sur la période : G_t=1982-1985, G_{t+1}=1985-1987

5.4 Conclusion

Dans ce chapitre, nous avons proposé une nouvelle méthodologie pour l'évaluation de la performance des algorithmes de détection de communautés. À travers les deux tâches : classification non-supervisée des données et prévision de liens dans les réseaux dynamiques, nous avons étudié la performance de 8 algorithmes de détection de communautés appartenants à différentes familles d'approches. Dans chaque tâche et dans chaque jeu de données, nous avons exposé de nouveaux comportements des différentes algorithmes, ce qui, sans doute, permet d'enrichir le schéma d'évaluation général des algorithmes de détection de communautés.

À court terme, nous envisageons d'appliquer les deux approches CLE et PLE sur d'autres benchmarks et d'évaluer plus d'algorithmes de détection de communautés. Pour le long terme, nous avons l'objectif d'étudier la tâche de visualisation de la structure communautaire comme une autre tâche d'évaluation.

Chapitre 6

Conclusion et perspectives

Nous résumons à présent les contributions apportées par ces travaux de thèse, puis nous proposons quelques perspectives.

Synthèses et contributions

Dans cette thèse, nous avons abordé trois problématiques : la multitude des mesures utilisées dans les algorithmes de détection de communautés guidées par l'identification de leaders, le problème de la non efficacité des mesures de centralité locales et le temps coûteux des mesures globales, et le problème d'évaluation des algorithmes de détection de communautés : l'évaluation sur des réseaux n'ayant pas la vérité de terrain est sanctionnée par la non significativité des mesures topologiques telle que la modularité. De même, l'évaluation sur des réseaux possédant une vérité de terrain est limitée par le petit nombre de ces réseaux, et par les différents inconvénients des générateurs artificiels. Pour y parvenir, nous nous sommes basés sur les méthodes suivantes.

Dans un premier temps, nous avons proposé un enrichissement de l'approche LICOD en intégrant d'autres mesures applicables dans ses différentes étapes. La comparaison des différentes stratégies de la configuration de LICOD sur les réseaux réels et artificiels, a donné un aperçu sur la performance de chaque mesure utilisée. Ensuite, nous avons proposé également une extension de LICOD, appelée it-LICOD. Dans cette extension, nous avons ajouté une étape itérative qui permet de valider le choix des bons leaders de chaque communauté. Les résultats trouvés sur les réseaux réels ont montré des améliorations de performance par rapport à ceux de LICOD. En outre, it-LICOD est plus performant sur les réseaux artificiels pour différentes configurations du générateur de graphe LFR.

Dans un deuxième temps, nous avons proposé une mesure de centralité semi-locale, appelée TopoCent. Cette mesure couvre plus de nœuds que les mesures locales sans atteindre tous les nœuds du réseau. Nous avons également mis en évidence l'aspect topologique en distinguant les voisins les plus importants des moins importants. L'expérimentation de TopoCent dans le cadre de détection de communautés avec LICOD sur différents types de réseaux, a montré que LICOD-TopoCent est toujours plus performant que LICOD-autre centralité.

Dans un troisième temps, nous avons introduit deux méthodes orientées-tâche pour l'évaluation des algorithmes de détection de communautés. La première méthode est orientée-classification non-supervisée de données, notée CLE. Cette méthode transforme un jeu de données numériques en un graphe de voisinage via une mesure de similarité, afin de confronter les algorithmes de détection de communautés au problème de classification non-supervisée. Cette méthode n'a pas seulement donné un nouveau cadre d'évaluation

de ces algorithmes, mais elle a aussi donné la possibilité de produire de nouveaux benchmarks ayant des vérités de terrain. Les résultats trouvés via cette méthode ont montré d'autres niveaux de performance des algorithmes étudiés. Dans la deuxième méthode que nous avons noté PLE, nous avons choisi comme tâche la prévision de liens non-supervisée. Cette méthode permet d'évaluer les algorithmes selon leurs contributions dans la tâche de prévision de nouveaux liens. Les résultats expérimentaux sur des graphes bibliographiques dynamiques ont montré d'autres performances de certains algorithmes. Ces résultats encourageants devraient être défendus par des expérimentations sur d'autres réseaux. Les deux méthodes orientées-tâche proposées représentent une nouvelle famille d'approches d'évaluation, qui va certainement enrichir le schéma d'évaluation général des algorithmes de détection de communautés.

Perspectives

Cette thèse ouvre différentes perspectives de travail.

Les premières perspectives concernent la modification de l'approche it-LICOD afin de prendre en compte les communautés chevauchantes. Plus précisément, les méthodes de fusion de votes devraient subir une modification pour permettre le multiple appartenance communautaire des nœuds.

Nous envisageons également le développement d'une extension hiérarchique de LICOD qui s'adapte plus aux réseaux contenant une hiérarchie de communautés. L'idée est d'appliquer LICOD sur chaque communauté de chaque niveau hiérarchique, et de s'arrêter quand aucune communauté n'est décomposable. Toutefois, vu les limites de la modularité présentées dans le chapitre 1 de ce manuscrit, il faut trouver une autre heuristique pour choisir la meilleur partition dans la hiérarchie.

Pour finir avec les perspectives sur la détection de communautés, nous envisageons d'exploiter les méthodes de fusion de votes pour proposer une méthode globale pour la détection de communautés. L'idée est de fusionner des partitions issues de différents algorithmes, ou d'un algorithme instable, en utilisant les méthodes de fusion de votes. Cela peut être utile pour la détection de communautés dans les réseaux multiplexes, où plusieurs types de relations peuvent connecter deux nœuds. L'application de cette méthode aux réseaux multiplexes consiste à fusionner les partitions trouvées sur chaque type de relations.

Concernant TopoCent, cette mesure peut être améliorée en introduisant une heuristique qui estime le nombre de niveaux de voisins nécessaires pour calculer la centralité d'un nœud. Une idée simple est d'utiliser les techniques présentées dans les travaux portant sur la détection de communautés par expansion, qui élargissent les communautés autour d'un nœud graine.

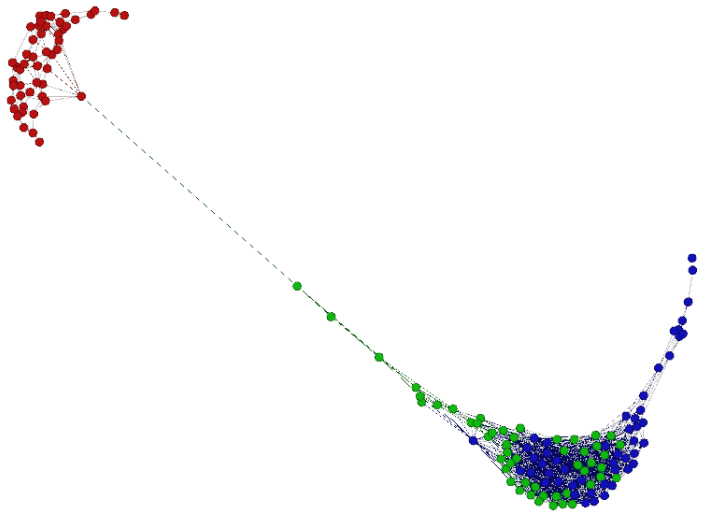
Pour l'évaluation des algorithmes de détection de communautés, nous nous intéressons à la tâche de visualisation comme une autre tâche d'évaluation. Des travaux récents [Yang *et al.*, 2013] ont commencé à aborder la visualisation des graphes en utilisant leur structure communautaire. Notre objectif sera d'évaluer les partitions trouvées par les différents algorithmes dans le cadre de la visualisation des graphes.

Annexes

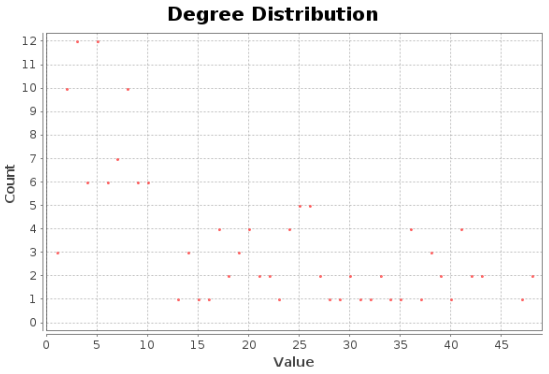
Annexe A

Graphes de voisinages générés pour la méthode CLE

A.1 Structure des GVRs générés

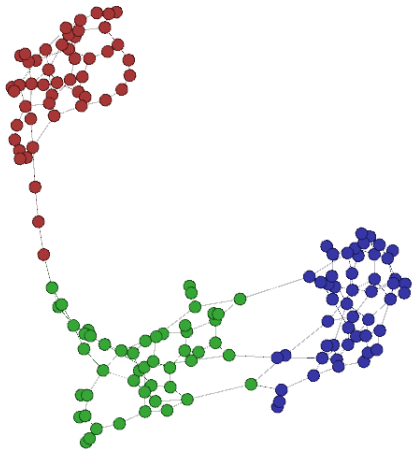


(a) Structure du graphe

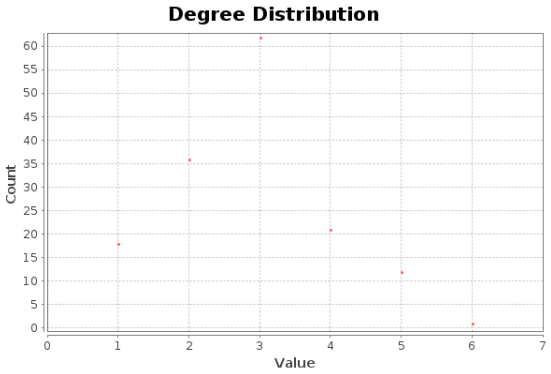


(b) Distribution de degré

FIGURE A.1 – Iris : GVR généré en fonction de la distance chebyshev

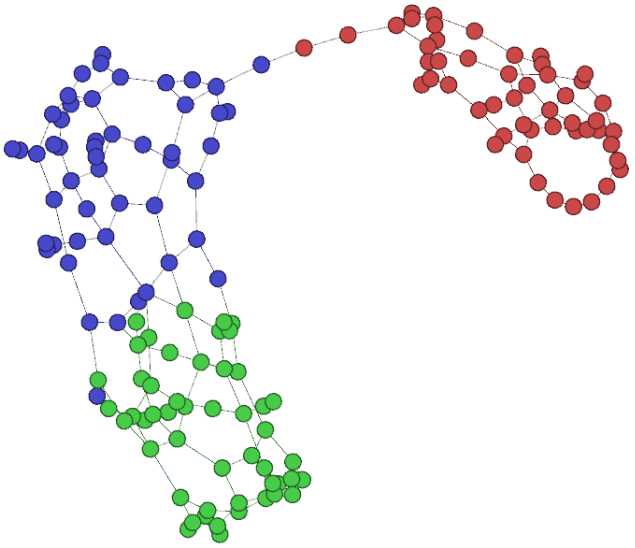


(a) Structure du graphe

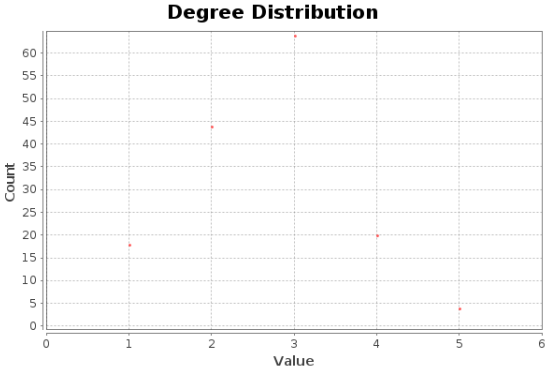


(b) Distribution de degré

FIGURE A.2 – Iris : GVR généré en fonction de la distance cosinus

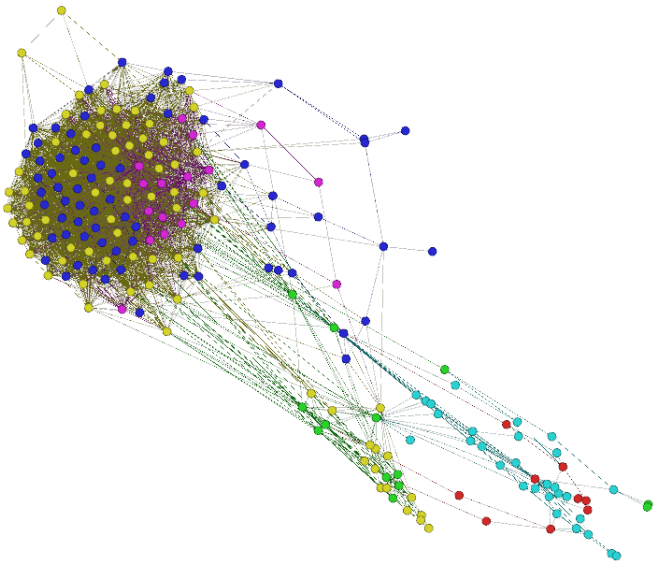


(a) Structure du graphe

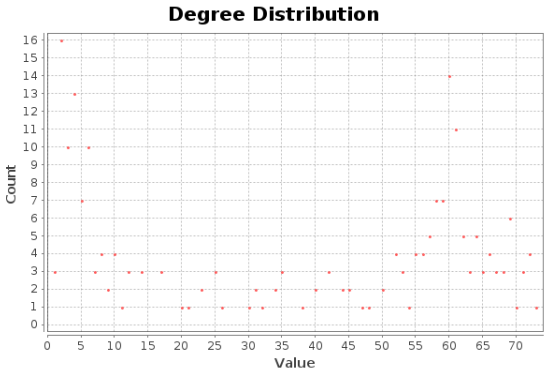


(b) Distribution de degré

FIGURE A.3 – Iris : GVR généré en fonction de la distance de corrélation

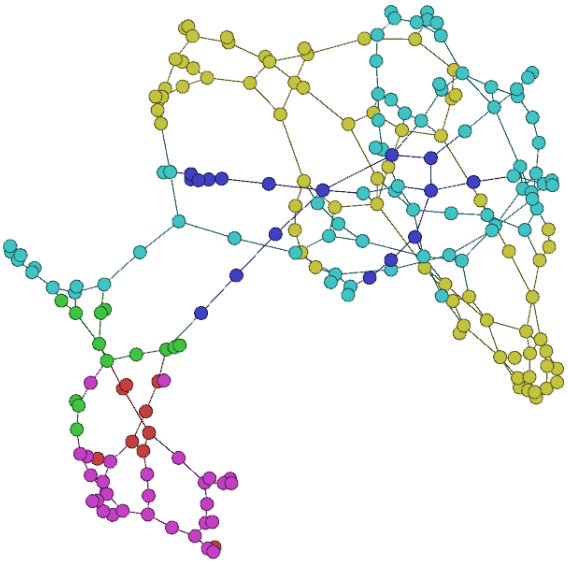


(a) Structure du graphe

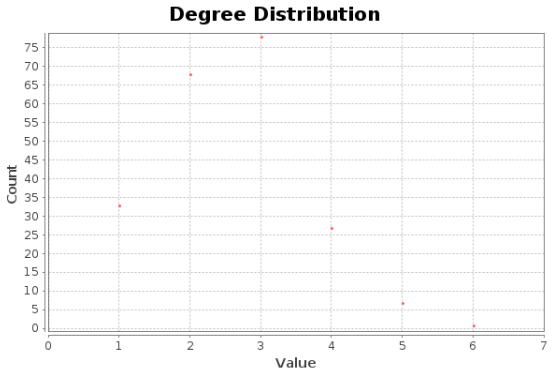


(b) Distribution de degré

FIGURE A.4 – Glass : GVR généré en fonction de la distance de chebyshev

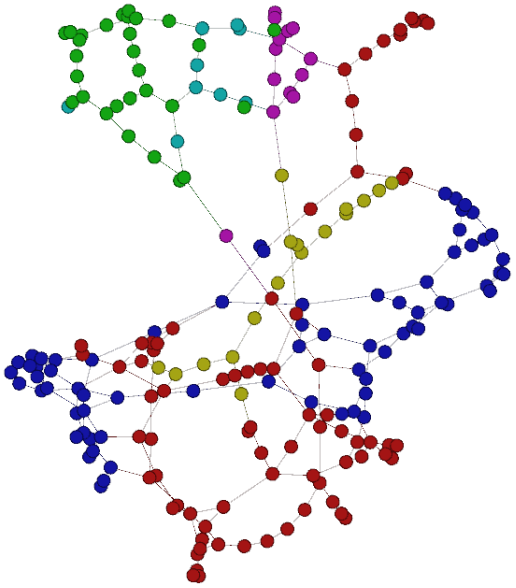


(a) Structure du graphe

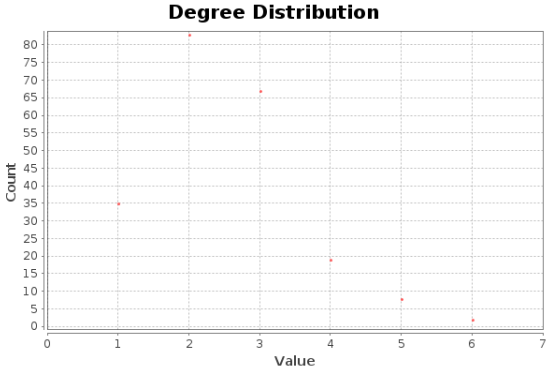


(b) Distribution de degré

FIGURE A.5 – Glass : GVR généré en fonction de la distance cosinus

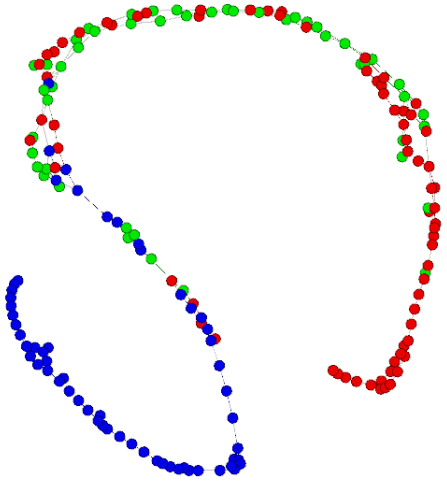


(a) Structure du graphe

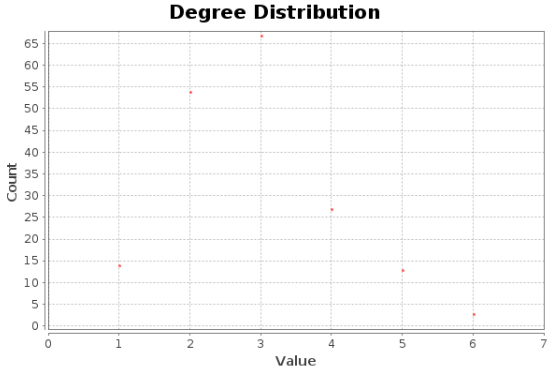


(b) Distribution de degré

FIGURE A.6 – Glass : GVR généré en fonction de la distance de corrélation

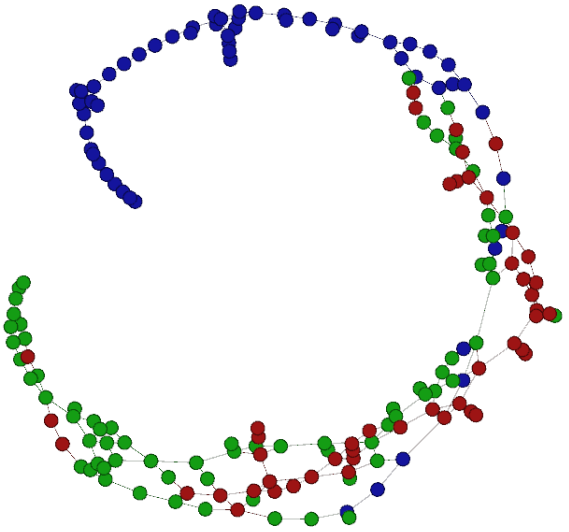


(a) Structure du graphe

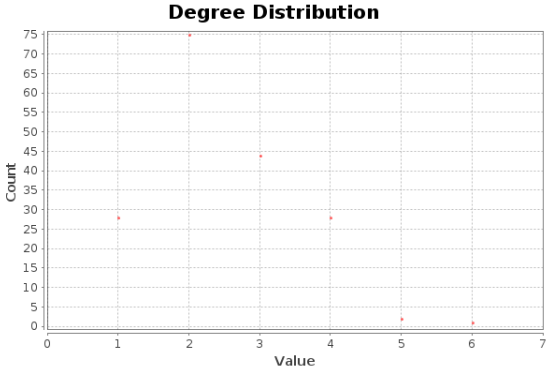


(b) Distribution de degré

FIGURE A.7 – Wine : GVR généré en fonction de la distance de chebyshev

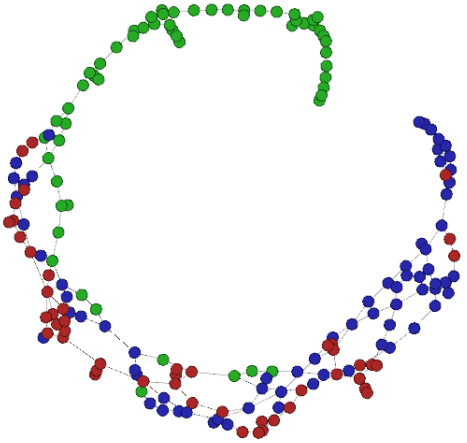


(a) Structure du graphe

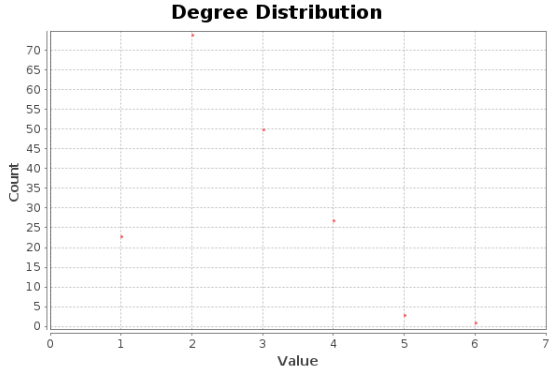


(b) Distribution de degré

FIGURE A.8 – Wine : GVR généré en fonction de la distance cosinus

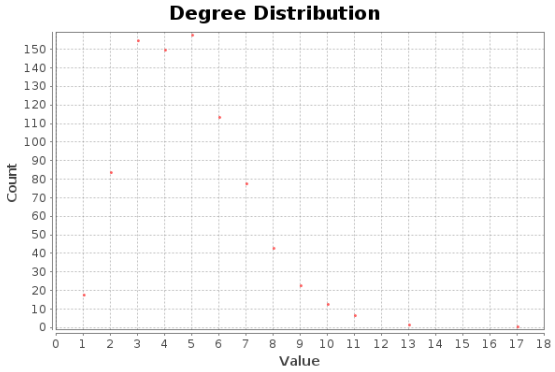


(a) Structure du graphe

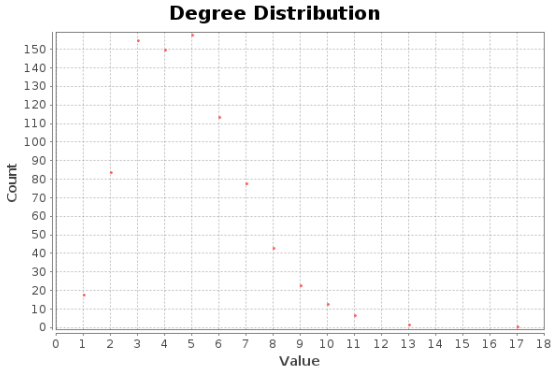


(b) Distribution de degré

FIGURE A.9 – Wine : GVR généré en fonction de la distance de corrélation

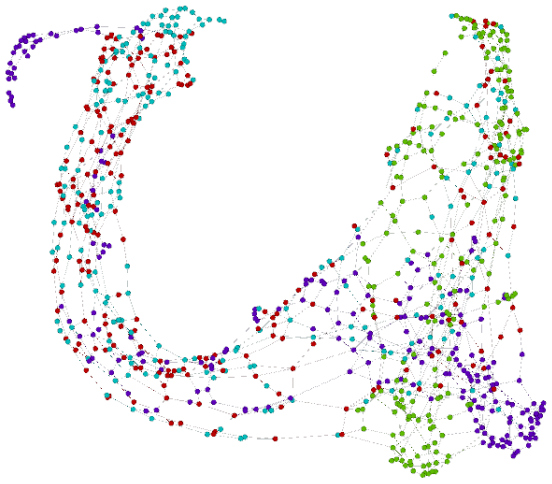


(a) Structure du graphe

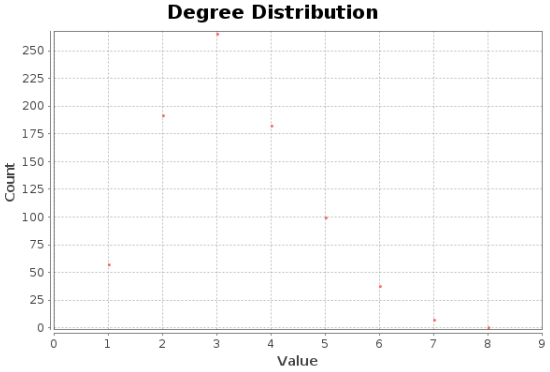


(b) Distribution de degré

FIGURE A.10 – Vehicle : GVR généré en fonction de la distance de chebyshev



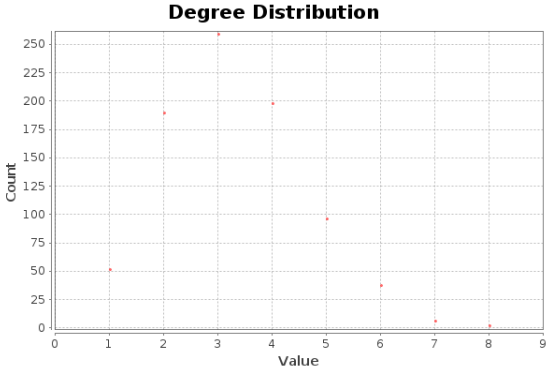
(a) Structure du graphe



(b) Distribution de degré

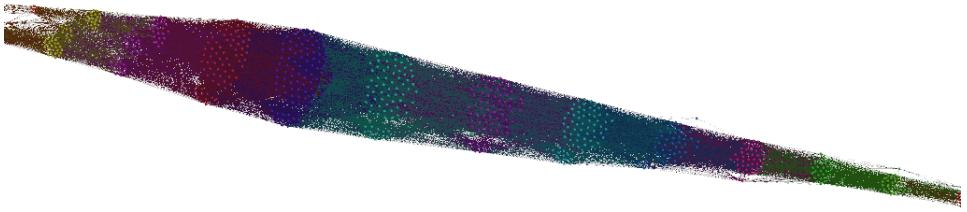
FIGURE A.11 – Vehicle : GVR généré en fonction de la distance cosinus

(a) Structure du graphe

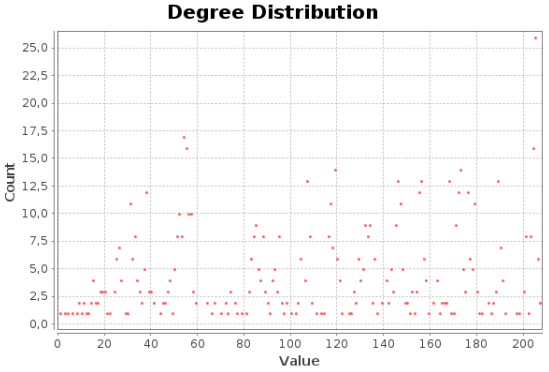


(b) Distribution de degré

FIGURE A.12 – Vehicle : GVR généré en fonction de la distance de corrélation



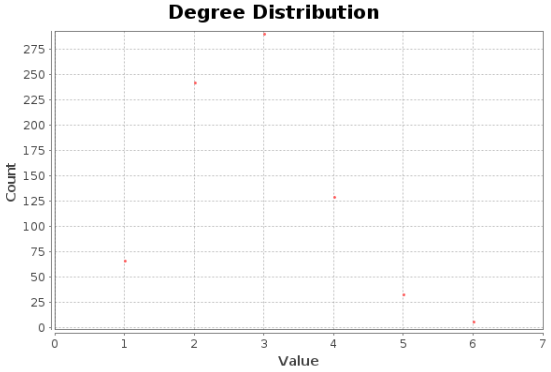
(a) Structure du graphe



(b) Distribution de degré

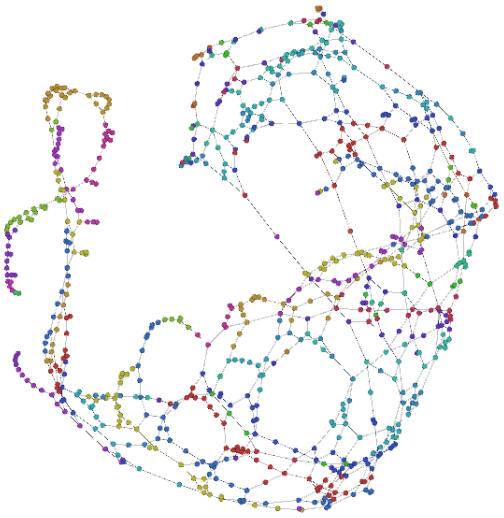
FIGURE A.13 – Abalone : GVR généré en fonction de la distance de chebyshev

(a) Structure du graphe

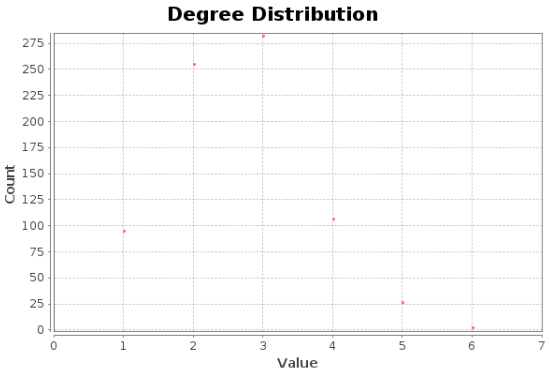


(b) Distribution de degré

FIGURE A.14 – Abalone : GVR généré en fonction de la distance cosinus



(a) Structure du graphe



(b) Distribution de degré

FIGURE A.15 – Abalone : GVR généré en fonction de la distance de corrélation

A.2 Résultats des algorithmes de détection de communautés sur les GVRs

Données	Algo	NMI	ARI	Q	#communautés
Iris	GN	0.62	0.47	0.21	7
	Walktrap	0.57	0.36	0.22	8
	EV	0.61	0.55	0.34	4
	Infomap	0.55	0.32	0.21	8
	FastGreedy	0.61	0.55	0.34	4
	Louvain	0.59	0.51	0.35	5
	LICOD	0.57	0.50	0.21	9
	it-LICOD	0.50	0.30	0.17	17
Wine	GN	0.32	0.12	0.83	14
	Walktrap	0.32	0.11	0.82	14
	EV	0.32	0.13	0.81	13
	Infomap	0.29	0.06	0.79	26
	FastGreedy	0.34	0.16	0.83	10
	Louvain	0.34	0.16	0.83	11
	LICOD	0.34	0.25	0.74	14
	it-LICOD	0.29	0.04	0.67	30
Glass	GN	0.47	0.25	0.09	29
	Walktrap	0.45	0.26	0.08	7
	EV	0.23	0.07	0.13	4
	Infomap	0.48	0.28	0.09	10
	FastGreedy	0.24	0.09	0.14	3
	Louvain	0.37	0.19	0.15	4
	LICOD	0.41	0.25	0.07	15
	it-LICOD	0.31	0.22	0.08	3
Vehicle	GN	0.21	0.09	0.77	15
	Walktrap	0.20	0.06	0.74	29
	EV	0.17	0.09	0.74	10
	Infomap	0.20	0.04	0.71	50
	FastGreedy	0.20	0.11	0.73	9
	Louvain	0.21	0.08	0.78	13
	LICOD	0.20	0.06	0.65	31
	it-LICOD	0.21	0.03	0.53	88
Abalone	GN	0.65	0.30	0.35	119
	Walktrap	0.73	0.40	0.44	8
	EV	0.72	0.44	0.43	6
	Infomap	0.75	0.44	0.44	7
	FastGreedy	0.39	0.14	0.37	2
	Louvain	0.64	0.35	0.45	4
	LICOD	0.18	0.43e-2	0.02e-2	98
	it-LICOD	0.12	0.03	0.04	

TABLE A.1 – Performance des algorithmes sur les GVRs-Chebyshev

Données	Algo	NMI	ARI	Q	#communautés
Iris	GN	0.66	0.44	0.72	9
	Walktrap	0.64	0.47	0.68	12
	EV	0.65	0.43	0.67	11
	Infomap	0.52	0.18	0.68	20
	FastGreedy	0.62	0.43	0.70	9
	Louvain	0.59	0.40	0.72	8
	LICOD	0.59	0.42	0.64	8
	it-LICOD	0.55	0.28	0.65	14
Wine	GN	0.32	0.14	0.79	11
	Walktrap	0.32	0.11	0.77	15
	EV	0.30	0.14	0.74	11
	Infomap	0.32	0.08	0.77	22
	FastGreedy	0.29	0.12	0.77	11
	Louvain	0.31	0.13	0.79	12
	LICOD	0.34	0.21	0.72	14
	it-LICOD	0.32	0.08	0.65	32
Glass	GN	0.45	0.21	0.76	11
	Walktrap	0.49	0.15	0.73	22
	EV	0.43	0.17	0.73	14
	Infomap	0.47	0.11	0.71	31
	FastGreedy	0.47	0.20	0.75	13
	Louvain	0.47	0.21	0.75	12
	LICOD	0.46	0.17	0.70	18
	it-LICOD	0.45	0.12	0.65	31
Vehicle	GN	0.23	0.10	0.79	17
	Walktrap	0.23	0.06	0.75	32
	EV	-	-	-	-
	Infomap	0.25	0.03	0.70	74
	FastGreedy	0.25	0.12	0.78	12
	Louvain	0.25	0.11	0.78	14
	LICOD	0.21	0.05	0.65	41
	it-LICOD	0.23	0.03	0.59	75
Abalone	GN	0.34	0.10	0.83	15
	Walktrap	0.33	0.08	0.82	21
	EV	0.29	0.08	0.80	15
	Infomap	0.51	0.09	0.76	80
	FastGreedy	0.30	0.09	0.84	14
	Louvain	0.35	0.10	0.83	19
	LICOD	0.44	0.08	0.70	68
	it-LICOD	0.50	0.06	0.61	128

TABLE A.2 – Performance des algorithmes sur les GVRs-Cosinus

....

Données	Algo	NMI	ARI	Q	#communautés
Iris	GN	0.61	0.43	0.73	8
	Walktrap	0.52	0.28	0.70	14
	EV	0.61	0.43	0.72	8
	Infomap	0.47	0.16	0.70	21
	FastGreedy	0.64	0.47	0.73	8
	Louvain	0.64	0.48	0.73	7
	LICOD	0.64	0.50	0.65	8
	it-LICOD	0.46	0.12	0.57	30
Wine	GN	0.28	0.14	0.80	10
	Walktrap	0.27	0.10	0.79	13
	EV	0.29	0.16	0.77	9
	Infomap	0.30	0.07	0.77	23
	FastGreedy	0.29	0.14	0.79	10
	Louvain	0.27	0.12	0.80	10
	LICOD	0.29	0.18	0.72	12
	it-LICOD	0.34	0.12	0.57	30
Glass	GN	0.54	0.29	0.78	11
	Walktrap	0.47	0.20	0.75	13
	EV	0.52	0.24	0.77	12
	Infomap	0.50	0.12	0.72	32
	FastGreedy	0.50	0.27	0.77	11
	Louvain	0.48	0.21	0.77	12
	LICOD	0.47	0.16	0.69	20
	it-LICOD	0.49	13	0.64	37
Vehicle	GN	0.23	0.10	0.79	17
	Walktrap	0.23	0.08	0.76	25
	EV	0.19	0.09	0.75	16
	Infomap	0.24	0.03	0.70	74
	FastGreedy	0.23	0.11	0.77	13
	Louvain	0.23	0.11	0.77	14
	LICOD	0.20	0.05	0.68	37
	it-LICOD	0.23	0.02	0.56	94
Abalone	GN	0.38	0.11	0.85	19
	Walktrap	0.41	0.10	0.82	36
	EV	0.36	0.11	0.81	24
	Infomap	0.53	0.10	0.78	82
	FastGreedy	0.38	0.12	0.85	18
	Louvain	0.37	0.11	0.85	18
	LICOD	0.47	0.11	0.74	59
	it-LICOD	0.52	0.08	0.66	113

TABLE A.3 – Performance des algorithmes sur les GVRs-Corrélation

Bibliographie

[Aggarwal et Reddy, 2014] Aggarwal, C. C. et Reddy, C. K., éditeurs (2014). *Data Clustering : Algorithms and Applications*. CRC Press.

[Albert *et al.*, 1999] Albert, R., Jeong, H. et Barabasi, A. L. (1999). The diameter of the world wide web. *Nature*, 401:130–131.

[Aynaud, 2011] Aynaud, T. (2011). *Détection de communatués dans les réseaux dynamiques*. Thèse de doctorat, Université Pierre et Marie Curie.

[Banerjee *et al.*, 2005] Banerjee, A., Dhillon, I. S., Ghosh, J. et Sra, S. (2005). Clustering on the unit hypersphere using von mises-fisher distributions. *J. Mach. Learn. Res.*, 6:1345–1382.

[Barabasi et Albert, 1999] Barabasi, A. L. et Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286:509–512.

[Baumes *et al.*, 2005] Baumes, J., Goldberg, M. K., Krishnamoorthy, M. S., Ismail, M. M. et Preston, N. (2005). Finding communities by clustering a graph into overlapping subgraphs. *In* Guimaraes, N. et Isaias, P. T., éditeurs : *IADIS AC*, pages 97–104. IADIS.

[Benchettara *et al.*, 2010] Benchettara, N., Kanawati, R. et Rouveirol, C. (2010). A supervised machine learning link prediction approach for academic collaboration recommendation. *In Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 253–256, New York, NY, USA. ACM.

[Blondel *et al.*, 2008] Blondel, V. D., Guillaume, J.-L., Lambiotte, R. et Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment*, P10008(10):1–12.

[Bollobás et Riordan, 2009] Bollobás, B. et Riordan, O. (2009). Clique percolation. *Random Struct. Algorithms*, 35(3):294–322.

[Box *et al.*, 2005] Box, G., Hunter, J. et Hunter, W. (2005). *Statistics for experimenters : design, innovation, and discovery*. Wiley series in probability and statistics. Wiley-Interscience.

[Brandes, 2001] Brandes, U. (2001). A Faster Algorithm for Betweenness Centrality. *Journal of Mathematical Sociology*, 25(2):163–177.

[Brandes *et al.*, 2008] Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z. et Wagner, D. (2008). On modularity clustering. *IEEE Trans. on Knowl. and Data Eng.*, 20(2):172–188.

[Brandes *et al.*, 2007] Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z. et Wagner, D. (2007). On finding graph clusterings with maximum modularity. *In* Brandstädt, A., Kratsch, D. et Müller, H., éditeurs : *Graph-Theoretic Concepts in Computer Science*, volume 4769 de *Lecture Notes in Computer Science*, pages 121–132. Springer, Berlin / Heidelberg.

[Cai *et al.*, 2011] Cai, Y., Shi, C., Dong, Y., Ke, Q. et Wu, B. (2011). A novel genetic algorithm for overlapping community detection. *In* Tang, J., King, I., Chen, L. et Wang, J., éditeurs : *ADMA (1)*, volume 7120 de *Lecture Notes in Computer Science*, pages 97–108. Springer.

[Chen *et al.*, 2012] Chen, D., Lü, L., Shang, M.-S., Zhang, Y.-C. et Zhou, T. (2012). Identifying influential nodes in complex networks. *Physica A : Statistical Mechanics and its Applications*, 391(4):1777–1787.

[Chen *et al.*, 2009a] Chen, J., Zaïane, O. et Goebel, R. (2009a). Local community identification in social networks. *In* Memon, N. et Alhajj, R., éditeurs : *ASONAM*, pages 237–242. IEEE Computer Society.

[Chen *et al.*, 2009b] Chen, J., Zaïane, O. R. et Goebel, R. (2009b). Detecting communities in large networks by iterative local expansion. *In* Abraham, A., Snásel, V. et Wegrzyn-Wolska, K., éditeurs : *CASoN*, pages 105–112. IEEE Computer Society.

[Chen et Fang, 2012] Chen, Q. et Fang, M. (2012). An efficient algorithm for community detection in complex networks. *In The 6th SNA-KDD Workshop'12*, Bijing, China.

[Chevaleyre *et al.*, 2007] Chevaleyre, Y., Endriss, U., Lang, J. et Maudet, N. (2007). A short introduction to computational social choice. *In* van Leeuwen, J., Italiano, G. F., van der Hoek, W., Meinel, C., Sack, H. et Plasil, F., éditeurs : *SOFSEM (1)*, volume 4362 de *Lecture Notes in Computer Science*, pages 51–69. Springer.

[Clauset, 2005] Clauset, A. (2005). Finding local community structure in networks. *Phys. Rev. E*, 72:026132.

[Clauset *et al.*, 2004] Clauset, A., Newman, M. E. J. et Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, 70(6):066111.

[Cordasco et Gargano, 2012] Cordasco, G. et Gargano, L. (2012). Label propagation algorithm : a semi-synchronous approach. *IJSNM*, 1:3–26.

[Coscia *et al.*, 2011] Coscia, M., Giannotti, F. et Pedreschi, D. (2011). A classification for community discovery methods in complex networks. *Stat. Anal. Data Min.*, 4(5):512–546.

[De Montgolfier *et al.*, 2011] De Montgolfier, F., Soto, M. et Viennot, L. (2011). Asymptotic modularity of some graph classes. *In* Asano, T., Nakano, S.-I., Okamoto, Y. et Watanabe, O., éditeurs : *ISAAC*, volume 7074 de *Lecture Notes in Computer Science*, pages 435–444. Springer.

[Duch et Arenas, 2005] Duch, J. et Arenas, A. (2005). Community detection in complex networks using extremal optimization. *Physical Review E*, 72:027104.

[Dunbar, 1998] Dunbar, R. (1998). *Grooming, Gossip, and the Evolution of Language*. Harvard University Press.

[Dwork *et al.*, 2001] Dwork, C., Kumar, R., Naor, M. et Sivakumar, D. (2001). Rank aggregation methods for the web. *In Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 613–622, New York, NY, USA. ACM.

[Ebel *et al.*, 2002] Ebel, H., Mielsch, L. I. et Bornholdt, S. (2002). Scale-free topology of e-mail networks. *Physical Review E*, 66:035103.

[Erdős et Rényi, 1959] Erdős, P. et Rényi, A. (1959). On random graphs. I. *Publ. Math. Debrecen*, 6:290–297.

[Faloutsos *et al.*, 1999] Faloutsos, M., Faloutsos, P. et Faloutsos, C. (1999). On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev.*, 29(4):251–262.

[Flake *et al.*, 2000] Flake, G. W., Lawrence, S. et Giles, C. L. (2000). Efficient identification of web communities. *In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 150–160, New York, NY, USA. ACM.

[Fortunato, 2010] Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174.

[Fortunato et Barthélemy, 2007] Fortunato, S. et Barthélemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1).

[Fortunato et Lancichinetti, 2009] Fortunato, S. et Lancichinetti, A. (2009). Community detection algorithms : A comparative analysis : Invited presentation, extended abstract. *In Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS '09, pages 27 :1–27 :2. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[Girvan et Newman, 2002] Girvan, M. et Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826.

[Good *et al.*, 2010] Good, B., Montjoye, Y. D. et Clauset, A. (2010). Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106.

[Guillaume et Latapy, 2006] Guillaume, J.-L. et Latapy, M. (2006). Bipartite graphs as models of complex networks. *Physica A : Statistical Mechanics and its Applications*, 371(2):795 – 813.

[Guimera *et al.*, 2004] Guimera, R., Sales-Pardo, M. et Amaral, L. (2004). Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2):025101.

[Hubert et Arabie, 1985] Hubert, L. et Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218.

[Jeong *et al.*, 2000] Jeong, H., Tombor, B., Albert, R., Oltvai, Z. N. et Barabasi, A. L. (2000). The large-scale organization of metabolic networks. *Nature*, 407:651–654.

[Kanawati, 2011] Kanawati, R. (2011). Licod : Leaders identification for community detection in complex networks. *In SocialCom/PASSAT*, pages 577–582. IEEE.

[Kanawati, 2014] Kanawati, R. (2014). Empirical evaluation of applying ensemble ranking to ego-centered communities identification in complex network. *In ZAZ, Y. (ed.) 4th International Conference on Multimedia Computing and Systems*, ICMCS'14.

[Karrer *et al.*, 2008] Karrer, B., Levina, E. et Newman, M. E. J. (2008). Robustness of community structure in networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 77:046119.

[Katz, 1953] Katz, L. (1953). A new status index derived from sociometric analysis. *Psychometrika*, 18:39–43.

[Kelley, 2009] Kelley, S. (2009). *The Existence and Discovery of Overlapping Communities in Large-Scale Networks.* Thèse de doctorat, Rensselaer Polytechnic Institute, Troy, NY.

[Koschützki *et al.*, 2005] Koschützki, D., Lehmann, K., Peeters, L., Richter, S., Tenfelde-Podehl, D. et Zlotowski, O. (2005). Centrality indices. *In* Brandes, U. et Erlebach, T., éditeurs : *Network Analysis*, volume 3418 de *Lecture Notes in Computer Science*, pages 16–61. Springer, Berlin / Heidelberg.

[Labatut, 2012] Labatut, V. (2012). Une nouvelle mesure pour l'évaluation des méthodes de détection de communautés. *In 3ème Conférence sur les modèles et l'analyse de réseaux : approches mathématiques et informatiques*, MARAMI'12, Villetaneuse, France.

[Lancichinetti et Fortunato, 2011] Lancichinetti, A. et Fortunato, S. (2011). Limits of modularity maximization in community detection. *Phys. Rev. E*, 84:066122.

[Lancichinetti *et al.*, 2008] Lancichinetti, A., Fortunato, S. et Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, 78:046110.

[Lee et Cunningham, 2013] Lee, C. et Cunningham, P. (2013). Community detection : effective evaluation on large social networks. *Journal of Complex Networks.*

[Leung *et al.*, 2009] Leung, I. X. Y., Hui, P., Liò, P. et Crowcroft, J. (2009). Towards real-time community detection in large networks. *Phys. Rev. E*, 79:066107.

[Li et Song, 2013] Li, J. et Song, Y. (2013). Community detection in complex networks using extended compact genetic algorithm. *Soft Comput.*, 17(6):925–937.

[Luo *et al.*, 2008] Luo, F., Wang, J. et Promislow, E. (2008). Exploring local community structures in large networks. *Web Intelligence and Agent Systems*, 6(4):387–400.

[Lusseau *et al.*, 2003] Lusseau, D., Schneider, K., Boisseau, O., Haase, P., Slooten, E. et Dawson, S. (2003). The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405.

[Mancoridis *et al.*, 1998] Mancoridis, S., Mitchell, B. S., Rorres, C., Chen, Y. et Gansner, E. R. (1998). Using Automatic Clustering to Produce High-Level System Organizations of Source Code. *In IWPC '98 : Proceedings of the 6th International Workshop on Program Comprehension*, Washington, DC, USA. IEEE Computer Society.

[Manning *et al.*, 2008] Manning, C. D., Raghavan, P. et Schütze, H. (2008). *Introduction to Information Retrieval.* Cambridge University Press, New York, NY, USA.

[Meilă, 2005] Meilă, M. (2005). Comparing clusterings : An axiomatic view. *In Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 577–584, New York, NY, USA. ACM.

[Milgram, 1967] Milgram, S. (1967). The small world problem. *Psychology Today*, 2:60–67.

[Molloy et Reed, 1995] Molloy, M. et Reed, B. (1995). A critical point for random graphs with a given degree sequence. *Random Struct. Algorithms*, 6(2/3):161–179.

[Nanopoulos *et al.*, 2009] Nanopoulos, A., Gabriel, H.-H. et Spiliopoulou, M. (2009). Spectral clustering in social-tagging systems. *In* Vossen, G., Long, D. D. E. et Yu, J. X., éditeurs : *WISE*, volume 5802 de *Lecture Notes in Computer Science*, pages 87–100. Springer.

[Newman, 2006a] Newman, M. E. (2006a). Modularity and community structure in networks. *Proc Natl Acad Sci U S A*, 103.

[Newman, 2001] Newman, M. E. J. (2001). Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Phys. Rev. E*, 64:016132.

[Newman, 2003] Newman, M. E. J. (2003). The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256.

[Newman, 2004] Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:066133.

[Newman, 2006b] Newman, M. E. J. (2006b). Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74.

[Okamoto *et al.*, 2008] Okamoto, K., Chen, W. et Li, X.-Y. (2008). Ranking of closeness centrality for large-scale social networks. *In Proceedings of the 2Nd Annual International Workshop on Frontiers in Algorithmics*, FAW '08, pages 186–195, Berlin, Heidelberg. Springer-Verlag.

[Orman et Labatut, 2009] Orman, G. K. et Labatut, V. (2009). A comparison of community detection algorithms on artificial networks. *In Proceedings of the 12th International Conference on Discovery Science*, pages 242–256. Springer-Verlag.

[Orman *et al.*, 2012] Orman, G. K., Labatut, V. et Cherifi, H. (2012). Comparative evaluation of community detection algorithms : A topological approach. *CoRR*, abs/1206.4987.

[Orman *et al.*, 2013] Orman, G. K., Labatut, V. et Cherifi, H. (2013). Towards realistic artificial benchmark for community detection algorithms evaluation. *IJWBC*, 9(3):349–370.

[Palla *et al.*, 2005] Palla, G., Derényi, I., Farkas, I. et Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818.

[Pan *et al.*, 2012] Pan, L., Dai, C., Wang, C., Xie, J. et Liu, M. (2012). Overlapping community detection via leader-based local expansion in social networks. *In ICTAI*, pages 397–404. IEEE.

[Papadopoulos *et al.*, 2010] Papadopoulos, S., Kompatsiaris, Y. et Vakali, A. (2010). A graph-based clustering scheme for identifying related tags in folksonomies. *In* Pedersen, T. B., Mohania, M. K. et Tjoa, A. M., éditeurs : *DaWak*, volume 6263 de *Lecture Notes in Computer Science*, pages 65–76. Springer.

[Papadopoulos *et al.*, 2012] Papadopoulos, S., Vakali, A. et Kompatsiaris, I. (2012). *Community Detection in Collaborative Tagging Systems*. Springer.

[Pfitzner *et al.*, 2009] Pfitzner, D., Leibbrandt, R. et Powers, D. M. W. (2009). Characterization and evaluation of similarity measures for pairs of clusterings. *Knowl. Inf. Syst.*, 19(3):361–394.

[Pizzuti, 2012] Pizzuti, C. (2012). Boosting the detection of modular community structure with genetic algorithms and local search. *In* Ossowski, S. et Lecca, P., éditeurs : *SAC*, pages 226–231. ACM.

[Plantié et Crampes, 2013] Plantié, M. et Crampes, M. (2013). Survey on social community detection. *In* Ramzan, N., van Zwol, R., Lee, J.-S., Clüver, K. et Hua, X.-S., éditeurs : *Social Media Retrieval*, Computer Communications and Networks, pages 65–85. Springer London.

[Poncela *et al.*, 2008] Poncela, J., Gómez-Gardeñes, J., Floría, L. M., Sánchez, A. et Moreno, Y. (2008). Complex cooperative networks from evolutionary preferential attachment. *PLoS ONE*, 3(6):2449.

[Pons et Latapy, 2004] Pons, P. et Latapy, M. (2004). Computing communities in large networks using random walks. *J. Graph Algorithms Appl.*, 10(2):284–293.

[Pujari et Kanawati, 2013] Pujari, M. et Kanawati, R. (2013). Link prediction in multiplex bibliographical networks. *International Journal of Complex Systems in Science proceedings of NET-WORKS 2013, El Escorial.*

[Radicchi *et al.*, 2004] Radicchi, F., Castellano, C., Cecconi, F., Loreto, V. et Parisi, D. (2004). Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences*, 101(9):2658.

[Raghavan *et al.*, 2007] Raghavan, U. N., Albert, R. et Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106.

[Rand, 1971] Rand, W. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

[Reichardt et Bornholdt, 2006] Reichardt, J. et Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E*, 74:016110.

[Reihaneh *et al.*, 2010] Reihaneh, R., Jiyang, C. et Osmar, R. Z. (2010). Top leaders community detection approach in information networks. *In 4th SNA-KDD Workshop on Social Network Mining and Analysis (in conjunction with the 16th ACM SIGKDD conference).*

[Reka et Barabási, 2002] Reka, A. et Barabási (2002). Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74:47–97.

[Resnick *et al.*, 1994] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P. et Riedl, J. (1994). Grouplens : An open architecture for collaborative filtering of netnews. *In Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186, New York, NY, USA. ACM.

[Rosvall et Bergstrom, 2008] Rosvall, M. et Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123.

[Schifanella *et al.*, 2010] Schifanella, R., Barrat, A., Cattuto, C., Markines, B. et Menczer, F. (2010). Folks in folksonomies : Social link prediction from shared metadata. *In Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 271–280, New York, NY, USA. ACM.

[Sculley, 2007] Sculley, D. (2007). Rank aggregation for similar items. *In SDM*, pages 587–592. SIAM.

[Serrano et Boguñá, 2005] Serrano, M. Á. et Boguñá, M. (2005). Weighted configuration model. *In* Mendes, J., Oliveira, J. G., Abreu, F. V., Povolotsky, A. et Dorogovtsev, S. N., éditeurs : *Science of Complex Networks : From Biology to the Internet and WWW; CNET 2004*, volume 776 de *American Institute of Physics Conference Series*, pages 101–107.

[Shah et Zaman, 2010] Shah, D. et Zaman, T. (2010). Community detection in networks : The leader-follower algorithm. *CoRR*, abs/1011.0774.

[Shi et Malik, 2000] Shi, J. et Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905.

[Soundarajan et Hopcroft, 2012] Soundarajan, S. et Hopcroft, J. E. (2012). Using community information to improve the precision of link prediction methods. *In* Mille, A., Gandon, F. L., Misselis, J., Rabinovich, M. et Staab, S., éditeurs : *WWW (Companion Volume)*, pages 607–608. ACM.

[Specia et Motta, 2007] Specia, L. et Motta, E. (2007). Integrating folksonomies with the semantic web. *In Proceedings of the 4th European Conference on The Semantic Web : Research and Applications*, ESWC '07, pages 624–639, Berlin, Heidelberg. Springer-Verlag.

[Steinley, 2004] Steinley, D. (2004). Properties of the Hubert-Arabie adjusted Rand index. *Psychological methods*, 9(3):386–396.

[Strehl et Ghosh, 2003] Strehl, A. et Ghosh, J. (2003). Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617.

[Toussaint, 1980] Toussaint, G. T. (1980). The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268.

[Watts et Strogatz, 1998] Watts, D. J. et Strogatz, S. H. (1998). Collective dynamics of 'small-world'networks. *Nature*, 393(6684):409–10.

[Williams et Martinez, 2000] Williams, R. J. et Martinez, N. D. (2000). Simple rules yield complex food webs. *Nature*, 404(6774):180–183.

[Xie *et al.*, 2013] Xie, J., Kelley, S. et Szymanski, B. K. (2013). Overlapping community detection in networks : The state-of-the-art and comparative study. *ACM Comput. Surv.*, 45(4):43 :1–43 :35.

[Xu *et al.*, 2007] Xu, X., Yuruk, N., Feng, Z. et Schweiger, T. A. J. (2007). Scan : a structural clustering algorithm for networks. *In* Berkhin, P., Caruana, R. et Wu, X., éditeurs : *KDD*, pages 824–833. ACM.

[Yang et Leskovec, 2012] Yang, J. et Leskovec, J. (2012). Defining and evaluating network communities based on ground-truth. *In Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12, pages 3 :1–3 :8, New York, NY, USA. ACM.

[Yang *et al.*, 2013] Yang, J., Liu, Y., Zhang, X., Yuan, X., Zhao, Y., Barlowe, S. et Liu, S. (2013). Piwi : Visually exploring graphs based on their community structure. *IEEE Trans. Vis. Comput. Graph.*, 19:1034–1047.

[Young et Levenglick, 1978] Young, H. P. et Levenglick, A. (1978). A Consistent Extension of Condorcet's Election Principle. *SIAM Journal on Applied Mathematics*, 35.

[Zachary, 1977] Zachary, W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473.