THÈSE DE DOCTORAT

Thèse encadrée par Fabien TARISSAN et Jean-Loup GUILLAUME

Pour obtenir le grade de
DOCTEUR DE L'UNIVERSITÉ PIERRE ET MARIE CURIE

Analyse de la structure communautaire des réseaux bipartis

Présentée par

Raphaël TACKX

soutenue le : 12/10/2018

Directrice :	Clémence MAGNIEN	Directeur de Rercherche, CNRS, UPMC
Rapporteurs :	Éric FLEURY	Professeur, ENS Lyon
	Bertrand JOUVE	Directeur de Recherche, CNRS, Université de Toulouse
Examinateur :	Matthieu LATAPY	Directeur de Recherche, CNRS, UPMC
Encadrants :	Jean-Loup GUILLAUME	Professeur, Université de La Rochelle
	Fabien TARISSAN	Chargé de Recherche, CNRS, ENS Paris-Saclay

Sorbonne Université - UPMC
Laboratoire d'Informatique de Paris 6

“Quand est-il venu si près de moi cet escargot ?”

Issa, Haïku

Remerciements

Cette thèse marque la fin d'une véritable aventure, riche en découvertes et parsemée de rencontres incroyables. Aujourd'hui, je peux constater la chance immense que j'ai eue...

Tout d'abord, je souhaiterai remercier mes encadrants, Fabien Tarissan et Jean-Loup Guillaume, pour leur accompagnement tout au long de ce travail. Sans eux cela n'aurait pas été possible. Je tiens à exprimer ma gratitude pour ce qu'ils m'ont transmis, pour leur ouverture d'esprit, leur patience et leur confiance. C'est aussi au fil de leurs conseils et réflexions que j'ai pu acquérir une vraie persévérance dans le travail, un esprit critique ainsi que pousser ma curiosité scientifique toujours plus loin sans pour autant perdre de vue mes objectifs. Je tiens également à les remercier de m'avoir intégrer dans le monde de la recherche, vaste terrain où l'on tisse des liens.

À ma famille et à mes proches qui m'ont encouragé dans cette voie, soutenu tout du long malgré mes doutes et incertitudes. Je le dois particulièrement à ma famille et mes amis, ma mère, mon père, mon grand-père, la famille Blondel/Larrondo, Liane Estrada et Pei-fen Yu.

J'adresse mes sincères remerciements à Clémence Magnien, du laboratoire d'Informatique de Paris 6, pour avoir accepté d'être ma directrice de thèse et pour ses précieux conseils.

Mais également à toute l'équipe *ComplexNetworks*, Matthieu Latapy, Maximilien Danisch, Lionel Tabourier, Rémy Cazabet, Patricia Conde Céspedes, Robin Lamarche-Perrin, Qinna Wang, Noé Gaumont, Tiphaine Viard, Audrey Wilmet, Thibaud Arnoux, Aurore Payen, Marwan Ghanem, Louisa Harutyunyan, Damien Nogues, Leonard Pancichi avec qui j'ai vécu le pire comme le meilleur dans une indescriptible effervescence de partage, de bonne humeur et d'entraide, à tout les niveaux. À tout ces neurones dépensés pendant les discussions poussées et résolutions de problèmes abracadabrantesques que j'ai eu avec vous. Tout aussi important, ces moments de relaxation passés ensemble ont été très réconfortants, salle de pause, niveau JU, bière Jussieu, session photos et frisbee, cantines, escalade, et j'en passe. Finalement, un grand merci à tous ceux avec qui j'ai pu collaborer durant ces années. Je tiens à remercier également l'équipe CNeRG de l'IIT Kharaghpur en Inde, particulièrement Soumajit Pramanik et Bivas Mitra, qui m'ont accueilli chaleureusement et avec qui j'ai pu créer des liens forts.

Table des matières

Liste des Figures

1 Introduction **1**

2 Notions générales pour les réseaux réels **5**
2.1 Propriétés des graphes . . . 6
2.1.1 Graphes unipartis . . . 6
2.1.1.1 Notations . . . 6
2.1.1.2 Voisinage d'un sommet . . . 8
2.1.1.3 Degré d'un sommet . . . 9
2.1.1.4 Concept de chemin dans les graphes . . . 9
2.1.1.5 Clustering . . . 10
2.1.1.6 Composantes connexes . . . 11
2.1.1.7 Clique et clique maximale . . . 11
2.1.1.8 Assortativité . . . 11
2.1.2 Graphes bipartis . . . 12
2.1.2.1 Clustering biparti . . . 14
2.1.2.2 Biclique . . . 15
2.2 Réseaux réels . . . 15
2.2.1 Principales propriétés . . . 16
2.2.2 Modèles aléatoires . . . 17

3 Structure communautaire **19**
3.1 Types de communautés . . . 21
3.1.1 Partitionnement . . . 21
3.1.2 Recouvrement . . . 21
3.1.3 Hiérarchie . . . 22
3.2 Une classification des algorithmes de détection de communautés . . . 23
3.2.1 Approches centrées réseau . . . 24
3.2.1.1 Approches d'optimisation . . . 24
3.2.1.2 Approches de clustering . . . 27
3.2.1.3 Approches par modèles de bloc . . . 31
3.2.2 Approches centrées groupes . . . 32
3.2.3 Approches centrées propagation . . . 33
3.3 Évaluation de la détection de communautés . . . 37

3.3.1 Les vérités de terrain . . . 37
3.3.2 La qualité d'une communauté . . . 41

4 Étude du recouvrement de la structure communautaire bipartie des réseaux sociaux d'Internet 44
4.1 Jeux de données . . . 46
4.2 Métriques pour le recouvrement . . . 47
4.2.1 Métriques classiques . . . 47
4.2.2 Nouvelles métriques . . . 48
4.3 Analyse de la structure bipartie . . . 49
4.3.0.1 Propriétés globales . . . 50
4.3.0.2 Métriques classiques biparties . . . 51
4.3.0.3 Affiner l'analyse avec les nouvelles métriques . . . 53
4.4 Comparaison à l'aléatoire . . . 55
4.5 Discussion . . . 58

5 Découvrir la structure communautaire des réseaux réels en utilisant la notion de cycle et la similarité des nœuds 60
5.1 Méthodologie . . . 62
5.1.1 L'algorithme ComSim . . . 63
5.1.2 Approches classiques . . . 64
5.2 Évaluation de ComSim . . . 66
5.2.1 Sur des jeux de données avec une vérité de terrain . . . 66
5.2.2 Sur un réseau social d'Internet . . . 68
5.2.2.1 Homogénéité des communautés détectées . . . 71
5.3 Discussion . . . 74

6 Structure communautaire des réseaux multi-couches 75
6.1 Représentation et résolution du problème . . . 78
6.2 Données . . . 79
6.2.1 Génération de réseaux artificiels . . . 79
6.3 Développement d'algorithmes de détection de communautés . . . 82
6.3.1 Propriétés souhaitables . . . 82
6.3.2 Modularité multi-couche . . . 82
6.3.2.1 Communautés multi-couches . . . 83
6.3.2.2 Communautés mono-couches . . . 84
6.3.3 Algorithmes d'optimisation . . . 85
6.4 Evaluation de Q_M . . . 86
6.4.1 Résultats expérimentaux . . . 87
6.4.1.1 Config A . . . 87
6.4.1.2 Config B . . . 88
6.5 Évaluation sur les réseaux artificiels . . . 88
6.5.1 Protocole expérimental . . . 88
6.5.2 Algorithmes en compétition . . . 89
6.5.2.1 Méthode avec mQ . . . 89
6.5.2.2 Approches à base de fusion . . . 89
6.5.2.3 Algorithmes de l'état de l'art . . . 89
6.5.3 Évaluation . . . 89

6.6 Évaluation sur des réseaux réels 91
6.6.1 Yelp 91
6.6.2 Meetup 93
6.7 Discussion 94

7 Conclusion **95**

Bibliographie **98**

Liste des Figures

2.1 Représentation graphique d'un graphe non-orienté (haut), orienté (milieu) et pondéré (bas) et de leur matrice d'adjacence A associée. 8
2.2 Graphe non connexe avec 3 composantes connexes. 11
2.3 Exemple d'un graphe biparti $\mathbb{B} = (\top, \bot, E_B)$ avec $\top = \{A, B, C, D, E, F\}$ et $\bot = \{1, 2, 3, 4, 5, 6, 7, 8\}$. 13
2.4 Exemples de bicliques. 15

3.1 Partition d'un graphe en 3 communautés. 20
3.2 Recouvrement entre deux communautés (bleue et rouge). 22
3.3 Un dendrogramme permet de représenter les différents niveaux d'une hiérarchie. La ligne rouge en pointillée indique le partitionnement optimal de la hiérarchie selon une mesure. Dans cet exemple, la meilleure coupe correspond aux communautés violette et verte. 23
3.4 Exemple de la réorganisation des lignes et colonnes d'une matrice d'adjacence. 29
3.5 Illustration du fonctionnement de Infomap (issue de l'article [RAB09]). L'image A représente les parcours du marcheur aléatoire que l'on souhaite encoder. L'image B présente la séquence des parcours : on attribue un identifiant unique sur chaque nœud qui sert à encoder le parcours (en noir, au-dessous). Les images C et D montrent comment optimiser la séquence : la répétition d'identifiants permet ainsi de définir les communautés conduisant à l'encodage minimal. 36
3.6 Une manière simple d'évaluer les communautés détectées par un algorithme avec la vérité de terrain. 38

4.1 Deux graphes bipartis présentant une structure différente autour du sommet 2 : non redondant dans (a) et fortement redondant dans (b). 48
4.2 Distribution cumulative inverse des degrés des ensembles $\top$ et $\bot$. 51
4.3 Distribution cumulative inverse des coefficients de clustering et de redondance. 51
4.4 Corrélation entre le degré et les coefficients de clustering et de redondance. 52
4.5 Distribution cumulative inverse des coefficients de dispersion et de monopole. 54
4.6 Corrélation entre le degré et les coefficients de dispersion et de monopole. 54
4.7 Distribution cumulative inverse du coefficient de clustering et corrélation entre le degré et le clustering dans les graphes bipartis aléatoires. 55

4.8 Distribution cumulative inverse du coefficient de redondance et corrélation entre le degré et la redondance dans les graphes bipartis aléatoires. 56
4.9 Distribution cumulative inverse du coefficient de dispersion et corrélation entre le degré et la dispersion dans les graphes bipartis aléatoires. 57
4.10 Distribution cumulative inverse du coefficient de monopole et corrélation entre le degré et le monopole dans les graphes bipartis aléatoires. 58

5.1 Exemple d'un graphe biparti $\mathbb{B} = (\top, \bot, E_B)$. 62
5.2 Exemple de la $\top$-projection pondérée du graphe biparti $\mathbb{B}$ en utilisant la similarité des voisins communs (CN) (voir Équation 3.3). 64
5.3 Évaluation de la qualité des partitions détectées par les algorithmes sur *20 newsgroups* et *Southern Women* à l'aide de la NMI (Équation 3.27) et du score-F1 (Équation 3.30). 67
5.4 Ces nuages de points montrent la relation entre les propriétés des communautés (en terme de densité interne et de séparabilité) et leurs tailles pour ComSim (haut), Louvain (milieu) et Infomap (bas) sur IMDb. 70
5.5 Exemple d'un graphe triparti $\mathbb{T} = (V_1, V_2, V_3, E_T)$ avec $V_1 = \{1, 2, 3, 4, 5, 6\}$, $V_2 = \{A, B, C, D, E\}$ et $V_3 = \{\alpha, \beta, \gamma, \delta, \epsilon, \zeta\}$. 71
5.6 Scores d'homogénéité normalisés par attribut pour les catégories Pays et Genres. 73

6.1 Illustration d'un réseau (Yelp) multi-couches. 76
6.2 Configurations avec deux types de communautés, communauté multi-couches et communautés mono-couches. 78
6.3 Variations de NMI en fonction de μ et p pour 'CompMod' et 'MetaFac' sur des réseaux à 2 couches avec 100 sommets sur chaque couche, générés avec un degré maximum $k^i_{max} = 10$, un degré moyen $\langle k_i \rangle = 6$ et une densité des liens de couplage $d = 0.07$. 81
6.4 87
6.5 NMI entre les vérités de terrain et les communautés identifiées pour différentes valeurs de d. 90
6.6 NMI entre les vérités de terrain et les communautés identifiées pour différentes valeurs de α et p. 91
6.7 Précision et score-F1 (moyenné sur tous les visiteurs) des communautés obtenues pour différents algorithmes et différentes recommandations. . . . 92
6.8 Précision et score-F1 (moyennés sur tous les groupes) des communautés obtenues sur Meetup N/W. 94

Chapitre 1

Introduction

Il existe dans le monde réel un nombre important de réseaux qui apparaissent naturellement, on les retrouve un peu partout, dans de nombreuses disciplines, par exemple en informatique avec les réseaux de routeurs, les réseaux de satellites, les réseaux de pages Web, en biologie avec les réseaux des neurones, en écologie avec les réseaux d'interactions biologiques, en linguistique avec les réseaux de synonymes, en droit avec les réseaux de décisions juridiques, en économie avec les réseaux interbancaires, en sciences humaines avec les réseaux sociaux. De manière générale, un réseau reflète les interactions entre les nombreuses entités d'un système. Ces interactions peuvent être de différentes natures, un lien social ou un lien d'amitié dans un réseau social constitué de personnes, un câble dans un réseau de routeurs, une réaction chimique dans un réseau biologique de protéines, un hyperlien dans un réseau de pages Web, etc. Plus encore, la rapide démocratisation du numérique dans nos sociétés, avec Internet notamment, a pour conséquence de produire de nouveaux systèmes qui peuvent être représentés sous forme de réseaux. Finalement, tous ces réseaux présentent des particularités bien spécifiques : ils sont issus de contextes pratiques, ils sont le plus souvent de grande taille (on retrouve parfois des réseaux constitués de plusieurs milliards de nœuds et de liens, contenant donc une grande quantité d'information), et ils présentent des propriétés statistiques communes. À cet égard, ils sont regroupés sous l'appellation de *réseaux réels*, *graphes de terrain* ou encore *réseaux complexes*.

Aujourd'hui, la science des réseaux est un domaine de recherche à part entière dont l'enjeu principal est de parvenir à décrire et modéliser leur structure afin de révéler leurs caractéristiques générales et de mieux comprendre leurs mécanismes. La plupart des travaux dans ce domaine utilisent le formalisme des *graphes* qui fournit un ensemble d'outils mathématiques particulièrement adaptés à l'analyse topologique et structurelle des réseaux. En général, on observe qu'un réseau réel, quelle que soit son origine, est

caractérisé par une densité faible (la probabilité est faible pour que deux nœuds choisis au hasard soient reliés), une distance moyenne faible (il existe en moyenne un chemin court entre n'importe quelle paire de nœuds), une distribution des degrés hétérogène (il existe un nombre non négligeable de nœuds qui ont un fort degré alors que la grande majorité des autres nœuds ont un degré très faible) et une densité locale forte (il est plus probable qu'un nouveau lien apparaisse entre deux nœuds qui sont proches que entre deux nœuds éloignés). Ces différentes propriétés constituent un ensemble de référence, considéré comme fondamental pour l'étude des réseaux réels.

Il existe de nombreuses problématiques dans ce domaine, par exemple la question de la propagation d'épidémie ou de virus informatique, la fragilité du réseau en cas de panne, sa résilience en cas d'attaque, l'étude de la dynamique, la prédiction de l'apparition de nouveaux liens, la recommandation, etc. L'un des problèmes complexes actuels, qui a beaucoup d'applications, est l'identification de la *structure communautaire*. Du fait de la faible densité globale des réseaux réels couplée à leur forte densité locale, on observe la présence de groupes de nœuds fortement liés entre eux et plus faiblement liés avec le reste du réseau, que l'on appelle *communautés*. Ces structures ont également du sens dans le réseau lui-même, par exemple les communautés d'un réseau social peuvent correspondre à des groupes sociaux (amis, familles, etc.), les communautés d'un réseau de protéines peuvent traduire des réponses fonctionnelles, elles peuvent correspondre à des sujets similaires dans un réseau de pages Web, pour donner quelques exemples.

La détection de communautés est un problème difficile à résoudre et il n'existe à ce jour aucune définition formelle. En effet, un très grand nombre de méthodes heuristiques ont été proposées et sont pour la plupart développées pour les réseaux unipartis, c'est-à-dire pour des réseaux constitués d'un seul type de nœuds et de liens. Cependant, il existe en pratique de nombreux systèmes constitués de plusieurs types d'entités et de plusieurs types de relations. L'approche traditionnelle pour modéliser de tels systèmes consiste à ne pas faire la distinction entre les différentes entités ou relations, et donc d'aboutir à une version plate du réseau. En revanche, cette approche s'avère souvent trop contraignante et inadaptée à la détection de communautés. Prenons un exemple emprunté à la sociologie qui permet de comprendre ces limitations. Un réseau social peut être décrit comme un ensemble de personnes qui interagissent entre elles par le biais de différentes relations. À première vue, il semble naturel de supposer que toutes ces connexions sont au même niveau. Cependant, les vraies relations entre les membres d'un réseau social dépendent en partie des groupes sociaux auxquels ils appartiennent, et il est peu pertinent de les considérer toutes identiques. Considérons le réseau social de Facebook représenté par un réseau où tous les membres sont vus comme des nœuds et les amitiés comme des liens. En réalité, deux membres de Facebook peuvent être « amis » soit parce qu'ils sont collègues de travail, dans la même équipe de sport, de la même ville, ou pour n'importe quelle

autre raison sociale. Maintenant, supposons qu'un utilisateur décide de propager à ses amis une information, ou une rumeur par exemple. Il est évident que ce dernier va choisir les destinataires en fonction du contenu et des intérêts de chacun. Le sous-groupe d'amis sélectionnés peut être considéré comme une communauté car il regroupe des personnes partageant les mêmes centres d'intérêts. Par conséquent, si la modélisation ne prend pas en compte les différences dans les relations alors la vraie structure sociale du réseau ne peut pas être correctement comprise. Une représentation plus réelle serait de modéliser ce réseau par un graphe multiplexe composé d'autant de niveaux que de type de relations sociales entre les personnes.

Dans cette thèse, nous nous intéressons à la structure communautaire des réseaux réels bipartis. Un réseau biparti est un type de réseau constitué d'un seul ensemble de liens connectant uniquement deux types de nœuds, comme par exemple un réseau de co-publications scientifiques dans lequel les auteurs sont connectés uniquement à leurs publications. Les auteurs ne sont donc pas connectés directement, ni les publications. L'objectif que nous nous sommes fixé est alors d'analyser les propriétés structurelles de ces réseaux afin de proposer de nouvelles méthodes pour la détection de communautés. Le mémoire est organisé comme suit :

Le Chapitre 2 donne une vue d'ensemble des définitions liées au formalisme des graphes, notamment celles des graphes unipartis et des graphes bipartis, ainsi que des concepts importants qui seront utilisés tout au long du mémoire. De plus, une section est réservée aux problématiques majeures rencontrées dans l'étude des réseaux réels et détaille formellement les propriétés de ces réseaux.

Le Chapitre 3 est dédié à la structure communautaire des réseaux réels. Nous faisons l'état de l'art des méthodes de détection de communautés les plus importantes. Afin de se situer par rapport aux nombreux travaux existants dans le domaine, nous tentons de catégoriser les principales méthodes existantes en fonction de leurs approches au problème. Pour cela, trois grandes familles d'approches sont utilisées : les approches centrées réseau, les approches centrées groupe, les approches centrées propagation. Cette classification couvre à la fois les méthodes classiques pour les graphes unipartis et les méthodes pour les graphes bipartis. Nous donnons aussi souvent que possible les limites et inconvénients des méthodes ou algorithmes. Afin de clore ce chapitre, nous présentons également comment la performance et la qualité des méthodes de détection de communautés sont généralement évaluées et confrontées.

Le Chapitre 4 traite de l'analyse de la propriété de recouvrement dans la structure communautaire bipartie des réseaux sociaux du Web. Afin de réaliser ce travail, nous nous appuyons sur des métriques classiques, le coefficient de clustering et le coefficient de redondance, et proposons deux nouvelles métriques, la dispersion et le monopole.

Le Chapitre 5 propose une nouvelle méthode de détection de communautés pour les graphes bipartis, appelée ComSim, qui se base sur la similarité entre les nœuds et la recherche de cycle. Nous confrontons cette nouvelle technique avec les algorithmes standards de la littérature et mettons en perspective les caractéristiques des communautés détectées ainsi.

Le Chapitre 6 s'intéresse à la détection de communautés dans les réseaux multi-couches. Un *réseau multi-couches* est composé de plusieurs couches, chacune composée de nœuds connectés entre eux mais aussi connectés vers d'autres couches. Chaque couche prise séparément peut être vue comme un graphe uniparti, tandis que les liens entre deux couches constituent un graphe biparti. Nous proposons une généralisation de l'indice classique de modularité pour évaluer deux types de communautés dans ce contexte, les communautés mono-couches et les communautés multi-couches.

Chapitre 2

Notions générales pour les réseaux réels

Sommaire

2.1 Propriétés des graphes 6

2.1.1 Graphes unipartis 6

2.1.2 Graphes bipartis 12

2.2 Réseaux réels 15

2.2.1 Principales propriétés 16

2.2.2 Modèles aléatoires 17

2.1 Propriétés des graphes

2.1.1 Graphes unipartis

La plupart des réseaux rencontrés en pratique peuvent être représentés naturellement par des graphes. En effet, la théorie des graphes offre un ensemble d'outils mathématiques adaptés à l'analyse des réseaux, ceci à travers de nombreux modèles et définitions qui permettent de décrire et de résoudre des problèmes complexes. Par exemple, on peut utiliser un *graphe uniparti* pour représenter un réseau d'acteurs, dans lequel les acteurs sont reliés entre eux s'ils jouent dans un même film. Ce formalisme permet d'étudier finement les interactions entre acteurs et de mettre en perspective les propriétés et caractéristiques de ce réseau. Ainsi, il est possible de mieux comprendre les relations sociales sous-jacentes (relations d'amitiés, professionnelles, familiales) ou des mécanismes plus globaux, comme par exemple comment le réseau s'est formé, quels sont les acteurs phares, quels sont les groupes ou coalitions d'acteurs, existe-t-il une forme de collaboration visible dans la structure du réseau, etc.

2.1.1.1 Notations

Un graphe[1] G défini par $G = (V, E)$ est constitué de deux ensembles, où $V \neq \emptyset$ est un ensemble d'éléments appelés *sommets*, et E est un ensemble de couples d'éléments, appelées *arcs*[2]. On note $V = \{v_1, v_2, ..., v_n\}$ où $|V| = n$ est le nombre de sommets, et $E = \{e_1, e_2, ..., e_m\}$ où $|E| = m$ est le nombre d'arcs. Un arc e_l est un couple de sommets tel que $e_l = (v_i, v_j)$, reliant le sommet v_i au sommet v_j. On distingue deux types de graphes selon si l'ensemble des arcs E est symétrique ou asymétrique. Lorsque E est une relation asymétrique sur V, on dit que le graphe est *orienté*. À l'inverse, lorsque E est une relation symétrique sur V alors on parle de graphe *non-orienté*. Dans ce cas, un arc $(v_i, v_j) \in E$ est noté $\{v_i, v_j\}$ et que l'on appelle *arête*.
Il peut également exister des poids sur les arcs, on définit alors $G = (V, E, W)$ un *graphe pondéré*, où $W : V \times V \mapsto \mathbb{R}$ est une fonction qui associe un poids à chaque arc. Un exemple de ces différents types de graphe est donné Figure 2.1. Plus généralement, un graphe peut aussi avoir des arcs multiples entre deux sommets et des boucles sur un sommet. Lorsqu'il n'existe aucun arc multiple ni boucle alors le graphe est dit *graphe simple*.

[1] Par soucis de simplicité pour la suite, lorsqu'on parle d'un graphe, on sous-entend un graphe non-orienté et simple, sauf précision.

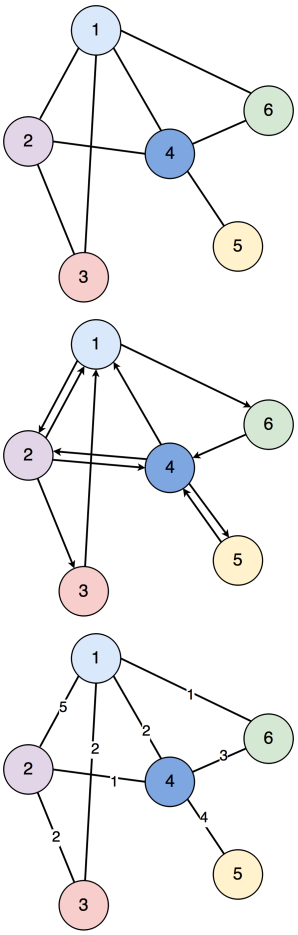
[2] Plus généralement, dans la terminologie lorsqu'on parle de *réseau*, un sommet est un *nœud* et un arc est un *lien*.

On fournit souvent une représentation graphique d'un graphe en dessinant les sommets par des points ou des petits cercles et les arêtes par des traits entre ces sommets afin de montrer leurs adjacences. Dans le cas d'un graphe orienté, les arcs sont des flèches indiquant la direction partant du sommet d'origine au sommet de destination. Si le graphe est pondéré, alors le poids est inscrit à côté de l'arête concernée. Lorsque la taille du graphe est petite, c'est-à-dire qu'il n'y a pas beaucoup de sommets, dessiner le graphe peut être utile. Cependant, une telle représentation graphique peut s'avérer rapidement limitée, notamment lorsque le graphe est grand. Dans ce cas, il est préférable d'utiliser une représentation formelle afin rendre compte des propriétés du graphe, qui sont le plus souvent non triviales. On peut par exemple représenter un graphe par une matrice d'adjacence. Ceci est également utile pour mener des calculs analytiques et permet de stocker efficacement le graphe dans la mémoire d'un ordinateur. Pour un graphe simple ou orienté, la *matrice d'adjacence* $A = [a_{ij}]$ est une matrice carrée $n \times n$ telle que :

$$a_{ij} = \begin{cases} 1 & \text{si } (v_i, v_j) \in E \\ 0 & \text{sinon.} \end{cases}$$

Pour un graphe pondéré, la matrice d'adjacence A est définie par :

$$a_{ij} = \begin{cases} W(v_i, v_j) & \text{si } (v_i, v_j) \in E \\ 0 & \text{sinon.} \end{cases}$$



$$A = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & 5 & 2 & 2 & 0 & 1 \\ 5 & 0 & 2 & 1 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 4 & 3 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 1 & 0 & 0 & 3 & 0 & 0 \end{pmatrix}$$

FIGURE 2.1: Représentation graphique d'un graphe non-orienté (haut), orienté (milieu) et pondéré (bas) et de leur matrice d'adjacence A associée.

2.1.1.2 Voisinage d'un sommet

Le *voisinage* d'un sommet u d'un graphe G, noté $N(u)$, est l'ensemble des voisins de u, c'est-à-dire reliés à u par une arête. Formellement :

$$N(u) = \{v \in V | \{u, v\} \in E\} \tag{2.1}$$

Si on considère un graphe orienté, le voisinage est défini par :

- le voisinage entrant : $N^{in}(u) = \{v \in V | (u, v) \in E\}$
- le voisinage sortant : $N^{out}(u) = \{v \in V | (v, u) \in E\}$

Le voisinage d'un sommet u peut être vu comme un *sous-graphe* de G constitué de l'ensemble des sommets voisins de u et des arêtes qui les relient. On dit alors que ce sous-graphe est induit par le voisinage de u.

2.1.1.3 Degré d'un sommet

Le *degré* d'un sommet u, noté $d(u)$, désigne le nombre d'arêtes connectées à u. Formellement, le degré est défini par $d(u) = |N(u)|$. Dans un graphe pondéré, le degré pondéré d'un sommet est alors la somme du poids des arêtes incidentes à ce sommet, soit formellement $d(u) = \sum_{v \in N(u)} W(u, v)$.

Si on considère un graphe orienté, trois façons différentes permettent de décrire le degré d'un sommet u :

- le degré entrant : $d^{in}(u) = |N^{in}(u)|$, qui est le nombre d'arcs allant vers u
- le degré sortant : $d^{out}(u) = |N^{out}(u)|$, qui est le nombre d'arcs sortant de u
- le degré total : $d(u) = d^{in}(u) + d^{out}(u)$, qui est le nombre total d'arcs entrant et sortant de u

Il est donc possible de mesurer le *degré moyen* d'un graphe. Pour un graphe non-orienté et pondéré, on note :

$$d(G) = \frac{1}{n} \sum_{u \in V} d(u) = \frac{2m}{n} \tag{2.2}$$

Pour un graphe orienté, $d^{in}(G) = d^{out}(G) = \frac{m}{n}$.

Le nombre d'arêtes est aussi étroitement lié à la densité, plus le degré moyen est grand plus la densité d'un graphe sera grande. La *densité* est le nombre d'arêtes existantes dans G par rapport au nombre total possible d'arêtes dans G. Pour un graphe non-orienté ou pondéré :

$$\delta(G) = \frac{2m}{n(n-1)} \tag{2.3}$$

Pour un graphe orienté, $\delta(G) = \frac{m}{n(n-1)}$.

2.1.1.4 Concept de chemin dans les graphes

Un *chemin* dans un graphe $G = (V, E)$ est une suite de sommets $(v_1, v_2, \ldots, v_p)$ de V de longueur $p - 1$ telle que deux sommets consécutifs de cette suite v_i et v_{i+1} sont reliés par une arête $\{v_i, v_{i+1}\}$. De la même manière, un chemin peut être décrit par une suite d'arêtes $(e_1, e_2, \ldots, e_{p-1})$ de E. Un chemin est dit :

- *simple*, si il ne passe pas deux fois par la même arête,
- *élémentaire*, si il ne passe pas deux fois par le même sommet.

La *distance* entre deux sommets dans un graphe est déterminée par la longueur du chemin le plus court entre ces deux sommets. Le *diamètre* dans un graphe est alors la plus grande distance possible existante entre deux sommets.

Un *cycle* est aussi un chemin mais dont les extrémités sont identiques.

2.1.1.5 Clustering

Le *clustering* est une notion qui décrit la propension d'un graphe à former des triangles, c'est-à-dire des sous-graphes complets de taille 3, ou cycles élémentaires de longueur 3, noté C_3. Le clustering est une manière de mesurer la *transitivité* globale d'un graphe.

En effet, par transitivité s'il existe une arête $\{u, v\}$ et une arête $\{v, w\}$, alors la probabilité qu'une arête $\{u, w\}$ existe est renforcée. Ceci est comparable au fait que deux personnes ont plus de chance de se connaître si elles ont un ami en commun [New03b], d'où l'expression populaire *les ami(e)s de mes ami(e)s sont mes ami(e)s*. Cette transitivité globale est calculée par le ratio de transitivité :

$$TR = \frac{3 \times \text{nombre de triangles dans G}}{\text{nombre de triplets connectés}} \tag{2.4}$$

Un triplet connecté est un sous-graphe avec au moins deux arêtes entre trois sommets.

D'un point de vue local, il est possible de compter le nombre de triangles aux abords du voisinage d'un nœud u. Le *coefficient de clustering* local d'un nœud u est donc défini par :

$$CC_3(u) = \frac{2 \cdot |\{\{v, w\} \in E | v \in N(u) \text{ et } w \in N(u), v \neq w\}|}{d(u) \cdot (d(u) - 1)} \tag{2.5}$$

Finalement, ces expressions expriment le ratio entre le nombre d'arêtes reliant les voisins de u et le nombre maximum d'arêtes possibles dans le voisinage de u. Si $G' = (V', E')$ est le graphe induit par le voisinage de u alors plus $CC_3(u)$ est proche de 1, plus G' est proche d'un graphe complet ou d'une clique. $CC_3(u)$, compris entre 0 et 1, est donc la densité du graphe induit.

Parallèlement, il est possible d'obtenir un coefficient de clustering global qui est en fait la moyenne des coefficients de clustering locaux des sommets du graphe :

$$CC_3(G) = \frac{1}{n} \sum_{u \in V} CC_3(u) \tag{2.6}$$

2.1.1.6 Composantes connexes

Une *composante connexe* d'un graphe G est un sous-graphe $Comp = (V', E')$ tel que $V' \subseteq V$ et $E' \subseteq E \cap (V' \times V')$ dans lequel, pour tous sommets u, v de $Comp$, il existe un chemin entre u et v (avec $u \neq v$).

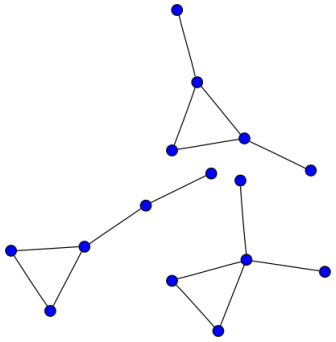


FIGURE 2.2: Graphe non connexe avec 3 composantes connexes.

2.1.1.7 Clique et clique maximale

Une *clique* de taille k est définie comme un graphe (ou sous-graphe) dans lequel les k sommets sont connectés les uns aux autres, c'est-à-dire un graphe complet (sous-graphe complet) de taille k. Une *clique maximale* est une clique de taille k non incluse dans une clique de taille $k+1$. Un triangle est par exemple une clique de taille 3.

2.1.1.8 Assortativité

L'assortativité est la tendance des sommets d'un graphe à partager les mêmes propriétés. Bien qu'il existe un nombre important de mesures de similarité (voir Section 3.2.1.2) capable de mettre en évidence les propriétés communes entre les sommets, on parle le plus souvent d'assortativité à travers la notion de degrés ou le nombre de voisins communs. Ainsi, dans un réseau à caractère assortatif, très répandu dans les réseaux sociaux [NP03, RCC+04], les nœuds possédant le même degré auront tendance à être reliés ensemble. À l'inverse, dans un réseau non-assortatif (ou disassortatif), un réseau biologique ou technologique, les noeuds de faible degré auront tendance à être reliés aux noeuds de fort degré (voir attachement préférentiel [BA99]).

2.1.2 Graphes bipartis

Une hypothèse récente suggère que la plupart des réseaux réels, représentés par un graphe uniparti, sont le résultat de la projection d'un réseau biparti [GL04]. Par exemple un réseau d'acteurs où les acteurs sont connectés entre eux lorsqu'ils jouent dans un même film. L'information contenue dans ce réseau réside dans les nœuds (les acteurs) et dans les liens (les films mettant en relation les acteurs). Néanmoins, l'utilisation d'un graphe biparti avec deux ensembles distincts de films et d'acteurs semble être plus représentatif de la réalité [WS98] et permet de mieux modéliser les propriétés intrinsèques au réseau [GL04].

Dans de nombreux contextes, les réseaux présentent naturellement une structure bipartie [GL04], dont la particularité est d'être constituée d'interactions entre deux types d'entités différentes, et seulement entre ces deux types d'entités. Par exemple, dans un réseau biologique composé de gènes et de protéines, une relation indiquant que tel gène est exprimé dans telle protéine [JTA+00]. En sciences sociales, avec des réseaux de collaboration scientifique représentant des auteurs et les articles qu'ils ont publiés [GL04, New01, WS98]. D'autres exemples existent, parmi les réseaux sociaux, les réseaux du web (Wikipédia, LiveJournal, Flickr, Twitter), les réseaux de films-acteurs, en linguistique avec des réseaux décrivant les liens entre langage et grammaire [iCS01], en droit avec des réseaux de décisions juridiques [TNG14]. En définitif, dès lors qu'il existe des interactions entre deux ensembles d'objets, il est possible d'utiliser un graphe biparti.

Formellement un graphe biparti est un graphe dans lequel il est possible de séparer les sommets en deux ensembles tel que les arêtes relient uniquement les sommets d'un ensemble à ceux de l'autre. Un graphe biparti est défini par un triplet $\mathbb{B} = (\top, \bot, E_B)$ où $\top$ est l'ensemble des sommets du haut[3], $\bot$ est l'ensemble des sommets du bas et $E_B \subseteq (\top \times \bot)$ est l'ensemble des arêtes reliant $\top$ et $\bot$, avec $m = |E_B|$ le nombre d'arêtes. Le nombre total de sommets dans $\mathbb{B}$ est $n = n_\top + n_\bot$, où $n_\top = |\top|$ est le nombre de sommets de $\top$ et $n_\bot = |\bot|$ est le nombre de sommets de $\bot$.

[3] Les termes *haut* et *bas* sont utilisés par convention afin de repérer visuellement l'ensemble des sommets concernés.

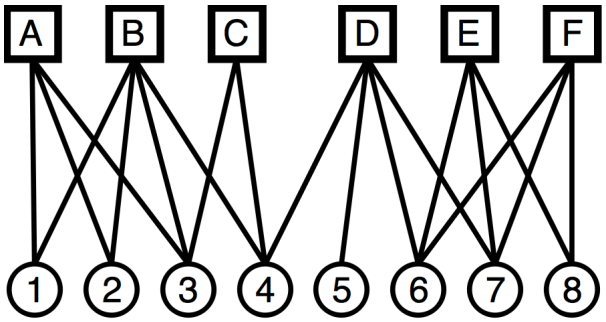


FIGURE 2.3: Exemple d'un graphe biparti $\mathbb{B} = (\top, \bot, E_B)$ avec $\top = \{A, B, C, D, E, F\}$ et $\bot = \{1, 2, 3, 4, 5, 6, 7, 8\}$.

Soit $u \in \top$, le voisinage de u est donc défini comme les sommets de $\bot$ en lien avec u, écrit formellement $N_\top(u) = \{v \in \bot | \{u, v\} \in E_b\}$. $N_\bot(u)$ est défini de la même manière pour l'ensemble des voisins d'un sommet $u \in \bot$. Le degré d'un sommet $u \in \top$ est $d_\top(u) = |N_\top(u)|$ (respectivement pour le degré d'un sommet $u \in \bot$ est $d_\bot(u) = |N_\bot(u)|$). Si l'on souhaite capturer l'ensemble des voisins à distance 2 de u, on note $N^2_\top(u) = \{w \in \top | u \neq w, \exists v \in \bot, \{u, v\} \in E_B$ et $\{w, v\} \in E_B\}$ et $|N^2_\top(u)|$ le nombre de voisins à distance 2.

Le degré moyen pour l'ensemble $\top$ est $d_\top(\mathbb{B}) = \frac{m}{n_\top}$, le degré moyen pour l'ensemble $\bot$ est $d_\bot(\mathbb{B}) = \frac{m}{n_\bot}$ et le degré moyen des deux ensembles est $d(\mathbb{B}) = \frac{2m}{n_\top + n_\bot}$. Finalement, la densité d'un graphe biparti est définie comme $\delta(\mathbb{B}) = \frac{m}{n_\top \times n_\bot}$.

Il peut arriver qu'un ensemble de nœuds du réseau biparti, l'ensemble $\top$, a plus d'importance pour un but particulier que le second, l'ensemble $\bot$. Dans ce cas, il est possible de projeter $\mathbb{B}$ sur l'ensemble $\top$ pour produire un graphe uniparti G à l'aide des relations biparties. Par exemple, cela est utile pour pouvoir utiliser les outils et méthodes déjà disponibles pour les graphes unipartis, comme la recherche de cliques ou de groupes cohésifs appelés *communautés* (voir Section 3). Cependant, il n'est pas toujours recommandé d'utiliser la projection pour plusieurs raisons pratiques. Premièrement, les projections amènent nécessairement à créer un graphe composé de cliques recouvrantes (voir Chapitre 4) qui rend la détection de communautés plus difficile, particulièrement pour les méthodes utilisant un modèle aléatoire (voir Section 3.2.1.1). D'autre part les mesures classiques telles que le coefficient de clustering ou l'assortativité se retrouvent surestimées. Il y a aussi perte d'information car différents graphes bipartis peuvent produire une projection identique [ZRMZ07]. Toutefois, il est possible d'utiliser une projection pondérée sur la matrice d'adjacence A du graphe biparti afin d'éviter ce problème [EB13, ZRMZ07].

2.1.2.1 Clustering biparti

Le concept de clustering existe également pour un graphe biparti, il mesure à quel point le voisinage d'un sommet est connecté. L'expression diffère de celle pour un graphe uniparti (voir Section 2.1.1.5) car, par définition, il n'existe pas de triangle dans un graphe biparti. Ainsi, plusieurs propositions ont été faites [Ops13, ZWL+08, LGH05, RA04] qui s'intéressent à capturer des *rectangles* ou des *cycles* de longueur 4. Autrement dit, un rectangle est formé lorsque un sommet u a deux voisins distincts v_1 et v_2 ayant en commun un sommet w ($u \neq w$). Le coefficient de clustering local proposé par [ZWL+08] est :

$$CC_{4,v_1v_2}(u) = \frac{q_{uv_1v_2}}{(d_\perp(v_1) - \eta_{uv_1v_2}) + (d_\perp(v_2) - \eta_{uv_1v_2}) + q_{uv_1v_2}} \tag{2.7}$$

où $q_{uv_1v_2}$ est le nombre de rectangles contenant les trois sommets u, v_1 et v_2. $\eta_{uv_1v_2} = 1 + q_{uv_1v_2}$.

Une autre définition se base sur l'*indice de Jaccard* qui mesure la similarité entre deux voisinages, ou plus largement entre deux ensembles de sommets. L'indice de Jaccard divise le cardinal de l'intersection des ensembles considérés par le cardinal de l'union des ensembles considérés :

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.8}$$

avec $0 \leq J(A, B) \leq 1$: 0 si les deux ensembles n'ont aucun élément en commun, 1 si les deux ensembles sont identiques.

Le coefficient de clustering local proposé par [LMDV08] calcule la moyenne des indices de Jaccard entre les voisins d'un sommet u et les voisins d'un sommet v pour chaque $v \in N^2_\top(u)$:

$$CC_4(u) = \frac{1}{|N^2_\top(u)|} \sum_{v \in N^2_\top(u)} J(N_\top(u), N_\top(v)) \tag{2.9}$$

Une autre mesure est également proposée par [LMDV08] afin d'évaluer le clustering d'un graphe biparti. Cette dernière s'intéresse à capturer la *redondance* qui mesure à quel point les voisins d'un sommet u ont tendance à être connectés aux mêmes voisins autre que u. Le *coefficient de redondance* est noté :

$$RD(u) = \frac{|\{(v_1, v_2) \in (N_\top(u) \times N_\top(u)) | v_1 \neq v_2 \text{ et } \exists w \in \top, (w, v_1) \in E_B \text{ et } (w, v_2) \in E_B\}|}{d_\top(u) \cdot (d_\top(u) - 1)/2} \tag{2.10}$$

En d'autres termes, le coefficient de redondance d'un sommet $v_i \in \top$ est la proportion de paires de voisins de u reliées à d'autres sommets que u.

2.1.2.2 Biclique

Une *biclique* est une clique dans le contexte d'un graphe biparti. Une biclique est donc un graphe biparti complet (ou sous-graphe biparti complet) notée $K_{a,b}$, avec $a = n_\top$ et $b = n_\bot$, dans lequel tous les sommets d'un ensemble sont connectés à tous les sommets de l'autre ensemble (voir Figure 2.4).

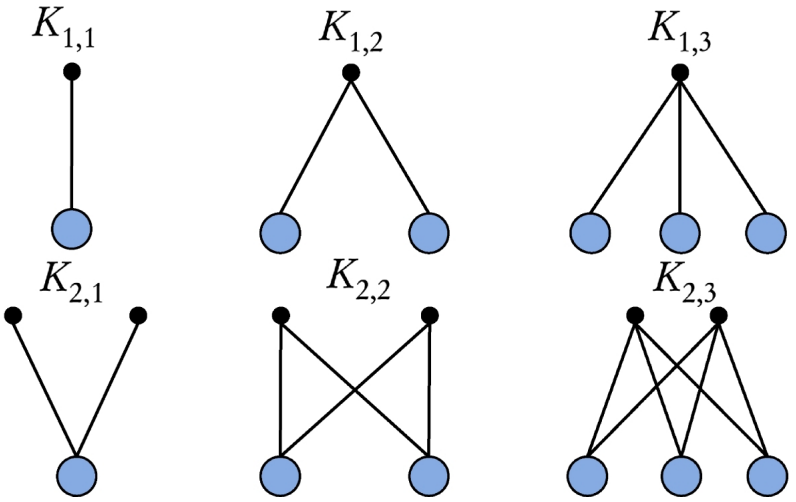


FIGURE 2.4: Exemples de bicliques.

2.2 Réseaux réels

Beaucoup de systèmes rencontrés en pratiques sont composés d'un grand nombre d'entités qui interagissent de manière complexe. Les mécanismes de ces systèmes complexes ne dépendent pas simplement des éléments de base mais sont liés à la présence de niveaux d'organisation et de propriétés locales et globales émergentes. Que ce soit en sciences sociales, en biologie, en économie ou en technologie, les systèmes complexes font l'objet de nombreuses études. Plus particulièrement, l'attention est portée sur l'étude de la topologie des réseaux réels, dits *réseaux complexes* ou *graphes de terrain*. Un réseau réel se définit également par opposition à un réseau généré ou artificiel qui ne correspond à aucune donnée du monde réel.

La recherche sur les réseaux réels a comme objectifs majeurs de :

1. identifier les propriétés topologiques décrivant les mécanismes des réseaux réels,

2. proposer des modèles réalistes permettant de simuler le fonctionnement (au niveau structurel et de la dynamique) de ces réseaux.

Un axe important de recherche concerne l'étude de la structure topologique des réseaux réels. Cette dernière permet d'adresser un certain nombre de problèmes liés aux mécanismes de formation des systèmes complexes, aux mécanismes qui régissent les relations entre entités dont certaines cachent des fonctions spécifiques. Par exemple, pour le réseau d'Internet constitué de routeurs, comment améliorer le routage du trafic ? Quelle est la résistance du réseau face à des attaques ? Comment un virus se propage-t-il sur Internet ? Quelle est la robustesse du réseau en cas de défaillance ? Existe-t-il des chemins plus sûrs, plus rapides pour acheminer l'information ?

2.2.1 Principales propriétés

Les réseaux réels se prêtent naturellement à la formalisation sous forme de graphe permettant d'identifier des propriétés structurelles non-triviales. À partir de l'analyse des propriétés, on peut ainsi distinguer les réseaux les uns des autres et les caractériser. Certaines de ces propriétés sont communes à plusieurs réseaux provenant de différents contextes ; les réseaux d'interactions entre protéines en biologie, les réseaux technologiques (Internet, le Web), les réseaux de transport, etc. Plus généralement, c'est le cas des *réseaux réels* qui partagent des propriétés communes :

Une faible densité Globalement, on observe dans les réseaux réels que le nombre de liens a le même ordre de grandeur que le nombre de nœuds et donc que la densité globale du graphe est faible ($\delta \approx 0$) (voir Section 2.1.1.3).

La forte densité locale À l'inverse, la plupart des réseaux réels ont une densité locale élevée, calculée par le coefficient de clustering (voir Section 2.1.1.5 et Section 2.1.2.1).

Composante connexe géante Dans les réseaux réels, on s'aperçoit que la taille des composantes connexes (voir Section 2.1.1.6) est très inégale et qu'une composante connexe géante regroupe la majorité des nœuds.

Invariant d'échelle On observe souvent que certains nœuds ont des degrés très élevés et que d'autres ont des degrés très faibles. Ce contraste entre les degrés des sommets du graphe est analysé à l'aide d'une distribution, on parle alors d'une *distribution hétérogène* des degrés. La distribution des degrés est une répartition ordonnée de degrés des nœuds en

donnant leurs fréquences, c'est-à-dire qu'on comptabilise par ordre croissant la fréquence d'apparition de chaque degré. Une distribution hétérogène est donc une distribution où l'on observe une répartition très inégale des fréquences d'apparition. De plus, la fonction de distribution des degrés est souvent proche d'une loi de puissance de la forme $P(k) = k^{-\gamma}$ où $P(k)$ est la fraction des sommets ayant un degré k et γ est un paramètre (dont la valeur est comprise entre 2 et 3 dans la plupart des cas). Cette loi de puissance détermine une classe de réseaux réels dits *invariants d'échelle* ou *réseaux sans échelle*.

L'effet petit-monde Cette propriété est mise en évidence historiquement par l'expérience du sociologue Stanley Milgram [TM67]. Le résultat de l'expérience montre qu'il existe une courte chaîne de relations sociales reliant chacun à n'importe qui, suggérant en moyenne un degré de séparation proche de 6. Cependant, l'expérience est très controversée par rapport aux hypothèses de départ qui se révèlent peu représentatives voire fausses. L'effet petit-monde est tout de même observé dans la majorité des réseaux réels, mais avec des degrés de séparation différents. Plus de trente ans plus tard, ce phénomène est modélisé par [WS98] qui fixe la distance moyenne (ou degré de séparation) entre deux nœuds comme étant proportionnelle au logarithme du nombre de nœuds dans la réseau.

Une autre propriété très importante des réseaux réels est la *structure communautaire* (voir Chapitre 3).

2.2.2 Modèles aléatoires

Un modèle aléatoire s'intéresse à reproduire les phénomènes observés des réseaux rencontrés dans la nature. Être capable de modéliser les propriétés de ces réseaux est crucial pour mieux comprendre leurs mécanismes et mettre en lumière leurs fonctions. Un modèle aléatoire est généralement décrit par un ensemble de règles qui gouvernent la façon dont les nœuds se connectent les uns aux autres. Plus précisément, à travers un processus stochastique, un modèle aléatoire génère un graphe qui possède certaines propriétés que l'on désire proches des propriétés observées sur les réseaux réels.

Le modèle aléatoire le plus simple est proposé par Erdős et Rényi en 1959, connu sous le nom de *ER random graphs* [Bol98]. Dans ce modèle, noté $G_{n,p}$, on fixe le nombre de sommets n et une probabilité p qu'il existe une arête entre une paire quelconque de sommets. Autrement dit, les arêtes sont générées indépendamment les unes des autres, et plus p sera proche de 1 plus le graphe généré contiendra d'arêtes. Inversement, la probabilité de connecter m arêtes aléatoirement entre toutes les paires de sommets est :

$$p = \frac{m}{n(n-1)/2} \tag{2.11}$$

où $n(n-1)/2$ est le nombre total possible d'arêtes qu'il peut exister entre tous les sommets du graphe (voir Section 2.1.1.3). Néanmoins, ce modèle ne suffit pas à décrire correctement les propriétés existantes des réseaux réels. Afin d'y parvenir, d'autres modèles ont été développés, notamment pour modéliser les réseaux invariants d'échelles et l'attachement préférentiel [Yul25, BA99], l'effet petit monde et le clustering [WS98, New09], la structure communautaire (voir *modèle à blocs stochastiques* Section 3.2.1.2).

Nous présentons également le *modèle de configuration* [CL02, NWS02] qui est utilisé dans une des contributions du Chapitre 4. Le modèle de configuration CM génère un graphe avec le même nombre de sommets et d'arêtes mais les arêtes sont distribuées uniformément au hasard entre les sommets en conservant leur degré initial. De cette manière, la probabilité p qu'il existe une arête $\{u, v\}$ est fonction du degré de chacun des sommets adjacents, soit :

$$p[\{u, v\} \in E] = \frac{d(u) \cdot d(v)}{2m} \tag{2.12}$$

Chapitre 3

Structure communautaire

Sommaire

3.1 Types de communautés 21

3.1.1 Partitionnement 21

3.1.2 Recouvrement 21

3.1.3 Hiérarchie 22

3.2 Une classification des algorithmes de détection de communautés 23

3.2.1 Approches centrées réseau 24

3.2.2 Approches centrées groupes 32

3.2.3 Approches centrées propagation 33

3.3 Évaluation de la détection de communautés 37

3.3.1 Les vérités de terrain 37

3.3.2 La qualité d'une communauté 41

La structure communautaire est une propriété importante des réseaux réels. Elle fait référence à l'existence de *communautés*, une notion qui, à l'heure actuelle, n'a pas trouvé de définition unique. Toutefois, il existe un consensus informel qui s'accorde à dire qu'une communauté est un groupe de nœuds densément interconnectés et faiblement reliés au reste du réseau (voir Figure 3.1). Cette définition est basée sur l'idée que les nœuds d'un même groupe partagent les mêmes ressources ou les mêmes propriétés, et qu'il existe donc une certaine assortativité dans le réseau. Dans ce contexte, la problématique la plus répandue s'intéresse à la détection automatique de communautés, c'est-à-dire à la mise au point d'algorithmes efficaces capables de révéler la structure communautaire des réseaux réels. Un nombre important d'algorithmes de détection de communautés (aussi appelés *algorithmes de clustering*) ont été développés et aujourd'hui encore le domaine fait l'objet d'intenses recherches [For10]. Il est possible de trouver dans la littérature, et selon les disciplines, d'autres termes interchangeables avec communauté comme par exemple *cluster* ou *module*.

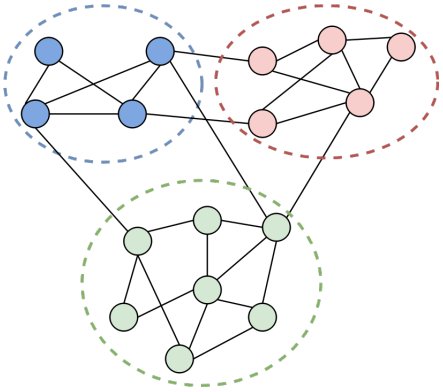


FIGURE 3.1: Partition d'un graphe en 3 communautés.

L'identification des communautés dans un réseau peut être un problème NP-complet selon la définition adoptée, difficile à résoudre en un temps polynomial. En effet, si on cherche à diviser les sommets d'un graphe en plusieurs groupes qui ne se chevauchent pas, on ne connaît a priori ni le nombre de ces groupes, ni leur taille. L'énumération de l'ensemble des solutions pour ce problème est alors de l'ordre de B_n, donné par le nombre de Bell, qui est exponentiel en nombre de sommets et donc irréalisable pour un algorithme qui les traiterait toutes. En conséquence, beaucoup de méthodes se basent sur l'optimisation d'une mesure de qualité qui permet d'explorer efficacement cet espace de solutions très large. D'autres méthodes utilisent des heuristiques qui exploitent directement les propriétés structurelles des réseaux. Du fait du grand nombre d'algorithmes existants, une classification est proposée en Section 3.2. Néanmoins, on distingue plusieurs types de communautés.

3.1 Types de communautés

3.1.1 Partitionnement

Généralement, la détection de communautés consiste à diviser le réseau en communautés. Une partition indique alors que les nœuds du réseau sont regroupés dans des communautés non-recouvrantes : dans ce cas là, un nœud appartient toujours à une communauté et une seule. Ainsi, l'objectif est de trouver une partition des sommets qui respecte une ou plusieurs propriétés. Par exemple, on peut chercher à minimiser le nombre d'arêtes externes reliant les différents groupes ou bien diviser le graphe en k communautés de taille équilibrée. Néanmoins, la forte densité locale observée dans les réseaux réels est une propriété indispensable pour capturer la structure communautaire et développer des heuristiques pour le partitionnement de graphe. Dans ce contexte, une fonction très répandue pour évaluer la qualité d'une partition est la *modularité* [NG04], l'objectif étant de trouver une partition qui maximise ce critère. D'autres approches utilisent des méthodes de coupe minimale ou de *partitionnement spectral* dans le graphe. Une coupe correspond à la partition d'un ensemble de sommets en deux sous-ensembles grâce à une fonction objective qui cherche à rendre minimal le nombre d'arêtes entre ces deux sous-ensembles [FCMR08].

3.1.2 Recouvrement

Les méthodes ne tenant pas compte du recouvrement (chevauchement ou *overlapping* en anglais) des communautés peuvent mener à une compréhension incomplète du réseau [PDFV05]. La complexité de la structure communautaire dans le contexte des réseaux réels peut inclure des relations multiples (voir Chapitre 6). Ainsi, prendre en considération le chevauchement entre communautés est une représentation plus réaliste et moins restrictive car elle autorise les nœuds du réseau à faire partie d'une ou plusieurs communautés. En effet, dans un réseau social composé de personnes, il est naturel d'imaginer que les communautés formant des groupes d'amis puissent être recouvrantes : une communauté pour la famille, une communauté de collègues de travail, une communauté d'amis d'enfance, etc (voir exemple Figure 3.2). Manifestement, le chevauchement est une caractéristique présente dans de nombreux réseaux réels [KGMI+12] et a mené à de nombreuses études [PDFV05, LFK09, LAV10, ABFX08].

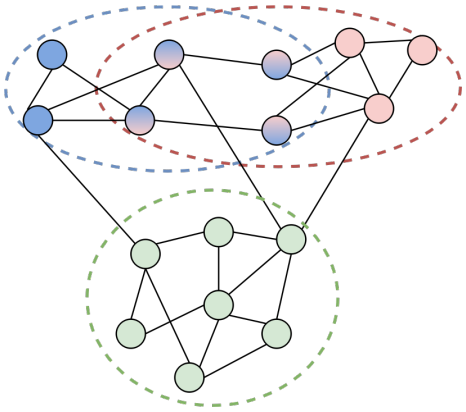


FIGURE 3.2: Recouvrement entre deux communautés (bleue et rouge).

3.1.3 Hiérarchie

Les réseaux réels présentent naturellement une hiérarchie de communautés [RSM+02]. La notion de *hiérarchie* indique qu'une communauté peut être divisée en plusieurs "sous-communautés". De nombreuses études sont consacrées à détecter la hiérarchie dans la structure communautaire [SPGMA07, NG04, SCCH09, BGLL08]. Certaines approches, comme l'algorithme de **Girvan-Newman** [GN02] identifient les communautés dans un premier temps puis cherchent à identifier les sous-communautés à partir des premières. D'autres approches, comme l'algorithme de **Louvain** [BGLL08] ou l'algorithme **EAGLE** [SCCH09], identifient d'abord les sous-communautés puis les assemblent entre elles.

D'autres méthodes plus directes, comme les méthodes de clustering hiérarchique [Joh67], s'intéressent à définir une *mesure de similarité* (voir Section 3.2.1.2) topologique entre paires de nœuds qui finalement produit une hiérarchie (représentée par un dendrogramme, voir exemple Figure 3.3). Finalement, identifier la hiérarchie dans la structure communautaire d'un réseau peut être indirectement lié au problème de recouvrement [SCCH09].

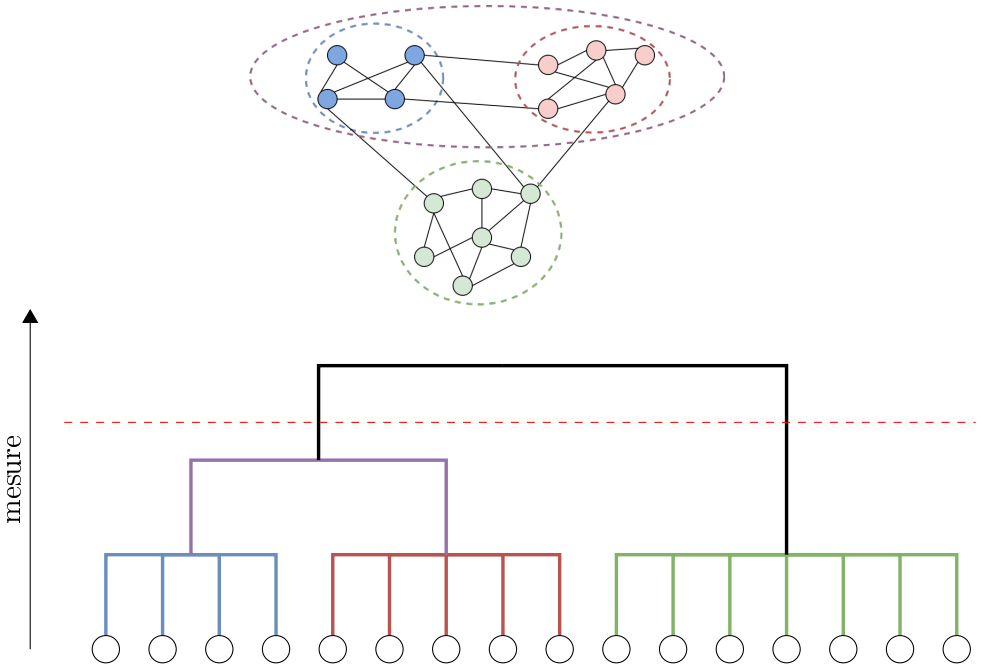


FIGURE 3.3: Un dendrogramme permet de représenter les différents niveaux d'une hiérarchie. La ligne rouge en pointillée indique le partitionnement optimal de la hiérarchie selon une mesure. Dans cet exemple, la meilleure coupe correspond aux communautés violette et verte.

3.2 Une classification des algorithmes de détection de communautés

La recherche de communautés est un sujet qui a été abondamment traité dans le domaine des réseaux complexes. Pendant cette dernière décennie, un nombre considérable d'articles scientifiques ont été produits sur ce sujet. Afin de dresser un tableau des différents algorithmes de détection de communautés, nous proposons une classification de ces derniers en tenant compte de leurs approches. D'un point de vue topologique, une communauté est un ensemble de nœuds plus fortement connectés entre eux qu'avec le reste du réseau. Cette définition a trouvé un consensus auprès de la communauté scientifique.

Il existe dans la littérature trois importantes études de synthèse [For10, PKVS12, TL10] dédiées aux méthodes de détection de communautés dans les graphes unipartis. De la même manière, nous tentons une classification des principales approches de ce domaine en faisant le lien avec les méthodes pour la détection de communautés dans les graphes bipartis. Manifestement, les graphes bipartis peuvent être traités comme des graphes unipartis en ignorant la partition $\top$ et $\bot$ des sommets ou en créant une projection sur l'un des ensembles. Cependant, il y a récemment un intérêt croissant à considérer les réseaux bipartis comme une classe de réseaux distincte et à analyser leur structure

communautaire. De façon générale, les méthodes développées pour les réseaux bipartis s'inspirent largement des méthodes pour les graphes unipartis, nous les classifions donc toutes autour de trois familles d'approches :

- *Approches centrées réseau*, où la structure globale du réseau est examinée pour la décomposition du graphe en communauté.
- *Approches centrées groupe*, où les nœuds du réseau sont regroupés en communautés en fonction de propriétés topologiques partagées. Nous nous focalisons dans cette thèse sur ce type d'approche. De plus, à la vue du nombre d'approches, méthodes et définitions heuristiques, nous proposons de considérer la détection de communautés comme étant plus largement *l'identification des propriétés communément partagées entre les entités d'un réseau* (voir Chapitre 6).
- *Approches centrées propagation*, où la structure communautaire est identifiée par un processus de diffusion de messages.

3.2.1 Approches centrées réseau

Les approches centrées réseau sont des méthodes qui s'appuient sur des calculs prenant en compte l'ensemble des connexions du graphe.

3.2.1.1 Approches d'optimisation

Le problème de la détection de communauté peut être rapporté à un problème d'optimisation en définissant une *fonction de qualité* (ou critère de qualité) que l'on cherche à minimiser ou maximiser. Une fonction de qualité peut également être utilisée pour évaluer la qualité d'une partition obtenue par différentes méthodes (voir Section 3.3.2).

La fonction de qualité la plus utilisée aujourd'hui est la *modularité* [NG04, New06b]. De façon informelle, la modularité d'une partition mesure la différence de liens internes aux communautés et la même quantité de liens si ces derniers étaient distribués aléatoirement dans le réseau en respectant les degrés des sommets. Pour un graphe uniparti, elle est notée formellement :

$$Q = \frac{1}{2m} \sum_{i,j} [A_{ij} - P_{ij}] \delta(c_i, c_j) \tag{3.1}$$

où A est la matrice d'adjacence du graphe, m le nombre total d'arêtes, c_i et c_j sont les communautés d'appartenance des sommets i et j, $\delta(c_i, c_j)$ est le delta de Kronecker qui vaut 1 si $c_i = c_j$ et sinon 0. P_{ij} décrit la probabilité d'existence de l'arête $\{i, j\}$ dans

le cas où les arêtes du graphe sont distribuées aléatoirement. P_{ij} est donc fonction du *modèle aléatoire* sous-jacent (aussi appelé *null model*). La valeur de la modularité varie de -0.5 à 1 selon si les liens tendent à être à l'extérieur ou à l'intérieur des communautés.

Le modèle aléatoire le plus utilisé dans ce contexte est le *modèle de configuration*. Celui-ci conserve la distribution des degrés du graphe, une importante propriété des réseaux réels. Autrement dit, ce modèle tient compte du même nombre de sommets et d'arêtes et reconnecte aléatoirement les sommets en respectant les degrés des sommets du graphe original. De cette manière, selon le modèle aléatoire, la modularité s'adapte à différents types de graphes en conservant leurs caractéristiques. Des recherches très intéressantes ont été menées afin de proposer d'autres modèles plus pertinents dans certains contextes [GdMC10, BPW+13], notamment avec les graphes bipartis [Bar07, GSPA07, Mur09, LWZZ16, XCL+15], les graphes multi-couches [PC16] (voir Chapitre 6).

De façon similaire, la modularité pour les graphes bipartis proposée par [Bar07] s'appuie sur une formulation proche de la modularité :

$$Q = \frac{1}{m} \sum_{i \in \top} \sum_{j \in \bot} [A_{ij} - P_{ij}] \delta(c_i, c_j) \tag{3.2}$$

où P_{ij} est le modèle aléatoire prenant en compte les contraintes d'un graphe biparti, il s'écrit simplement :

$$P_{ij} = \frac{d_\top(i) d_\bot(j)}{m}$$

Malheureusement, l'optimisation exacte de la modularité est un problème computationnellement difficile [BDG+06] et il est donc nécessaire d'utiliser des algorithmes d'approximation, *algorithmes gloutons* ou tout autre heuristique lorsque le réseau est de grande taille. Il est possible de regrouper ces derniers en 3 approches :

- Les méthodes d'optimisation directe utilisant des techniques d'algorithmes génétiques (uniparti [LS13], biparti [ZZGZ11, SABA14]) ou de recuit-simulé (uniparti [RB06]).
- Les méthodes agglomératives dans lesquelles on débute en considérant chaque sommet du graphe comme une communauté puis on fusionne deux communautés à chaque itération (uniparti [CNM04, PL05, BGLL08, PPDSLP14, SCCH09], biparti [Bar07]).
- Les méthodes divisives qui détectent les communautés en supprimant au fur et à mesure les sommets ou arêtes du graphe (uniparti [GN02, NG04, RCC+04, WH04]). Par exemple, le célèbre algorithme de *Girvan-Newman* [GN02] utilise une mesure

de centralité (*edge centrality*) afin d'isoler les communautés en supprimant du réseau les liens traversés par un grand nombre de *plus courts chemins* reliant les communautés entre elles.

Les méthodes agglomératives et divisives ont la particularité de produire des hiérarchies de communautés, qui peuvent être représentées par un dendrogramme (voir Figure 3.3). Parmi les articles cités ci-dessus, certains proposent des fonctions de qualité autres que la modularité [New06a, CNM04, WH04, PPDSLP14].

Actuellement, l'algorithme d'optimisation le plus performant est *la méthode de Louvain* [BGLL08]. Cette dernière utilise une approche agglomérative qui optimise localement la modularité. L'algorithme de **Louvain** procède en deux phases. Premièrement, lorsqu'un sommet est déplacé dans une communauté voisine, l'algorithme évalue le gain de ce déplacement, à savoir si la modularité est améliorée. Cette première phase est répétée pour chaque sommet du graphe et tant que la modularité globale est améliorée au vu des changements locaux. La deuxième phase compresse la partition obtenue en remplaçant chaque communauté par un seul sommet. Ainsi deux sommets v_i et v_j dans le nouveau graphe seront reliés si il existe une arête entre un sommet de la communauté v_i et un sommet de la communauté v_j. L'arête entre v_i et v_j est alors pondérée par le nombre d'arêtes entre ces deux communautés. Au vu de la performance des résultats fournis par l'algorithme, la complexité semble être fonction du nombre de sommets dans le réseau, soit $O(n \log n)$. Ainsi, Louvain est capable de traiter aisément des réseaux contenant plusieurs millions de nœuds et de liens. De plus, cet algorithme est compatible pour recevoir d'autres fonctions de qualité (ou mesures de similarité, voir Table 3.1) que la modularité, tant que ces dernières peuvent être formulées localement et globalement. Étonnement peu de travaux ont exploité cet aspect, cependant dans [CCG14] les auteurs proposent une grande variété de mesures qui à ce jour pourraient faire l'objet de nouveaux travaux.

L'optimisation de la modularité est sujet à une *limite de résolution* [FB07] liée à la taille des communautés pouvant être détectées. Il y a un risque de ne pas détecter la structure à petite échelle, c'est-à-dire les communautés structurellement bien définies dont la taille est petite comparée à la taille du réseau. Plus précisément, il a été montré que pour un réseau d'une taille donnée et d'une densité donnée, il existe une taille minimale de communautés détectables par une optimisation de la modularité. Autrement dit, plus le réseau a de nœuds et de liens, plus les communautés détectables seront grandes. Ceci indique donc que la valeur de la modularité pour deux petites communautés est inférieure à la valeur de la modularité si on fusionne ces deux communautés. En effet, le problème réside dans le modèle aléatoire qui suppose que chaque nœud peut interagir avec n'importe quel autre nœud (ce qui n'est pas le cas pour tous les réseaux réels [For10]).

Par conséquent, la probabilité qu'un lien existe entre deux groupes de nœuds décroît avec la taille du réseau. Or, si le réseau est suffisamment grand, cette probabilité intégrée dans la modularité devient négligeable et conduit donc à la fusion de ces deux groupes, même si ces derniers sont des sous-graphes complets. Plus généralement, la modularité n'est pas capable de détecter les communautés avec un nombre de liens internes d'ordre $\sqrt{m}$ ou inférieur [FB07].

3.2.1.2 Approches de clustering

Une approche simple pour la détection de communautés consiste à transformer ce problème en un problème de *clustering de données* qui correspond le plus souvent à définir un critère de proximité entre sommets, la *similarité*.

La similarité est déterminée par une mesure entre deux sommets (*similarité dyadique*) qui peut être définie soit localement (*mesure de similarité locale*), soit globalement (*mesure de similarité globale*), elle est notée $\theta(u, v)$. Les mesures de similarité sont en premier lieu développées pour les graphes unipartis, en voici quelques-unes d'entre elles Table 3.1. Les mesures de similarité sont naturellement transposables pour les graphes bipartis (voir la colonne de droite de la Table 3.1) car elles utilisent des notions comme le degré, le voisinage ou le chemin. Pour un graphe uniparti, l'approche classique revient donc à calculer une matrice de similarité S de dimension $n \times n$ où un élément S_{ij} exprime la similarité entre deux sommets v_i et v_j selon la mesure de similarité employée.

Pour un graphe biparti, trois approches sont possibles :

- La première approche consiste à ne considérer qu'un seul ensemble de sommets à la fois ($\top$ ou $\bot$). On peut soit utiliser la matrice d'adjacence A du graphe projeté ou bien calculer une matrice de similarité S de dimension $n_\top \times n_\top$ (respectivement $n_\bot \times n_\bot$) en utilisant par exemple une des mesures décrites Table 3.1. De cette manière, il est possible d'utiliser les méthodes classiques de clustering pour les graphes unipartis. Voici les différentes approches :
  - Le *regroupement hiérarchique* [Dhi01, ZHD+01, CDGS04]. Les méthodes de regroupement hiérarchiques ne se contentent pas de donner une partition d'un graphe mais cherchent une hiérarchie de partitions emboîtées, constituant un arbre binaire appelé dendrogramme (voir Figure 3.3). Le niveau le plus bas de la hiérarchie contient les sommets d'origine du graphe et les niveaux supérieurs regroupent successivement les sommets entre eux jusqu'à obtenir une hiérarchie complète, le dernier niveau étant la racine de l'arbre binaire. Les noeuds de l'arbre binaire sont donc des groupes de sommets du graphe qui sont

regroupés suivant une mesure de distance. Cette dernière peut être calculée de diverses façons :

* Tout d'abord, il est naturel d'utiliser une mesure topologique et qualitative du graphe comme la similarité (voir Table 3.1 et Section 3.3).
* Il est également possible de projeter la matrice d'un graphe (A ou S) sur un espace euclidien. Dans ce cas, la distance correspond à une mesure basée sur la distance euclidienne entre les points de l'espace : la moyenne (les méthodes de mean-cut, k-mean), le maximum (les méthodes de max-cut), le minimum (les méthodes de min-cut). Par exemple, la méthode de min-cut peut être utilisée pour minimiser le nombre d'arêtes entre les sous-ensembles à partir de la matrice d'adjacence A. Elle peut aussi être appliquée sur la matrice de similarité S, ce qui permet d'affiner le partitionnement et donc de créer une classification automatique par similarité.
* En dernier lieu, il est possible de réaliser une coupe dans l'arbre binaire à l'aide d'une *fonction objective* afin de déterminer la partition finale. Cette approche a l'avantage de trouver un nombre optimal de communautés par lecture de l'arbre binaire mais elle s'avère coûteuse en temps de calcul.

– Une autre manière d'analyser la structure communautaire d'un réseau biparti est de réaliser une projection sur un des deux ensembles de sommets et ensuite d'appliquer une méthode de détection unipartie. Cependant, le résultat d'une telle approche mène nécessairement à une perte d'information qui, à son tour, entraînera une moins bonne compréhension du réseau [EB13]. Dans cette étude [EB13], les auteurs proposent une méthode de "double projection" qui consiste à projeter les deux ensembles $\top$ et $\bot$ et à analyser la structure communautaire de chaque projection indépendemment pour finalement les recombiner à l'aide d'un critère à maximiser. L'avantage premier de cette méthode est qu'elle ne nécessite rien de plus qu'une mesure capable de faire émerger les équivalences structurelles entre les deux projections. Pour cela, les auteurs préconisent d'utiliser une mesure binaire, dite "proportion par correspondance" [Bur76], qui donne la tendance de deux sommets à être ou non reliés aux mêmes autres sommets. Il est également proposé de décomposer séparément les deux projections en communautés et de maximiser une version de la modularité pour les graphes bipartis [Bar07, GSPA07, Mur09]. Cette méthode est opposée à des algorithmes de détection qui considèrent les deux ensembles de sommets simultanément (voir méthode de *biclustering* et de partitionnement spectral). Il s'avère que ces derniers sont moins efficaces, venant du fait que le nombre de communautés doit être identique pour chacun des deux ensembles de sommets [Mel14]. Finalement, cette méthode de "double

projection" permet de s'affranchir de la perte d'information survenant lorsqu'un seul ensemble est considéré. Il est ainsi possible de retrouver la matrice d'adjacence bipartie d'origine à partir de celles des projections [EB13]. Elle a aussi l'avantage d'être flexible et ne requiert aucun algorithme supplémentaire.

- La deuxième approche utilise conjointement les deux ensembles $\top$ et $\bot$ du graphe biparti et s'apparente aux méthodes de *biclustering* [MO04] (ou *co-clustering*) qui permettent de regrouper simultanément les lignes et les colonnes d'une matrice. La procédure classique consiste à effectuer des permutations entre les lignes et les colonnes de la matrice afin de trouver un certain agencement appelé *bicluster*. Un bicluster est une sous-matrice contenant des valeurs similaires. Selon l'arrangement des valeurs, un bicluster peut être un *bicluster à valeur constante*, un *bicluster à valeur constante sur les lignes* (ou *sur les colonnes*), un *bicluster à valeur cohérente*. La méthode de permutations fonctionne si les données sont relativement en ordre. Cependant, puisque la plupart des données contiennent du bruit, d'autres méthodes doivent être envisagées. Pour y parvenir, plusieurs travaux ont été proposés [CST+00, YWWY03, CDGS04, GLD00, HDC+01, TSS02] s'inspirant des approches spectrales et de regroupement hiérarchique. L'avantage premier des méthodes de biclustering est qu'elles sont compatibles avec les graphes unipartis et les graphes bipartis. De plus, il est possible d'identifier des zones recouvrantes entre biclusters et donc de détecter des communautés recouvrantes [YWWY03].

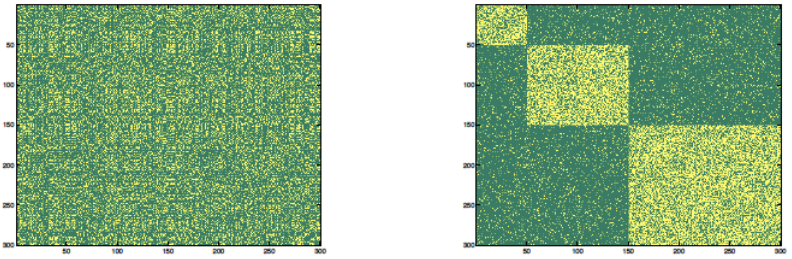


FIGURE 3.4: Exemple de la réorganisation des lignes et colonnes d'une matrice d'adjacence.

- La troisième s'intéresse au *partitionnement spectral* [WS05, TL10] du réseau. Ce dernier diffère du regroupement hiérarchique car il consiste à réduire au préalable l'espace des dimensions d'une matrice. Pour cela, une transformation de Laplace appliquée à cette matrice permet d'utiliser l'information contenue dans les vecteurs propres de la matrice de Laplace obtenue (à l'aide d'un algorithme de calcul de valeurs propres). En considérant cette nouvelle matrice, un partitionnement peut être effectué avec les algorithmes standards de clustering. Parallèlement, il existe une formulation matricielle de la modularité proposée par [TL10], qui se rapporte à un problème de partitionnement spectral [New06a].

Mesure $\theta(u,v)$	**Expr. unipartie**	**Expr. bipartie**
		Mesures locales
Voisins communs (CN)	$\lvert N(u) \cap N(v) \rvert$ (3.3)	$\lvert N_\top(u) \cap N_\top(v) \rvert$ (3.4)
Jaccard (CC)	$\frac{\lvert N(u) \cap N(v) \rvert}{\lvert N(u) \cup N(v) \rvert}$ (3.5)	$\frac{\lvert N_\top(u) \cap N_\top(v) \rvert}{\lvert N_\top(u) \cup N_\top(v) \rvert}$ (3.6)
HPI	$\frac{\lvert N(u) \cap N(v) \rvert}{min(\lvert N(u) \rvert, \lvert N(v) \rvert)}$ (3.7)	$\frac{\lvert N_\top(u) \cap N_\top(v) \rvert}{min(\lvert N_\top(u) \rvert, \lvert N_\top(v) \rvert)}$ (3.8)
HDI	$\frac{\lvert N(u) \cap N(v) \rvert}{max(\lvert N(u) \rvert, \lvert N(v) \rvert)}$ (3.9)	$\frac{\lvert N_\top(u) \cap N_\top(v) \rvert}{max(\lvert N_\top(u) \rvert, \lvert N_\top(v) \rvert)}$ (3.10)
Adamir-Adar (AA)	$\sum_{w \in N(u) \cap N(v)} \frac{1}{log(\lvert N(w) \rvert)}$ (3.11)	$\sum_{w \in N_\top(u) \cap N_\top(v)} \frac{1}{log(\lvert N_\perp(w) \rvert)}$ (3.12)
Allocation de ressource (RA)	$\sum_{w \in N(u) \cap N(v)} \frac{1}{\lvert N(w) \rvert}$ (3.13)	$\sum_{w \in N_\top(u) \cap N_\top(v)} \frac{1}{\lvert N_\perp(w) \rvert}$ (3.14)
Attachement préférentiel (PA)	$d(u) \times d(v)$ (3.15)	$d_\top(u) \times d_\top(v)$ (3.16)
Cosinus (Salton)	$\frac{\lvert N(u) \cap N(v) \rvert}{\sqrt{\lvert N(u) \rvert \times \lvert N(v) \rvert}}$ (3.17)	$\frac{\lvert N_\top(u) \cap N_\top(v) \rvert}{\sqrt{\lvert N_\top(u) \rvert \times \lvert N_\top(v) \rvert}}$ (3.18)
Sørensen Index	$\frac{2 \times \lvert N(u) \cap N(v) \rvert}{\lvert N(u) \rvert + \lvert N(v) \rvert}$ (3.19)	$\frac{2 \times \lvert N_\top(u) \cap N_\top(v) \rvert}{\lvert N_\top(u) \rvert + \lvert N_\top(v) \rvert}$ (3.20)
LHN1	$\frac{\lvert N(u) \cap N(v) \rvert}{d(u) \times d(v)}$ (3.21)	$\frac{\lvert N_\top(u) \cap N_\top(v) \rvert}{d_\top(u) \times d_\top(v)}$ (3.22)
		Mesures globales
Distance de Hamling	$1 - \frac{\lvert N(u) \cap N(v) \rvert}{n}$ (3.23)	$1 - \frac{\lvert N_\top(u) \cap N_\top(v) \rvert}{n_\perp}$ (3.24)
Mesure de Katz	$\sum_{l=1}^{\infty} \lvert \sigma^l(u,v) \rvert$ (3.25)	

où $\sigma^l(u,v)$ désigne l'ensemble des chemins de longueur l entre u et v.

ACT	$m(u,v) + m(v,u)$ (3.26)	

où $m(u,v)$ est le nombre moyen de pas pour un marcheur aléatoire partant de u à v.

TABLE 3.1: Colonne centrale : mesures de similarités dyadiques pour les graphes unipartis. Une liste plus complète peut être trouvée dans [LZ11, Kan13]. Colonne droite : mesures de similarités dyadiques pour les graphes bipartis sur l'ensemble $\top$. Inversement, ces mesures peuvent être appliquées sur l'ensemble $\perp$.

3.2.1.3 Approches par modèles de bloc

Le *modèle à blocs stochastiques* SBM (Stochastic Block Model) est une autre approche permettant la détection de communautés. Toutefois, les SBM ne sont pas employés en premier lieu dans le but d'identifier des communautés mais plutôt de modéliser les propriétés d'un réseau, notamment la structure communautaire.

Un SBM est un modèle génératif où il est question d'approximer la structure communautaire d'un graphe par une structure en blocs [HLL83, NS01]. De façon générale, un SBM, noté $SBM(n, k, Q)$, génère un graphe aléatoire $G = (V, E)$ où $n = |V|$ est le nombre de sommets, k est le nombre de blocs, groupes ou communautés. Une matrice de probabilités Q de dimension $k \times k$ permet d'échantillonner l'ensemble E du graphe. Ainsi, une arête $\{u, v\}$ existe avec une probabilité Q_{C_u,C_v}, où C_u et C_v indiquent à quelle communauté appartiennent u et v. Dans la littérature, on retrouve cette phrase illustrant l'idée première d'un modèle à blocs stochastiques : « *la probabilité d'une relation entre deux nœuds dépend des blocs auxquels ils appartiennent* ». Par exemple, une matrice diagonale Q (dont les valeurs sont nulles sauf sur la diagonale) produira des composantes déconnectées entre elles, tandis qu'ajouter des valeurs faibles en dehors de la diagonale aura tendance à générer une structure communautaire conventionnelle – la densité interne à une communauté est plus forte que sa densité externe. En modifiant la matrice de probabilité, il est donc possible de créer une grande variété de structures ; structure bipartie, k-partie, à plusieurs couches, avec une hiérarchie, etc. Cependant, le modèle décrit ci-dessus ne permet pas très bien de retranscrire les propriétés des réseaux réels. Par exemple, lorsque k devient arbitrairement grand alors il est possible de représenter n'importe quel réseau par un SBM où chaque sommet est dans sa propre communauté et où la probabilité de chaque arête est soit 1 soit 0 selon si il existe une arête entre les sommets dans le réseau original. Dans ce sens, plusieurs améliorations ont été proposées dont une version corrigeant le degré des nœuds [KN11, LCJ14], une version pour le recouvrement des communautés [LBA11], pour une hiérarchie des communautés [PMB10, ABFX08], pour les graphes bipartis [LCJ14].

Afin d'utiliser les modèles à blocs stochastiques pour la détection de communautés (ou *weak recovery* dans la littérature), il est nécessaire de rechercher les paramètres qui répondent au mieux aux propriétés observées ou connues *a posteriori*. Pour cela, une fonction objective est généralement utilisée pour guider l'exploration des paramètres (modularité [NG04, KN11], information mutuelle [SWB51, DMM03], méthodes spectrales [RCY+11]). En fait, un SBM est plus général qu'un algorithme de détection de communautés traditionnel car diverses formes de structures peuvent être générées. Les SBM sont aussi utilisés pour étudier les problèmes d'optimisation ou de clustering (voir Section 3.2.1.1 et Section 3.2.1.1) ou encore pour créer des réseaux synthétiques avec une

structure communautaire bien définie et ainsi évaluer la qualité des résultats fournis par les algorithmes de détection de communautés.

3.2.2 Approches centrées groupes

Une communauté peut être vue comme un ensemble de sommets ayant des caractéristiques communes. Dans ce sens, l'exemple le plus trivial est de considérer une communauté comme une *clique maximale* (voir Section 2.1.1.7). Typiquement, la recherche de cliques et de cliques maximales est un problème difficile, inadaptée aux réseaux de grande taille. Toutefois, des méthodes s'intéressent à utiliser cette notion de cliques comme élément de base pour la détection de communautés :

- La méthode de *percolation de cliques* [PDFV05] dite **CPM** (Clique Percolation Method) emploie les k-cliques (sous-graphes complets de taille k). Dans cette étude, une communauté est définie comme l'ensemble des k-cliques atteignables par une série de k-cliques adjacentes (deux cliques étant adjacentes si elles partagent entre elles $k-1$ sommets). Par défaut, l'algorithme énumère toutes les cliques de taille k et cherche ensuite les k-cliques adjacentes. Le principal avantage de cette méthode est qu'elle autorise le recouvrement entre communautés. De plus, le choix de la valeur k laissée à l'utilisateur permet de changer la résolution de l'observation du réseau. Une valeur élevée de k permet de détecter précisément les régions denses du graphe alors qu'une valeur plus basse permet d'étudier les zones éparses du graphe. Cette méthode fonctionne très bien avec les réseaux réels pour des valeurs de 3 ou 4 quand il y a un fort clustering. En revanche, il n'est pas toujours évident de trouver une valeur appropriée au réseau et l'algorithme peut parfois devenir gourmand en termes de complexité si le graphe est très dense (une amélioration de **CPM** est proposée dans [KKKS08]).

- Une version semblable [LSH08] existe aussi pour les réseaux bipartis à travers la notion de bicliques (voir Section 2.1.2.2). Dans ce contexte, deux bicliques $K_{a,b}$ sont adjacentes si elles partagent au moins une clique $K_{a-1,b-1}$. Il est également possible de détecter le recouvrement des communautés. Reprenant la même méthode, l'algorithme **BiTector** [DWWW08] identifie dans un premier temps les cliques maximales afin de les utiliser comme *cœur du réseau*. Ensuite, à l'aide d'une fonction de similarité, les communautés sont construites au fur et à mesure en élargissant ou en fusionnant les cœurs. Cette méthode fonctionne bien lorsque n et m ont le même ordre de grandeur, cependant elle est très lente lorsque le réseau est dense.

- Une autre étude [DZLH06], plus en marge, fonctionne avec les réseaux bipartis en utilisant une méthode de biclustering (voir Section 3.2.1.2) pour les bicliques.

- Une méthode énumère les γ-cliques à l'aide d'une heuristique d'optimisation [ARS02]. Une γ-clique ou quasi-clique est une version relâchée de la clique, dont la densité est supérieure à $\gamma \in [0; 1]$.

- L'algorithme **EAGLE** [SCCH09] est une méthode qui utilise à son avantage plusieurs approches en parallèle. Il est capable de détecter des communautés recouvrantes et hiérarchiques. L'algorithme commence par identifier toutes les cliques maximales, qui sont les communautés initiales. Ensuite, à l'aide d'une mesure de similarité (voir Section 3.2.1.2), les communautés semblables sont fusionnées formant de nouvelles communautés qui, à leur tour, seront fusionnées avec des communautés semblables. Ceci est répété jusqu'à ce qu'il ne reste plus qu'une seule communauté. Dans sa seconde phase, l'algorithme procède à une "coupe optimale" dans le dendrogramme (voir Figure 3.3) de la hiérarchie des communautés produites. Afin de déterminer l'endroit de la coupe, **EAGLE** utilise une extension de la modularité (voir Section 3.2.1.1) adaptée au recouvrement [NMMC08] qui va juger de la qualité de la partition.

3.2.3 Approches centrées propagation

Les approches par propagation exploitent la propriété de densité intra-communauté des réseaux réels. En effet, la structure communautaire d'un réseau se définit par l'existence des groupes de nœuds plus densément connectés qu'avec le reste du réseau. Comparativement, ceci implique que la densité interne à ces groupes (densité intra-communautaire) est relativement plus élevée que la densité externe de ces groupes (densité inter-communautaire). De cette manière, il est possible d'admettre qu'un signal émis par un nœud et retransmis par ses voisins a plus de chance de rester bloqué dans sa communauté, qu'il n'a de chance d'être propagé à une autre communauté. Il existe différents algorithmes utilisant cette propriété :

- Les méthodes basées sur la *propagation d'étiquettes* (ou labels). L'idée de ces algorithmes est de propager dans un graphe une étiquette unique de voisin en voisin en remplaçant les étiquettes de chacun. La première méthode est proposée par [RAK07] et dénommée **LPA** (Label Propagation Algorithm) dont le fonctionnement peut être résumé en trois étapes :

  1. À l'état initial, tous les sommets possèdent une étiquette unique.
  2. Un sommet du graphe choisit aléatoirement remplace son étiquette avec l'étiquette partagée avec le plus grand nombre de ses voisins. Cette même procédure est appliquée successivement pour chaque sommet du graphe choisi

aléatoirement. Après un certain nombre d'itérations, la même étiquette sera associée aux sommets appartenant à la même communauté.

3. Idéalement, l'algorithme s'arrête si aucun changement de label n'est effectué. Finalement, tous les sommets avec la même étiquette appartiennent à la même communauté.

Ce processus peut également être effectué de manière synchrone, en mettant à jour les étiquettes de tous les sommets du graphe en même temps. Ceci permet notamment de converger plus rapidement vers une solution.

Deux adaptations de cette méthode ont été proposées afin d'intégrer la détection de communautés chevauchantes :

- Dans l'article [Gre10] l'étape 2 est modifiée pour transmettre une liste des étiquettes les plus courantes de l'entourage. Ainsi, il est possible en fin d'exécution de choisir le nombre maximum d'étiquettes à retenir comme communautés, et donc de faire émerger des communautés recouvrantes.
- Une autre variante pour le recouvrement proposée par [XSL11] consiste à conserver une liste des étiquettes les plus courantes sur chaque sommet. Ceci permet d'attribuer à chaque sommet une force d'appartenance à une communauté. Dit autrement, un sommet est membre d'une communauté selon une certaine probabilité. Il est donc possible d'affilier plusieurs communautés à chaque sommet en fixant une valeur limite à cette probabilité.

L'approche par propagation d'étiquettes se prête efficacement à l'identification de communautés non-recouvrantes car la diffusion est automatiquement freinée par la structure communautaire du réseau, délimitant ainsi les communautés. Cependant, ce principe n'est plus respecté par la possibilité d'avoir du chevauchement entre communautés. On peut alors se demander si le recouvrement est réellement adapté aux deux premières méthodes décrites ci-dessus [Gre10, XSL11] : le choix de paramètres arbitraires se révèle déterminant et peut mener à des détections plus ou moins cohérentes. D'autres méthodes de propagation d'étiquettes plus récentes sont à considérer [ŠB11, CG12].

Les méthodes précédentes sont inadaptées aux graphes bipartis. En effet, dû à la bissection des deux ensembles, il est impossible de faire converger l'algorithme qui se retrouve piégé dans un échange infini de labels. Pour parer à ce problème, une solution est proposée par [Gre10]. L'auteur développe l'algorithme **COPRA** (Community Overlap PRopragation Algorithm), dont la particularité est de définir l'étiquette d'un sommet comme un ensemble de paires (c, b) où c est une communauté et b un coefficient d'appartenance à la communauté c. Étant donnés

un sommet u et une communauté c, ce coefficient d'appartenance est défini par :

$$b_t(u,c) = \frac{\sum_{v \in N_\top(u)} b_{t-1}(v,c)}{|N_\top(u)|}.$$

$b_t(u,c)$ indique l'appartenance de u à la communauté c à l'itération t en fonction des étiquettes des voisins de u à l'itération $t-1$. Chaque sommet met à jour son coefficient d'appartenance en faisant la moyenne des coefficients de ses voisins. Finalement, cette méthode permet le recouvrement entre communautés.

- Les méthodes basées sur un *marcheur aléatoire*. Un marcheur aléatoire est un explorateur qui choisit toujours sa prochaine destination au hasard. On emploie l'expression *marche aléatoire* pour une succession de *pas aléatoires* qui indiquent qu'à chaque instant la future destination est totalement décorrélée de l'endroit où l'on se trouvait au pas précédent. Formellement, dans un graphe uniparti G, une marche aléatoire est définie par une séquence de pas aléatoirement et uniformément choisis entre les sommets de G en suivant les arêtes. Si le marcheur aléatoire commence au sommet u alors la probabilité d'atteindre un sommet v par l'arête $\{u,v\}$ dépend du degré de u. Cette probabilité de transition est notée $p(u,v) = \frac{1}{d(u)}$ si $v \in N(u)$ sinon $p(u,v) = 0$ et donne la matrice des probabilités de transition T d'une marche aléatoire.

  Appliquée à la détection de communautés, la marche aléatoire exploite la même propriété que la propagation d'étiquettes, à savoir qu'un marcheur aléatoire a une forte probabilité de rester coincé dans la communauté du nœud de départ. Cette propriété est particulièrement bien exploitée par l'algorithme **WalkTrap** [PL05] qui calcule pour chaque sommet dans le graphe un vecteur donnant la probabilité qu'un marcheur aléatoire arrive aux autres sommets du réseau en k pas. Ces vecteurs sont ensuite utilisés pour calculer les similarités entre les sommets avec une mesure de distance. Le partitionnement est laissé à une méthode de regroupement hiérarchique (voir Section 3.2.1.2). La complexité de cet algorithme est en $O(mn^2)$ et n'est donc pas assez performante pour être appliquée efficacement aux réseaux réels contenant des millions de nœuds ou de liens.

L'approche centrée propagation considérée comme la plus efficace est l'algorithme **Infomap** [RAB09]. Cette méthode repose sur l'utilisation d'un marcheur aléatoire et de la propagation d'étiquettes. L'algorithme est, tout d'abord, relativement rapide, ce qui permet de l'appliquer aux réseaux réels. Mais ce qui a fait sa popularité est qu'il a montré qu'il était l'algorithme le plus performant sur la quasi-totalité des cas de figure, avec un écart important par rapport aux méthodes d'optimisation de la modularité, sauf pour l'algorithme de **Louvain** (voir Section 3.2.1.1). Un précieux avantage de cet algorithme

est qu'il est compatible avec n'importe quel type de graphes, graphes simples, orientés ou pondérés, graphes bipartis, graphes multi-couches, avec ou sans recouvrement. Le principe derrière Infomap est qu'il utilise une méthode issue de la théorie de l'information pour mesurer l'encodage optimal d'une marche aléatoire (voir Figure 3.5). Il s'inspire également de l'algorithme de Huffman pour la compression de l'encodage [Huf52].

Pour commencer, chaque nœud est décrit par un identifiant unique, qui est en fait un codage binaire. L'objectif est ensuite de décrire les mouvements du marcheur aléatoire par les identifiants des nœuds qu'il parcourt. L'algorithme cherche donc à minimiser la longueur des séquences parcourues par un marcheur aléatoire. Un encodage performant exploite nécessairement les régularités de motifs données par une succession d'identifiants. Si on peut trouver un code optimal décrivant les chemins réguliers du marcheur aléatoire alors ceci indique qu'on a trouvé une structure importante du réseau. Dans ce cas, dès lors qu'il est possible de compresser une partie de la séquence du marcheur aléatoire, le codage simplifié est attribué aux nœuds impliqués, définissant donc la communauté à laquelle ils appartiennent.

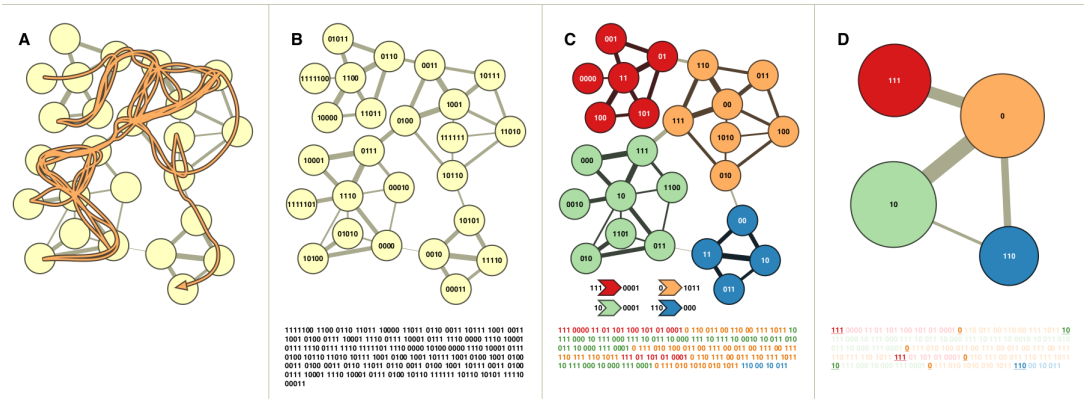


FIGURE 3.5: Illustration du fonctionnement de Infomap (issue de l'article [RAB09]). L'image A représente les parcours du marcheur aléatoire que l'on souhaite encoder. L'image B présente la séquence des parcours : on attribue un identifiant unique sur chaque nœud qui sert à encoder le parcours (en noir, au-dessous). Les images C et D montrent comment optimiser la séquence : la répétition d'identifiants permet ainsi de définir les communautés conduisant à l'encodage minimal.

Par exemple, tant qu'un marcheur aléatoire reste dans une communauté alors il est possible de compresser efficacement la séquence de son parcours. À l'inverse, si le marcheur aléatoire se retrouve sur un nœud en dehors de la communauté alors la compression est plus coûteuse et on peut donc relancer un nouvelle marche. Par conséquent, après un certain nombre de marches aléatoires, il est possible de distinguer la structure communautaire grâce à la qualité de la compression des séquences obtenues. On comprend donc qu'avec ce système de description des déplacements, que quitter une communauté se révèle très coûteux. Un bon découpage consistera donc à rendre ces déplacements le plus rare possible. De plus, la compression réalisée sur le réseau permet d'évaluer la qualité

du partitionnement et des communautés. Cependant, l'encodage obtenu avec Infomap est unique dans le sens qu'il ne peut pas être comparé avec une partition provenant d'une autre méthode. Il ne peut donc pas être utilisé comme méthode d'évaluation à part entière.

3.3 Évaluation de la détection de communautés

Nous avons présenté dans la section précédente (Section 3.2) une classification des approches et méthodes qui s'intéressent à la détection de communautés. Bien que la notion de communauté ne soit pas clairement définie, les recherches menées dans ce domaine en constituent quelques pistes ayant comme objectif commun de révéler les propriétés structurelles et topologiques des réseaux réels. De par le nombre important de méthodes proposées, il est nécessaire de développer des moyens d'évaluer les algorithmes de détection de communautés. Tout d'abord, l'évaluation d'une méthode peut être réalisée sur sa complexité algorithmique, c'est-à-dire sur sa capacité à délivrer un résultat en un temps raisonnable, ainsi que sur sa consommation en mémoire. D'autre part, l'évaluation peut porter directement sur la fiabilité du résultat fourni. Il est donc question, pour un algorithme de détection de communautés, d'être capable, grâce à un *critère*, d'évaluer rigoureusement la qualité d'une détection, et ceci quel que soit son type (partition, recouvrante, hiérarchique). Cette évaluation est aussi utile pour pouvoir confronter les méthodes entre elles, afin de sélectionner la meilleure. Pour l'identification des communautés, elle peut également faire office de guide ou de condition d'arrêt de l'algorithme, ou encore être utilisée comme fonction de qualité à optimiser. Lorsqu'il s'agit de réseaux de petites tailles, les communautés peuvent être évaluées manuellement par une simple visualisation. En revanche, la taille de la plupart des réseaux réels ne permet pas une telle évaluation. Ceci est un problème ouvert faisant l'objet de nombreuses recherches, il y a généralement deux façons de procéder.

3.3.1 Les vérités de terrain

La vérité de terrain empirique Une *vérité de terrain* est la connaissance d'un réseau, son processus de formation, sa structure communautaire, etc. Généralement, les vérités de terrain proviennent d'études interdisciplinaires, par exemple de la sociologie, de la biologie, ou d'un contexte plus pratique et technique. Une vérité de terrain est très utile car elle permet de comparer directement les communautés détectées avec les communautés de la vérité de terrain. Ceci permet donc d'évaluer la méthode de détection utilisée et de quantifier la précision du résultat (voir Figure 3.6). Il existe en réalité peu de vérités de terrain. En effet, même sur des réseaux de petite taille, ces dernières

exigent un investissement colossal en temps. Lorsqu'il est question d'étudier un réseau réel contenant plusieurs millions de nœuds et de liens alors ce travail est tout bonnement irréalisable. De plus, d'un point de vue critique, on peut questionner la pertinence d'une vérité de terrain, son objectivité scientifique, sa fiabilité, son exactitude [LC13].

La vérité de terrain artificielle À l'opposé, il est aussi possible de considérer une partie de l'information du réseau comme étant une vérité de terrain. Prenons l'exemple du réseau social de YOUTUBE, où les utilisateurs peuvent à la fois créer des liens d'amitiés entre eux et créer des groupes que les autres utilisateurs peuvent rejoindre, ici la vérité de terrain peut être vue comme les groupes créés par les utilisateurs. Les groupes sont donc des communautés au sens large du terme, elles sont définies par les utilisateurs.
Une autre façon de créer une vérité de terrain consiste à utiliser un modèle aléatoire (voir Section 2.2.2) permettant de fixer une structure communautaire au préalable. Le modèle le plus répandu est LFR [LFR08], ce dernier génère un graphe à partir de plusieurs paramètres (nombre de sommets, degré moyen, degré maximum, distribution des degrés) disposant d'une structure communautaire ajustable à l'aide d'un paramètre de mélange (le *mixing parameter* μ). Ce dernier permet de définir une structure communautaire plus ou moins cohésive. Basiquement, chaque sommet est affecté à une communauté si ce dernier partage une fraction d'au moins $1 - \mu$ voisins en commun avec les autres sommets de la communauté. Après exécution de LFR, l'utilisateur obtient un graphe qui respecte les paramètres donnés en entrée avec les communautés associées. Au moyen de ce graphe synthétique, la connaissance de la structure communautaire fait office de vérité de terrain et il est donc possible de mesurer la qualité des méthodes de détection.

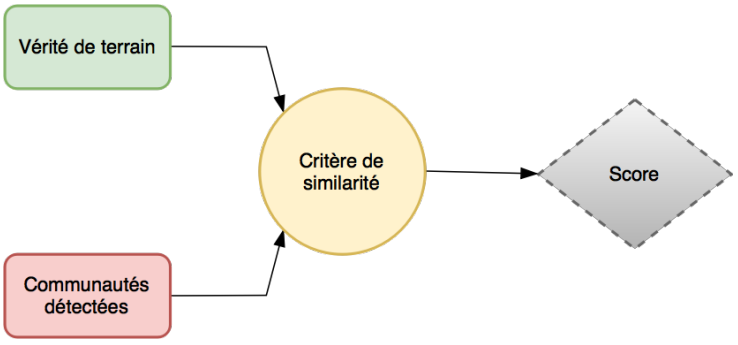


FIGURE 3.6: Une manière simple d'évaluer les communautés détectées par un algorithme avec la vérité de terrain.

Afin de confronter les communautés détectées avec la vérité de terrain, il est courant d'utiliser une mesure empruntée à la théorie de l'information, l'*information mutuelle* [SWB51]. L'information mutuelle entre deux systèmes s'exprime par la quantité d'information que l'un apporte sur l'autre, c'est-à-dire la variation d'information, cette

quantité est mesurée par l'*entropie* [SWB51]. Autrement dit, plus l'information est différente, plus l'entropie est forte car la quantité d'information nécessaire pour décrire les deux systèmes est grande. En fin de compte, l'information mutuelle permet de mesurer le degré de dépendance entre deux systèmes. Appliquée à l'évaluation de partitions, on utilise l'*information mutuelle normalisée* ou NMI [SG02], qui mesure à quel point l'information contenue entre deux partitions est similaire ou différente (dépendance ou indépendance). La NMI est finalement un critère de similarité notée :

$$NMI(\mathcal{P}_A, \mathcal{P}_B) = \frac{-2\sum_{i=1}^{k}\sum_{j=1}^{l} C_{ij}\log\left(C_{ij}n/C_{i.}C_{j.}\right)}{\sum_{i=1}^{k} C_{i.}\log\left(C_{i.}/n\right) + \sum_{j=1}^{l} C_{j.}\log\left(C_{j.}/n\right)} \tag{3.27}$$

où $\mathcal{P}_A$ et $\mathcal{P}_B$ sont respectivement la vérité de terrain et la partition obtenue par une méthode de détection avec $\mathcal{P}_A = \{A_1, A_2, \ldots, A_k\}$ contenant k communautés et $\mathcal{P}_B = \{B_1, B_2, \ldots, B_l\}$ contenant l communautés. Puisqu'il s'agit d'une partition, $\mathcal{P}_A$ est une division du réseau telle que $\bigcup_{i=1}^{k} A_i = V$ et $A_i \cap A_j = \emptyset$ $(\forall i \neq j)$. C représente la matrice de confusion de dimension $k \times l$ qui assigne à chaque entrée C_{ij} le nombre de sommets partagés entre les communautés A_i et B_j. Lorsque les deux partitions sont identiques la valeur de la NMI est 1 et lorsque les deux partitions sont complètement différentes la NMI prend la valeur 0.

Néanmoins cette version de la NMI présente quelques problèmes. Intuitivement, la similarité entre deux partitions aléatoires devrait être constante ou être représentée idéalement par une valeur nulle. Cependant, il a été montré que la NMI tend à favoriser les partitions dont le ratio entre le nombre de sommets et le nombre de communautés est petit [AP15]. De plus, la complexité de la NMI n'est pas toujours adaptée à des calculs sur de grands réseaux. Pour parer à ces problèmes, différentes variantes de la NMI ont été proposées, généralement en modifiant le terme de la normalisation. En voici quelques-unes : l'*information mutuelle ajustée* (AMI) [VEB09], l'*information mutuelle unifiée* (SMI) [VEB09], l'*information mutuelle à facteur d'échelle* (FNMI) [AP15], une version adaptée aux communautés recouvrantes [LFK09, MGH11] (ONMI).

Il existe également d'autres critères basés sur le comptage du nombre d'éléments, ou paires de sommets, entre deux partitions, dit *classification binaire* [Mei07] :

1. N_{11} nombre de paires de sommets qui sont dans les mêmes communautés dans $\mathcal{P}_A$ et $\mathcal{P}_B$,
2. N_{00} nombre de paires de sommets qui sont dans différentes communautés dans $\mathcal{P}_A$ et $\mathcal{P}_B$,
3. N_{10} nombre de paires de sommets qui sont dans les mêmes communautés dans $\mathcal{P}_A$ mais pas dans $\mathcal{P}_B$,
4. N_{01} nombre de paires de sommets qui sont dans les mêmes communautés dans $\mathcal{P}_B$ mais pas dans $\mathcal{P}_A$.

Les quatre cas de figure satisfont : $N_{11} + N_{00} + N_{10} + N_{01} = n(n-1)/2$. Un exemple typique de l'utilisation de ce comptage est l'indice de Jaccard (voir Équation 2.8) :

$$J(\mathcal{P}_A, \mathcal{P}_B) = \frac{N_{11}}{N_{11} + N_{10} + N_{01}}$$

Ce critère permet de calculer la proportion du nombre d'éléments (paires de sommets) appartenant aux mêmes communautés dans $\mathcal{P}_A$ et $\mathcal{P}_B$ (donné par N_{11}) par rapport à l'ensemble des éléments existants dans les communautés de $\mathcal{P}_A$ et $\mathcal{P}_B$ (donné par $N_{11} + N_{10} + N_{01}$).

Un autre critère développé est l'indice de Rand [Ran71] défini par :

$$R(\mathcal{P}_A, \mathcal{P}_B) = \frac{N_{11} + N_{00}}{n(n-1)/2} \tag{3.28}$$

Il calcule la fraction des éléments correctement et mal classifiés par rapport à l'ensemble des éléments existants. La valeur de R varie de 0, lorsque aucun élément n'est classifié de la même manière, à 1, lorsque les deux partitions sont identiques. Par ailleurs, cette valeur semble être affectée par le nombre de communautés et le nombre de sommets [MA84], ce qui a pour conséquence de faire tendre R vers 1 lorsque le nombre de communautés est important. Une solution est proposée pour parer à ce problème grâce à l'indice de Rand ajusté (ARI) [HA85].

Un autre critère, connexe au précédent, est le score-F1, ou *F1-score*, qui est la moyenne harmonique de la *précision* et du *rappel*. En statistique, la précision p correspond au nombre d'éléments correctement classifiés par rapport au nombre total d'éléments identifiés. Tandis que le rappel r correspond à ce même nombre d'éléments correctement classifiés mais rapporté au nombre d'éléments considérés comme pertinents. En d'autres termes, si on considère $\mathcal{P}_A$ comme étant la vérité de terrain et $\mathcal{P}_B$ comme étant la partition identifiée, avec $C_i \in \mathcal{P}_A$, $C_j \in \mathcal{P}_B$ et $Te = |C_i \cap C_j|$ le nombre de sommets en commun entre deux communautés C_i et C_j, la précision sera notée $p = \frac{Te}{|C_j|}$ et le rappel $r = \frac{Te}{|C_i|}$. Ainsi, le score-F1 est noté :

$$F_1(C_i, C_j) = 2 \times \frac{p * r}{p + r} \tag{3.29}$$

On peut également définir la moyenne du score-F1 entre deux partitions [YL13] :

$$F_1(\mathcal{P}_A, \mathcal{P}_B) = \frac{1}{2}\Big(\frac{1}{|\mathcal{P}_A|}\sum_{i=1}^{k} F_1(C_i, g'(i)) + \frac{1}{|\mathcal{P}_B|}\sum_{j=1}^{l} F_1(g(j), C_j)\Big) \tag{3.30}$$

où $g'(i) = \underset{C_j \in \mathcal{P}_B}{\operatorname{argmax}}\ F_1(C_i, C_j)$ et $g(j) = \underset{C_i \in \mathcal{P}_A}{\operatorname{argmax}}\ F_1(C_i, C_j)$.

En définitive, les critères présentés dans cette partie peuvent être utilisés avec n'importe quel type de détection. En revanche, selon le contexte, certains sont préférables à d'autres. Par exemple, lorsque les communautés détectées sont recouvrantes, l'évaluation peut être biaisée par certains critères. Pour cela, il est fréquent que plusieurs critères soient utilisés pour évaluer un seul résultat, ce qui de plus permet d'obtenir une évaluation plus précise ou de noter les différences entre critères.

3.3.2 La qualité d'une communauté

Dans de nombreux cas de figures, aucune connaissance ou vérité de terrain du réseau n'est disponible pour évaluer la qualité d'une détection. Pour remédier à cela, il est possible de mesurer les caractéristiques structurelles intrinsèques aux communautés. Ainsi, une valeur est attribuée à chaque communauté détectée à l'aide d'une fonction de score, qui est à proprement parler *une mesure d'évaluation interne*. Idéalement, l'ensemble de ces valeurs doivent refléter la qualité de la détection. Un autre intérêt de cette méthode d'évaluation est qu'elle permet d'analyser les communautés individuellement et ainsi de pouvoir les caractériser avec précision.

Afin d'illustrer l'idée principale de ce type d'évaluation, nous présentons une méthode de classification des communautés proposée par [RCC+04]. Les auteurs ici proposent de considérer deux catégories, les *communautés fortes* et les *communautés faibles*, et d'utiliser une fonction de score relativement simple dont la définition se base sur le degré des sommets. Plus précisément, cette dernière compare, pour une communauté C_i, le nombre d'arêtes à l'intérieur de la communauté au nombre d'arêtes sortantes de la communauté vers le reste du réseau, respectivement $d^{in}(C_i) = |\{\{u, v\} \in E | u \in C_i, v \in C_i\}|$ et $d^{out}(C_i) = |\{\{u, v\} \in E | u \in C_i, v \notin C_i\}|$. De cette manière, il est donc possible de ranger les communautés dans une des deux catégories, avec comme seule condition qu'une communauté forte satisfait $d^{in}(C_i) > d^{out}(C_i)$. Cependant, cette méthode rudimentaire n'est vraisemblablement pas assez fine pour traiter des cas pratiques.

Idéalement, on souhaiterait avoir une fonction de score qui permette de classer les communautés dans l'ordre décroissant grâce à une métrique et que cette dernière capture la qualité d'une détection en respectant certaines propriétés topologiques. Nous présentons plus bas différentes fonctions de scores toutes définies autour de l'intuition classique que les communautés sont des ensembles de sommets fortement reliés entre eux et peu reliés à l'extérieur. Il y a plusieurs façon de formaliser mathématiquement cette intuition. Un exemple classique de fonction de score est la modularité qui compare la densité interne de la communauté avec la densité moyenne qu'aurait la communauté

dans un réseau recâblé aléatoirement. La modularité d'une communauté C_i est notée :

$$Q(C_i) = \frac{d^{in}(C_i)}{m} - (\frac{2 \times d^{in}(C_i) + d^{out}(C_i)}{2m})^2 \tag{3.31}$$

Cette expression satisfait $\sum_{C_i \in \mathcal{P}} Q(C_i) = Q$ (voir Équation 3.1).

Globalement, il est possible de regrouper les fonctions de scores $f(C_i)$ autour de 3 classes, on note $n_{C_i} = |C_i|$ le nombre de sommets dans la communauté C_i (sa taille).

	Connectivité interne	
Densité interne	$f(C_i) = \frac{d^{in}(C_i)}{n_{C_i}(n_{C_i} - 1)/2}$	(3.32)
Séparabilité	$f(C_i) = \frac{d^{in}(C_i)}{d^{in}(C_i) + d^{out}(C_i)}$	(3.33)
Degré interne moyen	$f(C_i) = \frac{2 * d^{in}(C_i)}{n_{C_i}}$	(3.34)
	Connectivité externe	
Expansion	$f(C_i) = \frac{d^{out}(C_i)}{n_{C_i}}$	(3.35)
Ratio de coupe	$f(C_i) = \frac{d^{out}(C_i)}{n_{C_i}(n - n_{C_i})}$	(3.36)
	Connectivité interne et externe	
Conductance	$f(C_i) = \frac{d^{out}(C_i)}{2 * d^{in}(C_i) + d^{out}(C_i)}$	(3.37)
Flake-ODF	$f(C_i) = \frac{\lvert\{u \in C_i \lvert\lvert N(u) \cap C_i\rvert < \frac{d(u)}{2}\}\rvert}{n_{C_i}}$	(3.38)

TABLE 3.2: Différentes fonctions de score pour l'évaluation de la qualité des communautés détectées (voir d'autres dans [YL15]).

Les fonctions de score définies ci-dessus peuvent être considérées comme des mesures de similarités (voir Section 3.2.1.2 et Table 3.1). À ce même titre, les fonctions de similarité peuvent également servir de fonction de score. De manière générale, il est possible de donner une évaluation arbitraire de la structure communautaire identifiée par un algorithme. Le choix de la fonction de score détermine donc ce que l'on cherche à étudier, de l'intuition que l'on cherche à capturer, ou du contexte. Par exemple, il est logique que les algorithmes d'optimisation de la modularité obtiennent une meilleure évaluation si on utilise la fonction de score basée sur la modularité (Équation 3.31). À l'inverse, si

ce n'était pas le cas, alors on pourrait affirmer que ces algorithmes ne remplissent pas leurs objectifs. En fin de compte, le choix d'une fonction de score est subjectif et il est souvent nécessaire d'explorer différentes métriques afin de caractériser les communautés sous plusieurs angles de vue.

Chapitre 4

Étude du recouvrement de la structure communautaire bipartie des réseaux sociaux d'Internet

Sommaire

4.1 Jeux de données 46

4.2 Métriques pour le recouvrement 47

4.2.1 Métriques classiques 47

4.2.2 Nouvelles métriques 48

4.3 Analyse de la structure bipartie 49

4.4 Comparaison à l'aléatoire 55

4.5 Discussion 58

L'émergence d'Internet comme nouveau moyen de communication, accompagné d'un nouvel écosystème de données, rend les sociétés humaines de plus en plus digitalisées. En effet, ce nouveau support de communication entraîne de nombreuses mutations dans les rapports sociaux, les secteurs du travail, l'éducation ou l'accès à l'information. L'apparition rapide d'une grande variété de plateformes, services et applications sur Internet illustre bien l'attachement croissant au numérique. Un exemple emblématique est l'encyclopédie WikiPedia dont la vocation est de rendre accessible l'ensemble des connaissances de l'humanité et dont les utilisateurs sont eux-mêmes garant de ce contenu. Le développement des réseaux sociaux, tels que Facebook ou Twitter, est aussi une importante innovation d'Internet. Ces derniers sont des applications web permettant aux utilisateurs de communiquer et de partager de l'information facilement. Dans cette étude, on désigne un *réseau social* d'Internet comme étant plus largement un service en ligne (site web, plateforme, outil) impliquant des utilisateurs qui interagissent et créent du contenu. Ces réseaux sociaux sont donc issus d'un contexte réel, l'intérêt est alors de les analyser à l'aide des métriques généralement utilisées pour caractériser les réseaux réels (voir Section 2.2), et notamment leur structure communautaire (voir Chapitre 3).

Dans ce chapitre, nous nous intéressons à la structure communautaire des réseaux sociaux d'Internet qui présentent une structure bipartie. Par exemple, un réseau de films-acteurs qui relie des acteurs aux films dans lesquels ils ont joué, ou un réseau de co-publication qui connecte des auteurs à leurs publications [NSW01, NWS02], ont tous deux une structure bipartie. Il est donc naturel de distinguer formellement deux ensembles disjoints de nœuds : les acteurs d'un côté, les films de l'autre et les auteurs d'un côté, les publications de l'autre. Ou encore, un site web tel que WikiPedia où l'on retrouve d'un côté les pages web (sommets $\top$) et les contributeurs (sommets $\bot$).

Récemment, cette approche a été exploitée afin de proposer pour la première fois un modèle biparti de la topologie d'Internet [NWS02, TQM+13]. Le modèle est suffisamment général pour être appliqué à n'importe quel autre réseau présentant une structure bipartie car il s'appuie uniquement sur la séquence des degrés des nœuds des deux ensembles disjoints. Le travail a montré que malgré la simplicité du modèle, des propriétés réalistes et non triviales de la topologie d'Internet émergent naturellement. Mais, il a également montré que le modèle échoue à reproduire le recouvrement observé dans la structure bipartie. Ce recouvrement survient lorsque deux nœuds $\bot$ sont connectés à plusieurs nœuds $\top$ (deux auteurs publiant plusieurs articles ensemble par exemple). L'observation extraite de l'étude sur la topologie d'Internet a été étendue à une large variété de réseaux [CT13, TNG14], démontrant que les recouvrements sont communs dans les réseaux bipartis. Comprendre cette caractéristique est donc une préoccupation majeure dans le contexte de l'analyse et la modélisation de tels réseaux.

Afin d'aborder la question de la nature réelle des recouvrements observés dans les réseaux réels, nous analysons la structure de 4 réseaux réels provenant d'activités sociales sur Internet. Ce choix est motivé par le fait que de tels réseaux possèdent une authentique structure bipartie car les plateformes en ligne proposent le plus souvent aux utilisateurs la possibilité de se regrouper : les nœuds $\perp$ définissent les utilisateurs et les nœuds $\top$ représentent les groupes que les utilisateurs rejoignent. Ensuite, nous analysons les types de recouvrements à l'aide de deux métriques standards puis avec de nouvelles métriques que nous avons proposées.

4.1 Jeux de données

Comme expliqué précédemment, de nombreux réseaux sociaux issus du monde réel présentent une structure complexe impliquant deux niveaux. Afin d'avoir une approche permettant de dresser des conclusions qui puissent se généraliser à l'ensemble des réseaux bipartis, nous nous appuyons sur une variété de réseaux sociaux présentant des « communautés » au sens large, qui sont plutôt des groupes représentés par l'ensemble des nœuds $\top$. Nous nous concentrons en particulier sur deux réseaux d'affiliation (Flickr, LiveJournal) et deux réseaux de publications (CiteULike, WikiPedia). Ci-dessous, nous les décrivons brièvement en donnant l'interprétation contextuelle de ce que représentent leurs nœuds $\top$ et $\perp$ respectifs :

- LiveJournal [YL15] : Ce jeu de données concerne un site internet où les utilisateurs peuvent ajouter de l'information grâce à un système de blog. Ce dernier permet également de déclarer des liens d'amitié entre utilisateurs, de créer ou rejoindre des groupes. Ici, les nœuds $\top$ sont les groupes et les nœuds $\perp$ sont les utilisateurs. Il réunit approximativement 1 million d'utilisateurs et plus de 600 000 groupes.

- CiteULike [EC07] : Ce jeu de données provient d'une vaste bibliothèque en ligne qui permet aux utilisateurs de partager, citer ou simplement étiqueter des publications scientifiques. Ici, le jeu de données repose sur à peu près 150 000 tags (nœuds $\top$) et 700 000 publications (nœuds $\perp$).

- WikiPedia [ABHD13] : Ici les nœuds $\perp$ sont des articles de l'encyclopédie WikiPedia et les nœuds $\top$ sont des catégories qui regroupent l'ensemble des articles sur un sujet similaire. Pour des raisons calculatoires, nous n'avons pas utilisé le réseau entier mais une importante composante connexe [1]. Ce sous-réseau implique 3 millions d'articles et presque 500 000 sous-catégories.

[1] Nous avons choisi de nous concentrer sur la composante connexe contenant la catégorie *Network Protocols*.

- FLICKR [YL15] : Ce jeu de données est composé de 100 000 groupes et 400 000 utilisateurs (nœuds $\top$ et $\bot$ respectivement) provenant du site web FLICKR qui permet d'héberger et de partager des photos.

  Plus de détails statistiques sont fournis en Section 4.3.0.1.

4.2 Métriques pour le recouvrement

D'un point de vue local, les propriétés structurelles des nœuds $\top$, dont les définitions sont données dans le Chapitre 3, sont liées à la densité. Or la densité locale peut être analysée de différentes façons. Dans les graphes unipartis, la densité locale est communément définie comme la densité du sous-graphe induit par les voisins d'un sommet. Dans les graphes bipartis, cette densité locale est différente, elle capture habituellement, pour deux sommets voisins (deux voisins $\top$ à distance 2), la proportion de sommets $\bot$ en commun. Deux métriques sont donc proposées pour étudier cette propriété : le coefficient de clustering local (voir Équation 2.9) et le *coefficient de redondance* (voir Équation 2.10 [LMDV08]).

4.2.1 Métriques classiques

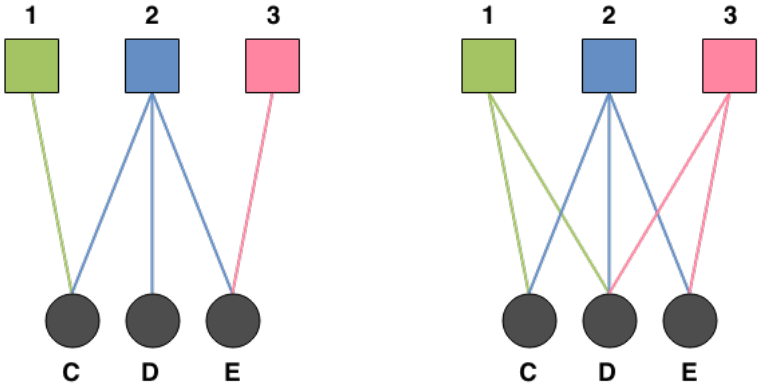
Pour un graphe biparti $\mathbb{B}$, le coefficient de clustering local se concentre sur l'intersection entre le voisinage de deux sommets (voir Section 2.1.2.1 pour plus de détails), tandis que le coefficient de redondance s'intéresse à l'impact qu'aurait le retrait d'un sommet $\bot$ sur la relation entre deux sommets $\top$.

Ainsi le coefficient de clustering local d'un sommet u, noté $CC_4(u)$, est égal à 1 si tous les voisins de u à distance 2 (noté $N^2_\top(u)$) ont tous les mêmes voisins $\bot$ (recouvrement complet), et 0 s'ils n'ont aucun voisins en commun. Si un film $f \in \top$ a $CC_4(f) = 0.5$, cela signifie que, en moyenne, la moitié de acteurs qui ont joué dans f ont rejoué ensemble dans d'autres films. Il est également possible de dériver CC_4 en $CC_4(\mathbb{B})$, le coefficient de clustering global du graphe biparti $\mathbb{B}$, comme étant la valeur moyenne pour tous les sommets (si $CC_4(\mathbb{B}) = 0.2$ alors cela signifie qu'en moyenne 20% des acteurs d'un film se retrouvent à jouer ensemble dans plus qu'un film).

Intuitivement une valeur élevée du coefficient de redondance $RD(v)$ indique que deux sommets voisins de v auront de fortes chances d'être connectés à un autre sommet (deux utilisateurs u_1 et u_2 dans un groupe v seront aussi dans un autre groupe en commun), révélant ainsi la présence de recouvrement dans la structure. Comme pour le

coefficient de clustering, nous pouvons naturellement dériver le coefficient de redondance $RD(\mathbb{B})$ du graphe biparti $\mathbb{B}$, comme la valeur moyenne pour tous les sommets.

En regardant les exemples de la Figure 4.1, nous pouvons estimer la pertinence du coefficient de redondance pour identifier les types de recouvrement. Dans la Figure 4.1(a), il n'y a pas de recouvrement pour le sommet 2, ce qui se traduit par $RD(2) = 0$. À l'inverse, dans la Figure 4.1(b), $RD(2) = 2/3$, reflétant le fait que 2 des 3 possibles relations entre les sommets C, D et E ne sont pas affectées par le retrait du sommet 2 car il y a recouvrement avec les sommets 1 et 3 : C et D sont aussi dans le groupe 1, et D et E sont aussi dans le groupe 3, par contre C et E ne sont ensemble que dans le groupe 2.



(a) Structure sans recouvrement (b) Structure avec recouvrement

FIGURE 4.1: Deux graphes bipartis présentant une structure différente autour du sommet 2 : non redondant dans (a) et fortement redondant dans (b).

4.2.2 Nouvelles métriques

Afin d'affiner les métriques classiques, nous proposons maintenant deux nouveaux coefficients, pour analyser le recouvrement dans la structure d'un graphe biparti. Nous définissons tout d'abord le *coefficient de dispersion* qui se concentre sur la répartition des liens parmi les voisins d'un sommet. Plus précisément, nous calculons le nombre maximal de nœuds $\top$ différents qui pourraient être reliées à l'ensemble des voisins de u. Le *coefficient de dispersion* $disp(u)$ est donc défini par :

$$disp(u) = \frac{|N^2_\top(u)|}{\sum_{v \in N_\top(u)} (|N_\perp(v)| - 1)} = \frac{|N^2_\top(u)|}{\sum_{v \in N_\top(u)} d_\perp(v) - 1} \tag{4.1}$$

Intuitivement, la dispersion d'un sommet $u \in \top$ est totale ($disp(u) = 1$) si le nombre de voisins à distance 2 de u ($|N^2_\top(u)|$) est égal à la somme des degrés (moins 1

pour ne pas compter u) des voisins de u à distance 1. Autrement dit, la dispersion de u est maximale lorsque chaque voisin à distance 2 de u n'est relié qu'à un seul voisin de u. C'est le cas pour le sommet 2 dans la Figure 4.1(a). Notons au passage que cette valeur est directement corrélée avec le fait que la redondance de ce même nœud est égale à 0. Pas de recouvrement implique une dispersion complète des liens. En revanche, comme le lecteur peut le remarquer dans la Figure 4.1(b), quand des recouvrements sont présents, la dispersion tend à diminuer : $disp(2) = \frac{|\{1,3\}|}{1+2+1} = \frac{2}{4}$ car le sommet 2 pourrait avoir au plus 4 voisins à distance 2 mais n'en n'a que 2.

Finalement, nous proposons une autre métrique qui prend en compte la spécificité des sommets de degré 1 qui sont particulièrement nombreux dans les réseaux réels pour l'ensemble $\perp$ (acteurs ayant joué dans un seul film, tags utilisés une seule fois, etc.), ceci impactant l'analyse des recouvrements. Nous définissons le coefficient de monopole comme la proportion de voisins de degré 1 d'un sommet v_i. Formellement :

$$mon(u) = \frac{|\{v \in N_{\top}(u), d_{\perp}(v) = 1\}|}{|N_{\top}(u)|} \tag{4.2}$$

Intuitivement, un sommet $u \in \top$ a un monopole de 1 si ses voisins ne sont connectés qu'à lui et a un monopole de 0 si tous ses voisins sont connectés à au moins un autre sommet. Comme pour les coefficients précédents, nous pouvons bien entendu dériver la dispersion, ou le monopole, d'un graphe biparti comme la valeur moyenne pour tous les sommets. Comme nous verrons dans la prochaine section, ces métriques sont capables de raffiner les résultats apportés par les métriques classiques et peuvent aider à mieux discerner la vraie nature des recouvrements observés dans les réseaux bipartis.

4.3 Analyse de la structure bipartie

L'objectif de cette section est de comprendre le comportement des différentes métriques appliquées sur les quatre jeux de données présentés précédemment. Nous commençons par fournir des statistiques générales pour les différents réseaux (Section 4.3.0.1). Ensuite nous détaillons l'analyse en examinant en particulier la pertinence des coefficients de clustering et de redondance pour caractériser les types de recouvrement dans les réseaux bipartis (Section 4.3.0.2), tout en questionnant si les nouvelles métriques sont capables de préciser l'analyse (Section 4.3.0.3).

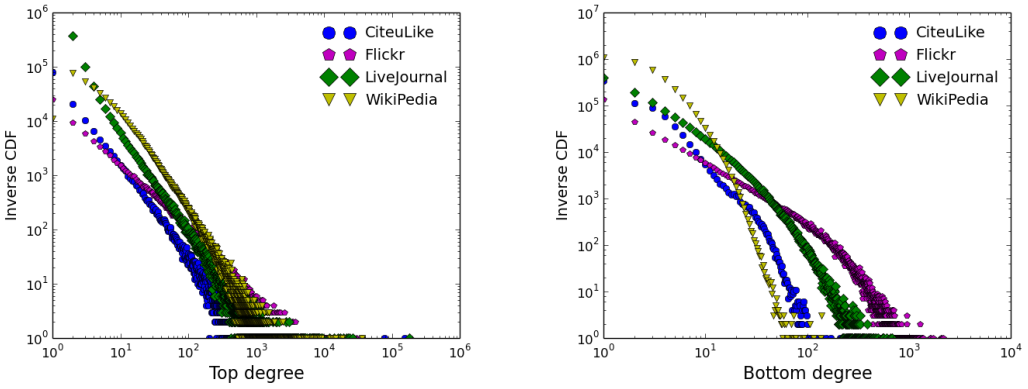
4.3.0.1 Propriétés globales

Tout d'abord, nous nous appuyons sur quelques statistiques des graphes bipartis définies dans la section précédente. Le Tableau 4.1 présente les résultats pour les quatre jeux de données. Il montre que les réseaux étudiés présentent les propriétés statistiques classiques que l'on observe dans les réseaux réels [GL06]. En particulier, on peut remarquer que ces réseaux sont globalement très peu denses ($\delta(\mathbb{B})$) tandis que les densités locales ont des ordres de grandeur plus élevés ($CC_4(\mathbb{B})$ et $RD(\mathbb{B})$). En outre, l'ordre de grandeur entre les degrés moyens ($d_\top(\mathbb{B})$ et $d_\bot(\mathbb{B})$) et les degrés maximaux ($d^+_\top(\mathbb{B})$ et $d^+_\bot(\mathbb{B})$) indique une certaine hétérogénéité dans la distribution de degrés des nœuds. Ceci est parfaitement illustré par la Figure 4.2, en échelle logarithmique, où l'on remarque que les degrés suivent une loi à queue lourde, donc avec une forte hétérogénéité. En ce qui concerne les deux métriques classiques habituellement utilisées pour capturer les recouvrements dans les structures biparties ($CC_4(\mathbb{B})$ et $RD(\mathbb{B})$), on peut remarquer qu'elles diffèrent fortement. En se basant sur le coefficient de clustering, ces réseaux ne semblent présenter que des recouvrements assez faibles alors que, au contraire, le coefficient de redondance tend à indiquer la présence de recouvrements. Le cas de LiveJournal est assez éloquent, montrant un écart important entre ces deux coefficients : $CC_4(\mathbb{B}) = 0.117$ et $RD(\mathbb{B}) = 0.703$. La question se pose alors de savoir lequel de ces deux coefficients est le plus pertinent pour identifier la présence de recouvrements dans la structure. Cette question sera étudiée plus profondément dans la Section 4.3.0.2.

	CiteULike	LiveJournal	WikiPedia	Flickr
$n_\top$	153,3 K	664,4 K	484,5 K	103,6 K
$n_\bot$	731,8 K	1,15 M	3,13 M	396 K
$\delta(\mathbb{B})(*10^{-5})$	2.1	0.9	0.7	20.8
$d_\top(\mathbb{B})$	15.02	10.79	22.31	82.46
$d_\bot(\mathbb{B})$	3.20	6.24	3.46	21.58
$d^+_\top(\mathbb{B})$	153 K	149 K	36 K	35 K
$d^+_\bot(\mathbb{B})$	1,3 K	682	138	2.2 K
$CC_4(\mathbb{B})$	0.138	0.117	0.063	0.055
$RD(\mathbb{B})$	0.521	0.703	0.387	0.646
$disp(\mathbb{B})$	0.725	0.842	0.705	0.769
$mon(\mathbb{B})$	0.071	0.070	0.088	0.148

Table 4.1: Propriétés globales de la structure bipartie des jeux de données. Se reporter à la Section 2.1.1.7 pour les notations. $d^+_\top(\mathbb{B})$ et $d^+_\bot(\mathbb{B})$ sont le degré maximum des nœuds de l'ensemble $\top$ et $\bot$ respectivement.

Si on examine les nouvelles métriques ($disp(\mathbb{B})$ et $mon(\mathbb{B})$), on peut remarquer que le monopole moyen est très faible pour chaque réseau, les noeuds $\top$ ont peu de voisins de degré 1. Ceci semble donc indiquer que globalement les noeuds $\bot$ de degré 1 soient connectés uniformément aux noeuds $\top$. Au contraire, la dispersion moyenne est



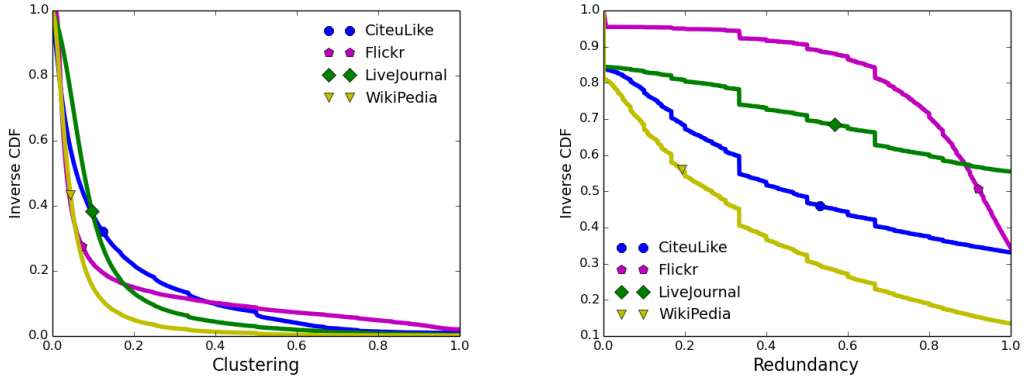
(a) Degrés $\top$ (b) Degrés $\bot$

FIGURE 4.2: Distribution cumulative inverse des degrés des ensembles $\top$ et $\bot$.

haute. Cela semble contredire le coefficient de redondance puisqu'il indique un manque de recouvrements. Mais une telle valeur agrégée est difficile à analyser, une discussion plus détaillée est abordée dans la Section 4.3.0.3.

4.3.0.2 Métriques classiques biparties

Les statistiques globales présentées dans la section précédente ne permettent pas de saisir la diversité des situations possibles pour tous les nœuds. Afin d'affiner cette première analyse, nous avons donc calculé la valeur des différents coefficients pour tous les nœuds, ce qui permet d'étudier la distribution de ces valeurs aussi bien que les corrélations entre les métriques.

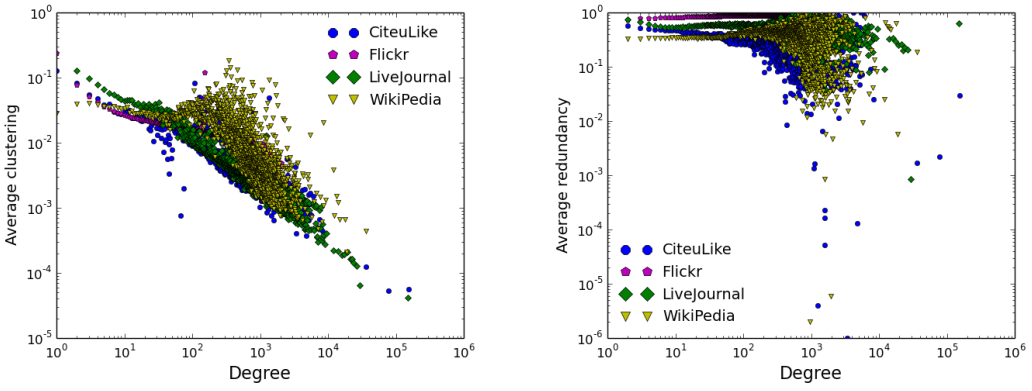


(a) Coefficients de clustering (b) Coefficients de redondance

FIGURE 4.3: Distribution cumulative inverse des coefficients de clustering et de redondance.

La Figure 4.3 présente la distribution cumulative inverse du coefficient de clustering (Figure 4.3(a)) et du coefficient de redondance (Figure 4.3(b)). Nous pouvons observer clairement que les deux coefficients sont très différents. Alors que les distributions décroissent nettement pour le coefficient de clustering, il semble que le coefficient de redondance soit distribué plus uniformément, à l'exception des valeurs extrêmes (0 et 1) qui sont fortement sur-représentées.

Plus important encore, on peut remarquer que la proportion de nœuds avec un faible clustering est haute. Plus de 75% des nœuds ont une valeur inférieure à 0.2 dans tous les jeux de données. Par contraste, la proportion des nœuds redondants est particulièrement haute. La fraction des nœuds ayant une redondance à 1 est par exemple extrêmement importante pour tous les jeux de données : 17% pour WikiPedia, 37% pour Flickr, 38% pour CiteULike et 58% pour LiveJournal. Dans ce dernier cas, cela veut dire que pour plus de la moitié des groupes créés dans LiveJournal, *chaque paire* de membres appartient également a (au moins) un autre groupe en commun. Cela est surprenant, notamment puisque que le nombre moyen de groupes d'un utilisateur est très bas (6.24 sur ce jeu de données, voir Tableau 4.1). Cela indique la présence de recouvrements non triviaux dans ces réseaux qui ne sont pas capturés par le coefficient de clustering, et cela mène donc à la conclusion que la notion de redondance semble mieux adaptée pour détecter les véritables recouvrements dans les réseaux bipartis.



(a) Degré vs. Clustering

(b) Degré vs. Redondance

FIGURE 4.4: Corrélation entre le degré et les coefficients de clustering et de redondance.

Afin de comprendre cette différence entre les deux coefficients, nous présentons Figure 4.4 la corrélation entre le degré d'un nœud et son coefficient de clustering ou de redondance moyen. La Figure 4.4(a) montre un fait éclairant : la valeur du coefficient de clustering en moyenne semble être très fortement corrélée avec le degré du nœud. On observe en effet que plus le degré d'un nœud est élevé, plus son clustering est faible. On

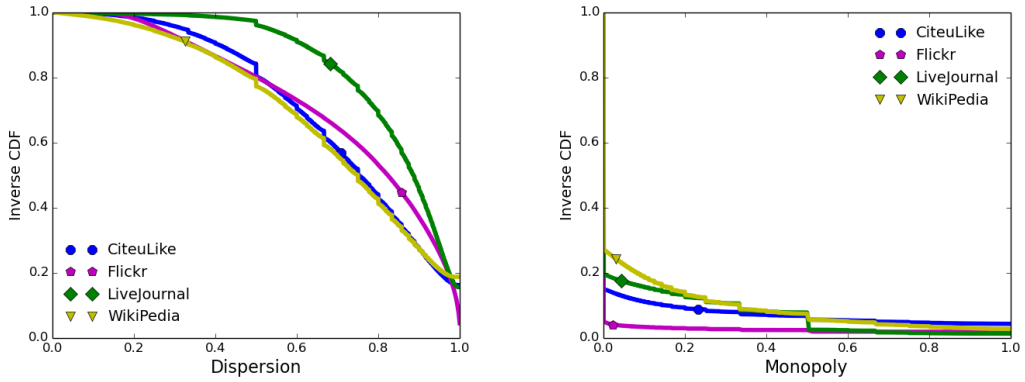
n'observe pas de telle corrélation pour la redondance, Figure 4.4(b), qui semble indépendante du degré du nœud considéré. Cela renforce donc notre première conclusion sur la pertinence de la redondance pour détecter des recouvrements. En effet, la redondance semble être indépendante du degré, tandis que le clustering semble plus lié au nombre de voisins et moins à la *structure* des relations.

4.3.0.3 Affiner l'analyse avec les nouvelles métriques

Les résultats présentés ci-dessus se focalisent sur les métriques classiques proposées comme extensions à la notion de densité locale pour les graphes bipartis. Ils montrent que la notion de redondance semble plus adaptée pour détecter les recouvrements dans les réseaux. Cependant, une telle notion ne nous permet pas de comprendre comment ces recouvrements sont organisés autour d'un nœud. Les notions de dispersion et de monopole proposées ici, sont une tentative d'affiner notre compréhension d'une telle organisation. La Figure 4.5 présente la distribution cumulative inverse des coefficients de dispersion (Figure 4.5(a)) et de monopole (Figure 4.5(b)). Comme suggéré par les valeurs moyennes (voir Tableau 4.1), la dispersion est haute pour une part importante des nœuds. Dans tous les jeux de données, plus de 80% des nœuds ont une valeur supérieure à 0.5. De telles valeurs autorisent cependant la présence de recouvrement. Un coefficient de dispersion à 0.5 signifie que la moitié des liens engendre des relations semblables parmi les nœuds $\top$ et $\bot$, indiquant donc des recouvrements. Paradoxalement, un coefficient de dispersion élevé implique peu de recouvrement, voire aucun si la dispersion est totale. Les valeurs élevées marquent donc un contraste avec celles du coefficient de redondance et réduisent l'importance des recouvrements dans ces réseaux. En ce sens, cela affine l'utilisation de la redondance.

Concernant les coefficients de monopole, les distributions précisent notre déclaration précédente sur le fait que les nœuds $\bot$ de degré 1 sont bien répartis. En effet, cela se vérifie sur la Figure 4.5(b) car toutes les courbes décroissent doucement sur un intervalle de valeurs strictement positives. Cependant, le lecteur peut remarquer l'importante fraction des nœuds n'ayant pas de monopole du tout. Ceci est dû essentiellement à la très faible proportion de nœuds $\bot$ de degré 1 par rapport au nombre de liens dans les réseaux. Bien que les liens soient bien distribués dans le réseau, la probabilité qu'un nœud soit lié à un nœud $\bot$ de degré 1 reste faible, aboutissant alors en général à un coefficient de monopole de 0.

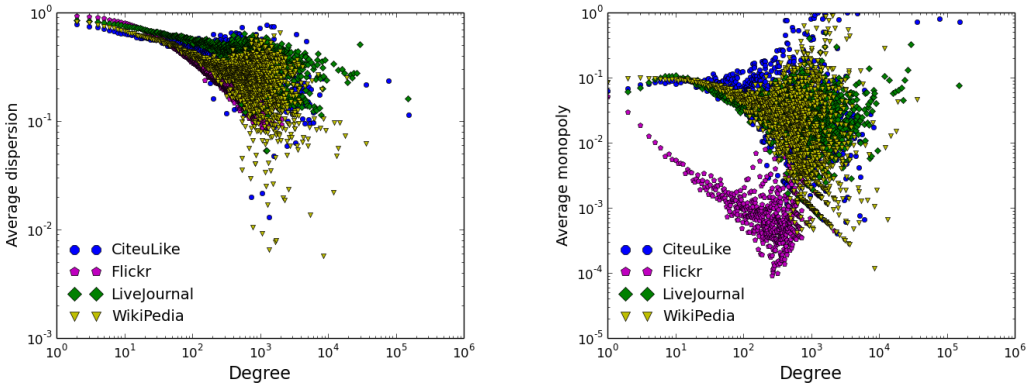
De la même manière que pour le clustering et la redondance, nous pouvons analyser les corrélations entre les métriques. La Figure 4.6 présente la corrélation entre le degré d'un nœud et sa dispersion, ainsi que celle entre son degré et son monopole. La



(a) Coefficients de dispersion (b) Coefficients de monopole

FIGURE 4.5: Distribution cumulative inverse des coefficients de dispersion et de monopole.

Figure 4.6(a) montre une tendance claire : plus le degré est élevé, plus la dispersion est faible en moyenne (notons que l'ordonnée est en échelle logarithmique). Puisque de faibles valeurs pour la dispersion impliquent des valeurs élevées pour la redondance et donc des recouvrements, alors il est possible maintenant de préciser quels types de nœuds sont impliqués dans les recouvrements : ce sont donc les nœuds de degré élevé qui génèrent le plus de recouvrements.



(a) Degré vs. Dispersion (b) Degré vs. Monopole

FIGURE 4.6: Corrélation entre le degré et les coefficients de dispersion et de monopole.

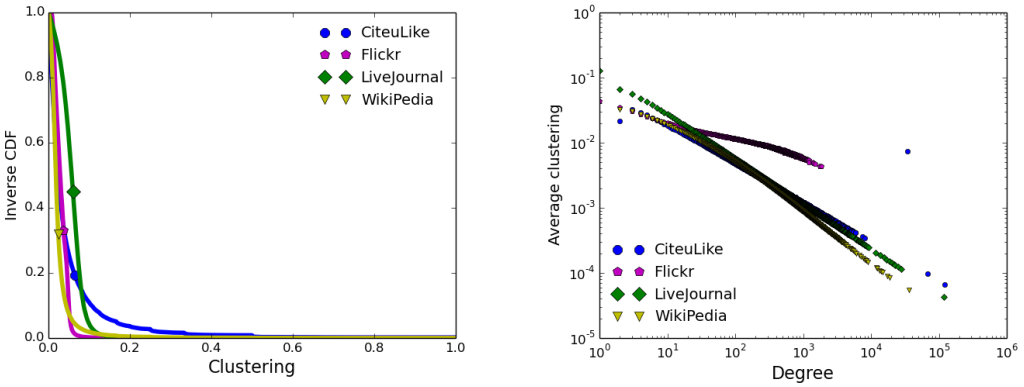
Concernant les corrélations entre le degré et le monopole, la Figure 4.6(b) montre que, mis à part pour le cas de FLICKR qui affiche une décroissance nette lorsque le degré augmente, les valeurs des monopoles semblent indépendantes en moyenne du degré. Notons au passage que la situation de FLICKR est cohérente avec notre précédente

remarque : sachant que les nœuds de degré 1 ne peuvent pas être responsables des recouvrements dans la structure bipartie, il est tout à fait cohérent de faire l'hypothèse que ces recouvrements sont plus présents pour les nœuds de degré élevé.

4.4 Comparaison à l'aléatoire

Après avoir analysé les recouvrements dans les jeux de données en utilisant deux métriques traditionnelles et deux nouvelles métriques, nous étudions l'impact des valeurs obtenues sur ces coefficients à partir de modèles aléatoires. Pour cela, nous utilisons une variante du modèle de configuration (voir Section 2.2.2). Le graphe biparti généré a ainsi le même nombre de nœuds et de liens mais les liens sont distribués uniformément au hasard parmi les nœuds $\top$ et $\bot$, selon leur degré initial. Cette section montre l'impact d'un tel mélange sur les métriques étudiées précédemment.

Les figures ci-dessous (Figures 4.7, 4.8, 4.9, 4.10) montrent les distributions cumulatives inverses des coefficients de clustering, de redondance, de dispersion et de monopole (Figures 4.7(a), 4.8(a), 4.9(a), 4.10(a)) ainsi que les corrélations entre le degré et chacun des coefficients (Figures 4.7(b), 4.8(b), 4.9(b), 4.10(b)), pour des graphes bipartis générés aléatoirement en respectant les distributions de degrés $\top$ et $\bot$ des jeux de données.

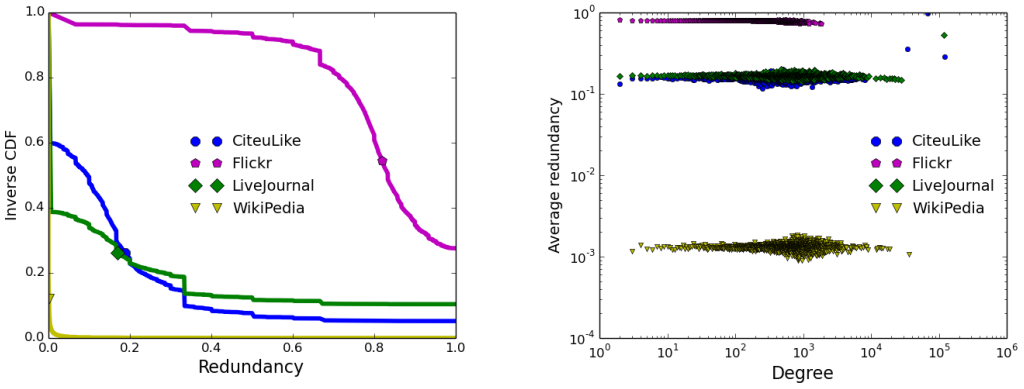


(a) Distribution cumulative inverse du clustering

(b) Corrélation entre degré et clustering

FIGURE 4.7: Distribution cumulative inverse du coefficient de clustering et corrélation entre le degré et le clustering dans les graphes bipartis aléatoires.

Comme attendu, la Figure 4.7 indique que la génération aléatoire a un faible impact sur la distribution du clustering. En effet, la Section 4.3.0.2 a déjà montré que la valeur du clustering d'un sommet est directement affectée par son degré. Sachant que les degrés ont été conservés par le modèle aléatoire alors par transitivité le clustering conserve son comportement antérieur. Cela renforce notre conclusion précédente.



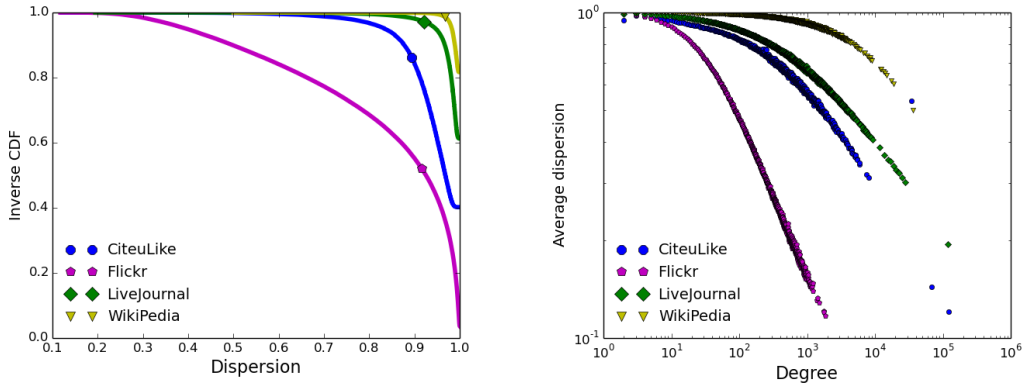
(a) Distribution cumulative inverse et de la redondance (b) Corrélation entre degré et redondance

FIGURE 4.8: Distribution cumulative inverse du coefficient de redondance et corrélation entre le degré et la redondance dans les graphes bipartis aléatoires.

En ce qui concerne la redondance, le comportement est variable selon le jeu de données (Figure 4.8(a)). Le modèle aléatoire semble avoir complètement effacé les recouvrements dans le jeu de données de WIKIPEDIA, en revanche on observe le phénomène inverse pour le réseau de FLICKR. Dans ce dernier, le modèle semble avoir renforcé les recouvrements et il est difficile de se prononcer sur les véritables raisons d'un tel changement. Cependant, comme le lecteur peut remarquer, la densité du réseau de FLICKR est nettement supérieure aux autres (30 fois) et pourrait expliquer cet effet.

Par ailleurs, alors que dans la Figure 4.8(a) on peut remarquer que les jeux de données de WIKIPEDIA et FLICKR ont des distributions des coefficients de redondance très différentes, voire antinomiques, on trouve que dans les graphes aléatoires la redondance est complètement indépendante du degré. Cela est parfaitement naturel car la redondance mesure la proportion de paires (v_1, v_2) de voisins d'un nœud u pour lesquels $N_\perp(v_1) \cap N_\perp(v_2) \neq \{u\}$. Dans le cas d'un graphe aléatoire tous les liens sont choisis aléatoirement et indépendamment et le fait que v_1 et v_2 soient connectés à un autre nœud que u ne dépend que du degré de v_1 et de v_2, et est indépendant de u et de son degré.

Les courbes pour la dispersion (Figure 4.9(a)) indiquent que le modèle aléatoire a renforcé les valeurs dans tous les jeux de données. Cela n'est pas surprenant car la définition du coefficient de dispersion est liée à la notion de la distribution des liens attendus. Autrement dit, en distribuant les liens uniformément au hasard dans les réseaux, le modèle tend à maximiser ce coefficient. La Figure 4.9(b) montre, quant à elle, l'impact du modèle aléatoire sur la corrélation entre degré et dispersion. Étonnement, alors que dans la Section 4.3.0.3 nous avions clairement identifié que le degré influait directement



(a) Distribution cumulative inverse et de la dispersion

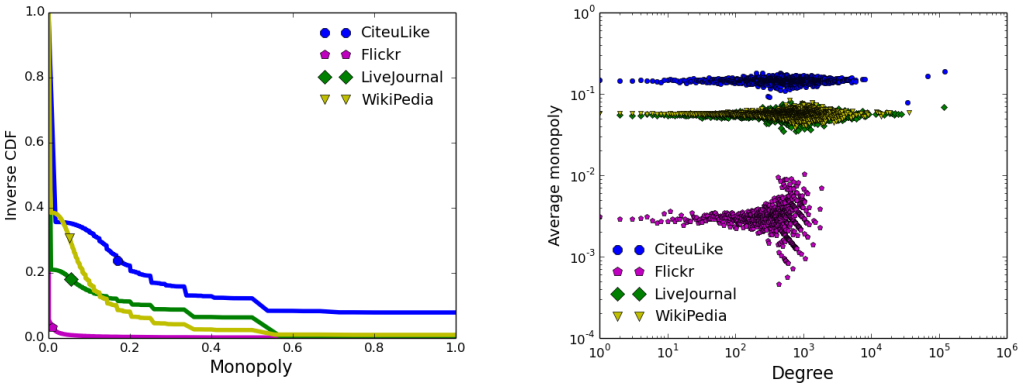
(b) Corrélation entre degré et dispersion

FIGURE 4.9: Distribution cumulative inverse du coefficient de dispersion et corrélation entre le degré et la dispersion dans les graphes bipartis aléatoires.

sur la dispersion dans tous les jeux de données mais semblait très similaire qualitativement et quantitativement sur tous les jeux de données, ici le modèle montre que cette corrélation est due à la nature du réseau puisque après avoir réassigné aléatoirement les liens, la corrélation se trouve fortement affectée. Toutefois, nous pouvons voir que la tendance identifiée précédemment reste correcte : plus le degré est élevé, plus la dispersion est faible. Le graphe étant de taille finie, un nœud de fort degré va être connecté à une proportion non négligeable de sommets $\perp$. Par exemple pour FLICKR il existe un nœud *top* connecté à 10% des sommets $\perp$ et chaque nœud $\perp$ étant connecté à 21 nœuds en moyenne, on peut estimer que le nœud est connecté à tous les autres nœuds au moins 2 fois, d'où une dispersion naturellement très faible. Dans un graphe aléatoire la dispersion décroît donc naturellement en fonction du degré du nœud et de la densité du graphe.

Concernant le monopole, on peut remarquer sur les courbes de la Figure 4.10(a) que le modèle aléatoire n'a pas modifié la tendance générale décrite précédemment (voir Figure 4.5(b)), à savoir qu'une importante fraction des nœuds n'ont pas de monopole sur leurs voisins. Cependant, il est notable d'indiquer que cette fraction est globalement plus faible pour chaque jeu de données. Le cas de WIKIPEDIA est éloquent pour exprimer ce point. En effet, on peut observer que les coefficients de monopole de ce dernier sont repartis dans l'intervalle $]0.0; 0.2]$, confirmant de fait que la génération aléatoire a bien mélangé les nœuds $\perp$ de degré 1 uniformément à travers l'ensemble des nœuds $\top$. Le cas de FLICKR est identique, bien que la proportion de nœuds sans monopole soit bien plus forte, s'expliquant vraisemblablement par la forte densité de ce réseau. La corrélation entre le degré et le monopole, Figure 4.10(b), est naturellement affectée car la proportion de nœuds de degré 1 auquel un nœud de degré k est connecté est indépendant de k. Le

monopole attendu dans un graphe aléatoire est donc égal à la proportion de liens menant à des nœuds de degré 1, c'est-à-dire $\frac{|\{u \in \top | d_{\top}(u)=1\}|}{m}$.



(a) Distribution cumulative inverse et du monopole

(b) Corrélation entre degré et monopole

FIGURE 4.10: Distribution cumulative inverse du coefficient de monopole et corrélation entre le degré et le monopole dans les graphes bipartis aléatoires.

4.5 Discussion

Dans ce chapitre, nous avons étudié les propriétés liées aux recouvrements dans quatre jeux de données provenant de réseaux sociaux d'Internet qui présentent une structure bipartie réelle. Pour cela, nous nous sommes tout d'abord appuyés sur deux métriques proposées pour caractériser la notion de densité locale dans les graphes bipartis, à savoir le coefficient de clustering et le coefficient de redondance. Notre analyse a révélé que le coefficient de clustering n'est pas particulièrement capable de détecter les recouvrements réels de ces réseaux. En effet, l'information capturée par la notion de clustering s'avère peu pertinente et semble être beaucoup plus liée au degré d'un nœud $\top$ qu'à la structure des relations. À l'inverse, le comportement du coefficient de redondance sur ces quatre jeux de données est plus intéressant, il semble beaucoup moins prévisible au regard de simples propriétés locales.

De plus, nous avons proposé de compléter l'analyse des propriétés liées aux recouvrements par l'observation de deux nouvelles métriques, la dispersion et le monopole. Les résultats montrent qu'elles aident à affiner la caractérisation des recouvrements mis en perspective par le coefficient de redondance en donnant des idées sur le type de nœuds impliqués dans les recouvrements. Le coefficient de monopole nous a permis de mettre en évidence que la majeure partie des nœuds $\top$ n'ont aucun monopole et que ce dernier est cependant présent mais seulement au sein de nœuds $\top$ de faible degré, et donc par

conséquent que les recouvrements se situent dans les nœuds $\top$ de fort degrés. La dispersion, quant à elle, nous informe qu'une part importante des nœuds $\top$ ont un coefficient de dispersion élevé. Cela relativise l'importance du coefficient de redondance car la définition de la redondance est contraire à celle de la dispersion, une dispersion trop élevée impliquant automatiquement une redondance faible voire nulle.

Finalement, nous avons appliqué un modèle biparti aléatoire afin d'évaluer comment le remaniement des liens peut affecter les propriétés observées précédemment. Ce dernier a montré qu'un tel processus aléatoire affecte la dispersion des liens en particulier alors qu'il conserve le clustering, renforçant donc l'intérêt de cette nouvelle métrique comme candidat valide pour la caractérisation des recouvrements.

De ce point de vue, le présent travail ouvre la voie à plusieurs améliorations des modèles récents. Il montre en particulier qu'on pourrait améliorer les modèles basés sur les graphes bipartis en intégrant dans de tels modèles les propriétés mises en évidence par le coefficient de redondance, mettant ainsi de côté celui de clustering qui semble moins pertinent. Une approche possible, comme celle suggérée dans [CT13] pourrait consister à coder la redondance dans un troisième niveau afin de contrôler le coefficient par des permutations de liens dans la structure tripartie.

Le présent travail a fait l'objet d'une présentation devant la communauté du domaine, de deux articles acceptés en conférence, un article préliminaire (*Structures bipartites et communautés recouvrantes des graphes de terrains* MARAMI'14), un article plus abouti (*Revealing intricate properties of communities in the bipartite structure of online social networks* IEEE RCIS'15), ainsi que d'un article soumis en journal (RNTI 2015).

Chapitre 5

Découvrir la structure communautaire des réseaux réels en utilisant la notion de cycle et la similarité des nœuds

Sommaire

5.1 Méthodologie 62
5.1.1 L'algorithme ComSim 63
5.1.2 Approches classiques 64
5.2 Évaluation de ComSim 66
5.2.1 Sur des jeux de données avec une vérité de terrain 66
5.2.2 Sur un réseau social d'Internet 68
5.3 Discussion 74

L'étude de la structure des réseaux réels a révélé que les nœuds s'organisent autour de groupes denses, appelés communautés. Généralement, une communauté est décrite comme étant un groupe de nœuds plus densément connectés entre eux qu'ils ne le sont avec le reste du réseau (voir Chapitre 3). L'enjeu est alors de chercher à comprendre comment et pourquoi une structure communautaire émerge de ce type de réseaux. D'un certain point de vue, on peut s'intéresser à savoir quelles sont les propriétés structurelles qui constituent l'organisation du réseau en communautés [New03b]. La plupart des algorithmes pour la détection de communautés cherchent à regrouper les nœuds selon une mesure de densité. Par exemple, l'optimisation de la modularité (voir Section 3.2.1.1) permet d'obtenir des « groupes denses » dans le sens où les membres d'une communauté ont une faible probabilité d'être reliés entre eux dans un réseau où les liens seraient redistribués aléatoirement. La densité peut aussi être capturée par le clustering (voir Section 2.1.1.5), qui comptabilise la proportion de triangles dans le réseau, ou grâce à la recherche de cliques par exemple. Finalement, il existe de multiples manières de formaliser cette définition de communauté, grâce à une métrique ou une heuristique particulière. De plus, il est possible d'envisager que d'autres propriétés (autres que la densité) puissent être à l'origine de la structure communautaire d'un réseau.

Dans cette étude, nous nous intéressons à un autre aspect des réseaux réels, l'assortativité (voir Section 2.1.1.8). Cette notion s'exprime par le fait que les noeuds ont tendance à se relier entre eux selon une certaine propriété. Il peut y avoir diverses façons de formaliser cette notion mais en général la propriété mise en valeur fait référence à la corrélation entre degrés [CLN13]. Ainsi, une forte assortativité signifie que les nœuds de « même degré » ont tendance à se connecter entre eux. Toutefois, l'assortativité peut être considérée plus largement qu'avec le degré, en utilisant par exemple une fonction de similarité (voir Table 3.1). De cette manière, il est nécessaire de délimiter formellement dans quelle mesure on considère qu'il y a une forte assortativité ou non.

L'essentiel des réseaux que l'on rencontre en pratique ont un caractère non-assortatif, c'est le cas des *réseaux techniques*, réseaux du routage d'Internet, réseaux électriques, ou encore les réseaux en biologie par exemple. À l'inverse, il existe une forte assortativité dans les réseaux sociaux en général [New03a]. Néanmoins, la présence de communautés laisse sous-entendre qu'il existe des zones homogènes dans les réseaux réels. Cette intuition permet de mettre en évidence l'existence d'assortativité entre les nœuds d'une même communauté. Par exemple, le réseau des AS (Autonomous Systems) d'Internet est un réseau non-assortatif. Pourtant, il est clairement visible que l'assortativité est « locale » [PPZ09], à la fois au niveau de chaque AS mais aussi entre les nœuds hubs des différents AS (nœuds ayant un degré bien au-dessus de la moyenne). Le premier cas est généralement détectable par les algorithmes classiques de détection de communautés,

car la forte densité interne des AS permet de discerner efficacement ces groupes. À l'inverse, le second cas n'est pas détectable par ces algorithmes car obtenir des communautés constituées seulement de nœuds hubs est difficile lorsque l'on considère la propriété de densité, ces derniers se retrouvent donc éparpillés dans les communautés des AS du premier cas. Cependant, il devient possible de détecter le second cas lorsque l'on considère l'assortativité entre les degrés des nœuds par exemple. Par conséquent, nous proposons de définir une communauté plus largement comme étant un groupe de nœuds partageant les mêmes propriétés.

La principale contribution de ce chapitre est de combiner, pour la détection de communautés, deux notions différentes, les cycles et la similarité entre les nœuds. Cette technique a l'avantage d'être utilisable avec les graphes unipartis, les graphes bipartis, les graphes k-partis. La façon de construire les communautés est également différente des autres méthodes proposées jusqu'ici. Elle permet notamment de mettre en valeur d'autres caractéristiques structurelles du réseau. L'algorithme ComSim[1] proposé ici utilise la similarité entre les sommets d'un graphe pour générer un graphe qui est pondéré. Dans ce nouveau graphe pondéré, la structure du graphe d'origine est modifiée mais le poids attribué aux arêtes permet de conserver une certaine information qui dépend de la mesure de similarité employée. Dans un second temps, l'algorithme s'intéresse à rechercher les cycles présents dans le graphe pondéré. Ces cycles sont ensuite considérés comme des communautés, auxquelles s'ajoutent ensuite d'autres sommets. La qualité des communautés obtenues est évaluée sur des jeux de données empiriques possédant une vérité de terrain (Section 5.2.1), et sur des jeux de données provenant d'Internet pour lesquels une évaluation des propriétés structurelles est donnée (Section 5.2.2).

5.1 Méthodologie

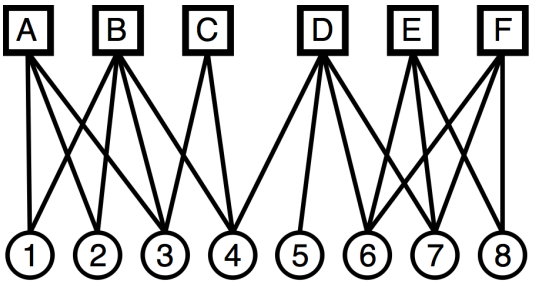


Figure 5.1: Exemple d'un graphe biparti $\mathbb{B} = (\top, \bot, E_B)$.

[1] Le code est disponible en libre accès sur https://github.com/rtackx/ComSim.

5.1.1 L'algorithme ComSim

Dans cette section, nous détaillons le fonctionnement de l'algorithme ComSim sur un graphe biparti. Les procédures décrites ci-dessous sont pareillement utilisables dans le cas d'un graphe uniparti, ou plus largement pour tout graphe k-parti. En effet, ceci est possible car l'algorithme se focalise sur la projection pondérée d'un ensemble de sommets. Cet ensemble peut être tous les sommets d'un graphe uniparti, l'ensemble $\top$ ou $\bot$ d'un graphe biparti, l'ensemble k d'un graphe k-parti.

Pour un graphe biparti, si la $\top$-projection nous intéresse alors on peut étudier comment les sommets $\top$ se connectent à partir de la similarité mesurée par les arêtes reliant les sommets $\bot$ en commun. Formellement, une telle similarité est capturée par une fonction de similarité θ. Ceci permet de définir formellement le poids dans le graphe projeté et pondéré $G_\top = (\top, E, \theta)$ où $\theta : \top \times \top \mapsto \mathbb{R}$ (voir la définition d'un graphe pondéré 2.1.1.1). Ce graphe indique donc la force des relations entre les sommets $\top$. Par exemple, la $\top$-projection du graphe biparti Figure 5.1 est le graphe représenté par la Figure 5.2, si on utilise comme fonction de similarité θ le nombre de voisins communs.

Notons que dans la suite, nous utilisons cette fonction de similarité *voisins communs*, définie formellement pour tout couple $(u, v) \in \top \times \top$ par $\theta(u, v) = |N_\top(u) \cap N_\top(v)|$. Mais il est tout à fait possible d'utiliser d'autres fonctions de similarité (voir Table 3.1) afin de mettre en valeur d'autres relations entre les sommets $\top$ [LHP+16][2].

À partir d'un graphe biparti $\mathbb{B} = (\top, \bot, E_B)$ et d'une fonction de similarité θ, ComSim génère une partition en deux étapes. Premièrement, il identifie les *cœurs de communautés*, qui sont des groupes de sommets très similaires selon la fonction θ. Dans l'Algorithme 1, on peut voir quels sont les détails de cette première étape. L'algorithme génère une chaîne de sommets où les arêtes sortantes ont un poids maximum. Quand la chaîne atteint un sommet qui est déjà présent, cela signifie qu'un cycle a été détecté dans la chaîne. Ce cycle forme ensuite le cœur de la future communauté.

En prenant l'exemple de la Figure 5.2, il s'ensuit que les sommets A et B forment un cycle et sont détectés comme un cœur de communauté, mais aussi les sommets E et F. Ceci est conforme à la Figure 5.2 qui montre que A et B, ainsi que E et F, ont les poids les plus importants. Les autres sommets C et D ne sont pas considérés de suite et sont stockés dans l'ensemble K.

La deuxième étape de l'algorithme essaie ensuite de positionner les sommets restants de K dans les communautés déjà existantes. Ceci est réalisé en cherchant à maximiser la similarité entre ces sommets et tous les sommets des cœurs de communautés.

[2]Selon la fonction de similarité utilisée, il se peut que la projection résulte en un graphe pondéré et dirigé si θ n'est pas symétrique.

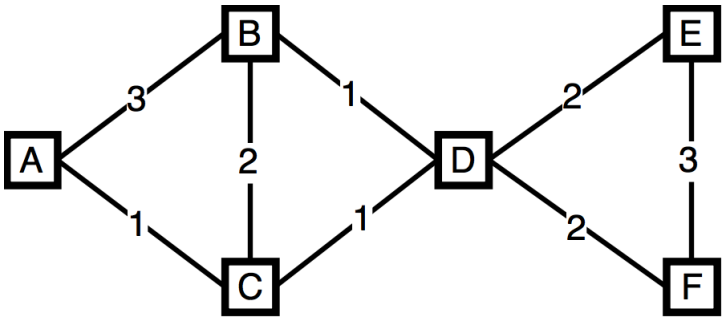


FIGURE 5.2: Exemple de la $\top$-projection pondérée du graphe biparti $\mathbb{B}$ en utilisant la similarité des voisins communs (CN) (voir Équation 3.3).

Comme décrit dans l'Algorithme 2, la deuxième étape se focalise sur tous les sommets qui n'ont pas pris part au partitionnement dans la première étape. Pour chaque sommet x, l'algorithme identifie les communautés qui ont au moins une arête avec x. Ensuite, il choisit la communauté qui maximise la somme des similarités entre x et les membres de cette communauté. Sur l'exemple de la Figure 5.1, indépendemment de l'ordre dans lequel les sommets C et D sont traités dans la deuxième étape, C est affecté à la communauté $C_1 = \{A, B\}$ (la somme des similarités est 3) et D est affecté à la communauté $C_2 = \{E, F\}$ (la somme des similarités est 4).

Il est intéressant de souligner que plusieurs arêtes peuvent avoir le même poids. Les deux étapes peuvent alors se retrouver face à plusieurs options. Dans ce cas, l'algorithme choisit une option prise uniformément et aléatoirement parmi celles possibles. Pour cette raison, l'algorithme est non-déterministe et plusieurs exécution peuvent résulter en différentes partitions.

5.1.2 Approches classiques

Afin d'évaluer la pertinence de ComSim, nous allons comparer les communautés détectées avec celles provenant de trois algorithmes de détection de communautés décrits ci-dessous.

Louvain : L'algorithme de Louvain [BGLL08] (voir Section 3.2.1.1) optimise une fonction de qualité dans le but d'extraire les communautés d'un réseau uniparti. Il est communément utilisé avec la modularité [NG04] qui mesure la densité des communautés par rapport à leur densité si les liens étaient distribués au hasard dans le réseau. Afin d'évaluer les performances de Louvain et pour une comparaison équitable, nous projetons d'abord le graphe biparti sur l'ensemble de sommets $\top$, en générant un graphe pondéré selon la fonction de similarité θ employée (*voisins commun* dans notre cas). Ensuite nous appliquons Louvain sur ce graphe pondéré

Algorithme 1 : ComSim- **première étape**

Data : un graphe biparti $\mathbb{B} = (\top, \bot, E_B)$, une fonction de similarité θ.
Result : une partition P de sommets $\top$ et un ensemble K des sommets restants (pour la **seconde étape**).

$P := \emptyset$ // la partition
$T := \top$ // l'ensemble des sommets à considérer
$x := rand_and_remove(T)$ // sommet aléatoire
$V := \emptyset$ // l'ensemble des sommets en train d'être visité
$K := \emptyset$ // l'ensemble des sommets restants
while $T \neq \emptyset$ **do**
  /* trouve un voisin $y \in N^2_\top(x)$ de x maximisant $\theta(x, y)$ */
  $y := \underset{y \in N^2_\top(x)}{\operatorname{argmax}}\ \theta(x, y)$
  if $y \in V$ **then**
    $C := cycle(V, y, x)$ // extrait le cycle détecté partant de y à x
    $P.add(C)$
    $K := K \cup (V - C)$ // stocke les sommets qui ne sont pas dans le cycle C
    $V := \emptyset$
    $x := rand_and_remove(T)$
  else
    if $y \in T$ **then**
      $V := V \cup \{y\}$
      $x := y$
      $T := T - \{y\}$
    else
      /* y fait déjà parti de l'ensemble P, les sommets visités sont stockés */
      $K := K \cup V$
      $V := \emptyset$
      $x := rand_and_remove(T)$
return P et K

et nous conservons le premier niveau de hiérarchie identifiée afin d'obtenir une résolution fine de la structure communautaire.

Infomap : Infomap est un algorithme récursif, similaire à Louvain, où les communautés sont identifiées en minimisant le taille de la description des sommets visités par un marcheur aléatoire (*map equation* [RAB09], voir Section 3.2.3). Infomap peut tenir compte de la structure bipartie, nous utilisons cette particularité pour générer seulement une partition des sommets $\top$.

lpBRIM : lpBRIM est un algorithme de détection de communautés qui optimise la bimodularité (ou modularité bipartie) [Bar07] qui est une extension de la modularité pour les graphes bipartis (voir Équation 3.2). L'algorithme s'appuie sur l'algorithme BRIM (Bipartite, Recursively Induced Modules) et sur une procédure qui

Algorithme 2 : ComSim- **deuxième étape**

Data : un graphe biparti $\mathbb{B} = (\top, \bot, E_B)$; une partition P ; un ensemble de sommets restants K (provenant de **la première étape**), une fonction de similarité θ.

Result : retourne une partition P' des sommets $\top$ et un ensemble de sommets non satisfaits R.

$R := \emptyset$ `// sommets restants`

$P' := P$

foreach $x \in K$ **do**

  $P_x := com_neigh(x, P)$ `// trouve toutes les communautés voisines de` x

  if $P_x = \emptyset$ **then**

    $R := R \cup \{x\}$

  else

    $C := \underset{C_x \in P_x}{\text{argmax}} \sum_{y \in C_x} \theta(x, y)$

    Ajoute x dans la communauté C de P'

return P' et R

utilise la propagation d'étiquettes (voir Section 3.2.3). Puisque lpBRIM fournit une partition complète d'un graphe biparti – communautés composées des sommets $\top$ et $\bot$ –, nous adaptons l'algorithme et définissons une communauté en ne gardant que ses sommets $\top$ afin de comparer équitablement avec les autres algorithmes.

5.2 Évaluation de ComSim

Cette section est consacrée à l'évaluation de la pertinence de la méthode proposée. Nous commençons par étudier comment se comportent plusieurs algorithmes sur des petits jeux de données dont nous connaissons les communautés (Section 5.2.1) avant de montrer comment ComSim fonctionne sur un réseau social d'Internet de grande taille (Section 5.2.2).

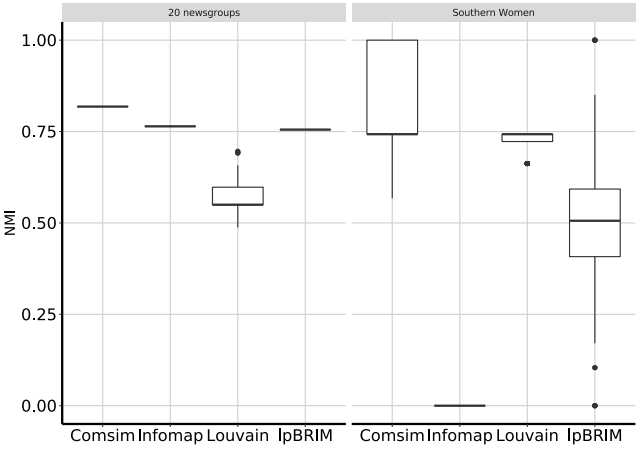
5.2.1 Sur des jeux de données avec une vérité de terrain

Premièrement, nous appliquons notre algorithme sur deux réseaux qui ont une petite taille et qui sont munis d'une structure communautaire fournie par une vérité de terrain. Nous utilisons cette dernière comme référence pour comparer les quatre algorithmes.

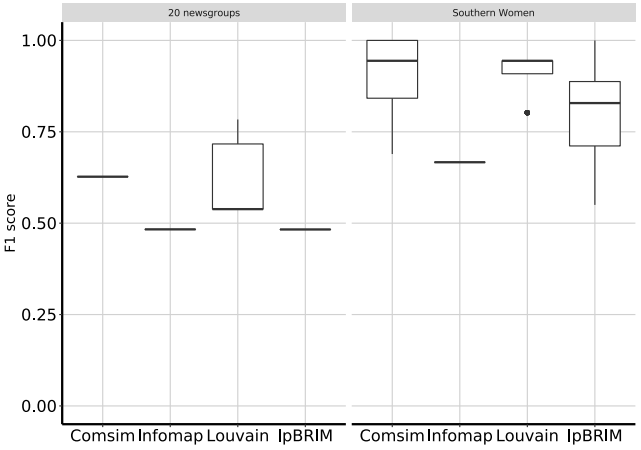
Southern Women [DGG41] Ce réseau représente la participation de 18 femmes ($\top$) à 14 événements ($\bot$) aux États-Unis observée pendant une durée de 9 mois en 1930. Bien que ce jeux de données soit petit, il est intéressant car il a été largement étudié par des sociologues pour comprendre comment les groupes sociaux se forment et évoluent [Gre97,

Fre03]. Dans cette étude, nous utilisons la partition trouvée dans la littérature comme vérité de terrain à laquelle nous confrontons le résultat de quatre algorithmes.

20 newsgroups [Lan95] Ce jeu de données est un enregistrement d'environ 50000 posts soumis par 30000 utilisateurs ($\perp$) à travers 20 groupes de discussion ($\top$). La vérité de terrain est donnée ici par la sémantique des sujets de chaque groupe de discussion, par exemple les groupes de discussion traitant de Sciences sont regroupés dans la même catégorie (*sci.crypt, sci.electronics, sci.med, sci.space*).



(a) NMI



(b) score-F1

FIGURE 5.3: Évaluation de la qualité des partitions détectées par les algorithmes sur *20 newsgroups* et *Southern Women* à l'aide de la NMI (Équation 3.27) et du score-F1 (Équation 3.30).

La Figure 5.3 montre les résultats de COMSIM et des trois algorithmes de référence pour l'évaluation de la détection de communautés sur des réseaux bipartis, décrite Section 5.2. Tous les algorithmes ont été exécutés 100 fois sur Southern Women et 20 newsgroups. Les boîtes à moustaches donnent les valeurs maximales, minimales et les

moyennes. Afin de comparer les communautés de la vérité de terrain à celles détectées par chaque algorithme, nous utilisons premièrement la NMI (Équation 3.27). La Figure 5.3(a) révèle que pour les deux jeux de données, ComSim est l'algorithme qui propose en moyenne la partition la plus similaire à la partition donnée par la vérité de terrain. On remarque que Infomap et lpBRIM retrouvent de bonnes partitions pour 20 newsgroups et Louvain de bonnes partitions pour Southern Women. Il est intéressant de noter que Infomap ne parvient pas du tout à détecter les communautés attendues dans Southern Women.

Une examination manuelle des résultats de Infomap dévoile que l'ensemble des femmes de Southern Women sont en fait regroupées dans une seule communauté, ce qui est bien capturée par la NMI ($NMI = 0$). À l'inverse dans 20 newsgroups, chaque sommet est placé dans une communauté différente, ce qui est complètement surestimé par la NMI ($NMI = 0.7643$).

Afin d'apporter un second point de vue, nous utilisons également le score-F1 (Équation 3.30), une métrique classique pour évaluer la performance des prédictions d'un algorithme (voir Section 3.3.1). La Figure 5.3(b) montre de nouveau que ComSim est le meilleur algorithme pour les deux jeux de données, en moyenne. Curieusement, pour cette métrique, Louvain semble identifier de bonnes partitions en moyenne sur les deux jeux de données. Finalement, il semblerait que ComSim soit capable de proposer des communautés cohérentes vis-à-vis des communautés de la vérité de terrain de ces réseaux bipartis. La prochaine section s'intéresse à étudier le comportement de l'algorithme sur des réseaux de grande taille.

5.2.2 Sur un réseau social d'Internet

Afin de tester la performance de notre algorithme en terme d'efficacité et de qualité, nous nous appuyons sur un large jeux de données extraits de *IMDb* (Internet Movie Database). Ce jeu de données [MDP+11] représente un réseau biparti composé de 118 258 acteurs ($\bot$) ayant joué dans 122 131 films ($\top$) entre les années 1980 et 2013[3].

La Table 5.1 présente la performance en termes de temps d'exécution et de pic de mémoire pour les quatre algorithmes sur les trois jeux de données. On peut voir que Louvain reste le plus efficace en terme de temps et de mémoire, il est seulement un peu plus lent sur le plus petit jeu de données. Cependant, il est important de noter que la performance de Louvain Table 5.1 a été enregistré **après** la $\top$-projection. Cela veut dire qu'une partie de la charge de calculs liée à la fonction θ n'est pas prise en compte, ce qui

[3]Pour une analyse homogène, nous avons retiré les séries et les documentaires de cette période et gardé seulement les 7 premiers acteurs listés dans le casting

	Southern women	20 newsgroups	IMDb
$\|\top\|/\|\bot\|$/liens	18/14/89	20/30K/42K	122K/118K/531K
ComSim	**1.7 ms** / 11.5 MB	1.1 s / 30 MB	33.5 s / 591.6 MB
Infomap	13 ms / 10.7 MB	951 ms / **6.3 MB**	100s / 374 MB
Louvain	11 ms / 6.5 MB	**86 ms** / 10.1 MB	**21 s** / **43 MB**
lpBRIM	6.7 s / **6.1 MB**	74.2 s / 59.7MB	-/-

Table 5.1: Performances en termes de temps d'exécution et de pic de mémoire des quatre algorithmes.

n'est pas le cas pour les autres algorithmes et donc avantage l'approche de Louvain. Sur IMDb en particulier, ComSim est très proche de Louvain en terme de temps et trois fois plus rapide que Infomap.

Les résultats ci-dessus montrent que notre algorithme est capable de passer à l'échelle de grands réseaux mais n'éclairent pas sur la qualité des communautés détectées. Contrairement à la section précédente où une vérité de terrain était disponible, aucune étude n'a été conduite sur IMDb proposant une partition possible des nœuds du réseau. Il n'est donc pas possible d'utiliser ici la NMI ou le score-F1 pour comparer les algorithmes[4].

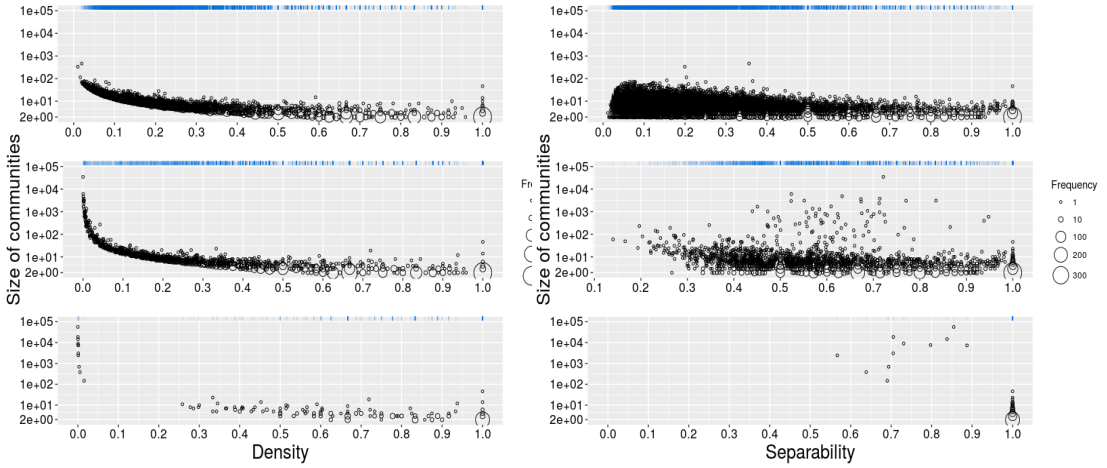
Afin d'évaluer la qualité des communautés détectées, nous reprenons la proposition faite dans [YL15] où les auteurs introduisent deux mesures d'évaluation interne (voir Section 3.3.2), la densité interne et la séparabilité, qui tentent de quantifier la pertinence d'une communauté en terme de connectivité. Plus précisément, la connectivité de la structure de chaque communauté est examinée afin d'observer comment sont réparties les arêtes entre les sommets de la communauté.

Plus formellement, on note P la partition des sommets $\top$, $C_i \in P$ une communauté de P et $\overline{C_i} = \bigcup_{u \in C_i} N_\top(u)$ l'ensemble des sommets $\bot$ induits par le voisinage de C_i. On note également m_{C_i} comme étant le nombre d'arêtes entre C_i et $\overline{C_i}$ (le nombre d'arêtes interne à C_i dans le graphe biparti d'origine) et $m_{\overline{C_i}}$ comme étant le nombre d'arêtes entre $\overline{C_i}$ et C_j (le nombre d'arêtes externes à C_i), où $j \neq i$. Nous redéfinissions donc les Équations 3.32 et 3.33 pour les graphes bipartis :

- Densité interne d'une communauté C_i : $\frac{m_{C_i}}{|C_i|*|\overline{C_i}|}$
- Séparabilité d'une communauté C_i : $\frac{m_{C_i}}{m_{C_i}+m_{\overline{C_i}}}$

La Figure 5.4 présente la distribution de la densité interne (Figure 5.4(a)) et de la séparabilité (Figure 5.4(b)) des communautés selon leur taille et pour les trois algorithmes (ComSim en haut, Louvain au milieu et Infomap en bas). Ces résultats montrent que

[4]Puisque lpBRIM ne passe pas à l'échelle sur IMDb, nous ne le mentionnons pas par la suite.



(a) Densité interne (b) Séparabilité

FIGURE 5.4: Ces nuages de points montrent la relation entre les propriétés des communautés (en terme de densité interne et de séparabilité) et leurs tailles pour COMSIM (haut), Louvain (milieu) et Infomap (bas) sur IMDb.

Infomap, pour la plupart des communautés détectées, ne parvient pas à retrouver des communautés cohérentes du point de vue sens de leur connectivité. En réalité, bien que la plupart des communautés ont de très hautes valeurs pour ces deux propriétés, elles concernent seulement les communautés de petite taille. Mais pour les communautés de grande taille, les indicateurs chutent. Ceci est particulièrement visible sur la Figure 5.4(a), où l'on peut voir deux parties distinctes sur la distribution. À cet égard, il est frappant de remarquer que la plus grande communauté identifiée par Infomap réunit plus de 46% des nœuds du réseau. La même observation peut être faite pour Louvain mais dans une moindre mesure avec une communauté impliquant 29% des nœuds. Comparé à Louvain et Infomap, COMSIM détecte des communautés plus équilibrées en terme de taille. La plus grande communauté est plutôt petite (seulement 1% des nœuds) tandis que la densité est comparable à celle de Louvain (voir Figure 5.4(a)). Cependant, concernant la séparabilité, les valeurs sont faibles comparées à Louvain, ce qui indique que la qualité des partitions pourrait être améliorée pour cette propriété.

Finalement, ces quatre indicateurs permettent d'évaluer la qualité d'une communauté par rapport aux arêtes internes et externes, et donc ils se rapportent à la densité structurelle. Toutefois, notre intuition de départ est de produire des communautés qui soient cohérentes dans un sens plus large, sans obligatoirement se rapporter à un critère de densité. Dans la section suivante, nous nous aidons d'attributs disponibles sur les sommets pour évaluer, de manière non-structurelle, les communautés détectées par les algorithmes.

5.2.2.1 Homogénéité des communautés détectées

Dans la section précédente, nous avons analysé deux propriétés structurelles des communautés détectées par trois algorithmes. Afin d'étudier plus finement les résultats fournis par ces algorithmes, nous construisons une évaluation qualitative qui s'appuie sur les attributs des films.

En effet, il est possible d'attribuer à chaque film du réseau un ou plusieurs genres, pays ou langues grâce aux données de IMDb. Ces informations supplémentaires permettent d'organiser les films autour de certaines ressemblances ou *catégories*, la trame narrative pour le genre, le lieu de production pour le pays, le langage employé pour la langue. Une catégorie prise individuellement peut être incluse dans un graphe triparti qui est construit en ajoutant, au graphe biparti initial, un troisième niveau (voir l'ensemble V_3 sur la Figure 5.5). Ce dernier consiste en un nouvel ensemble de sommets donnés par les attributs d'une catégorie et en un nouvel ensemble d'arêtes reliant les attributs aux films concernés. Par exemple, la catégorie "Genres" a comme attributs *drame, comédie, documentaire, aventure, science-fiction, etc.*

Un graphe triparti est noté $\mathbb{T} = (V_1, V_2, V_3, E_T)$ où V_1, V_2, V_3 sont les ensembles du niveau 1, 2 et 3 respectivement et $E_T \subseteq (V_1 \times V_2) \cup (V_2 \times V_3)$ est l'ensemble des arêtes reliant les sommets de l'ensemble V_1 aux sommets de l'ensemble V_2 et les sommets de l'ensemble V_2 aux sommets de l'ensemble V_3. Ici nous considérons seulement les arêtes entre deux niveaux consécutifs, soit $E_{T'} \subseteq V_1 \times V_2$ et $E_{T''} \subseteq V_2 \times V_3$ avec $E_T = E_{T'} \cup E_{T''}$ (voir Figure 5.5).
Ainsi, le jeu de données de IMDb peut être modélisé par un graphe triparti $\mathbb{T} = (V_1, V_2, V_3, E_T)$ dans lequel les sommets de V_1, V_2 et V_3 sont respectivement les acteurs, les films et les attributs d'une catégorie choisie. Une arête dans $E_{T'} \subseteq V_1 \times V_2$ représente un acteur jouant dans un film, une arête dans $E_{T''} \subseteq V_2 \times V_3$ représente un film ayant un attribut.

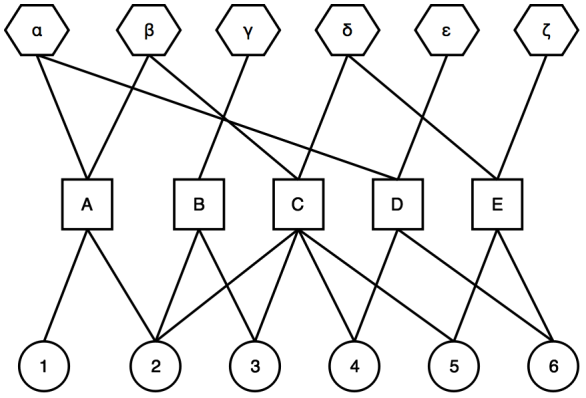


FIGURE 5.5: Exemple d'un graphe triparti $\mathbb{T} = (V_1, V_2, V_3, E_T)$ avec $V_1 = \{1, 2, 3, 4, 5, 6\}$, $V_2 = \{A, B, C, D, E\}$ et $V_3 = \{\alpha, \beta, \gamma, \delta, \epsilon, \zeta\}$.

Afin d'évaluer la qualité des communautés détectées par ComSim, Louvain et Infomap, nous nous appuyons sur l'indice de Herfindahl-Hirschmann [Rho93], une métrique provenant de la microéconomie qui mesure la concentration ou le degré de monopole du marché. Elle est noté :

$$IHH = \sum_{i=1}^{n} s_i^2$$

où n est le nombre d'entreprises et s_i la part du marché de l'entreprise i. Le marché est dit concurrentiel lorsque IHH tend vers $1/n$ et à l'inverse IHH tend vers 1 lorsque le marché est monopolistique.

Afin de rendre cet indice compatible à notre contexte, nous proposons un *score d'homogénéité* qui permet de rendre compte de l'homogénéité des attributs dans une communauté constituée de films. Cette métrique se base sur le formalisme des graphes pour calculer à quel point les films d'une communauté ont des attributs similaires (valeur de 1) ou des attributs différents (valeur proche de 0). Pour une communauté $C_i \subseteq V_2$, elle est définie par :

$$H(C_i) = \sum_{a \in N_3(C_i)} \Big(\frac{|\{u \in C_i | a \in N_3(u)\}|}{d_3(C_i)} \Big)^2 \tag{5.1}$$

avec

$$N_3(u) = \{a \in V_3 | \{u, a\} \in E_{T''}\} \text{ pour } u \in V_2$$

$$N_3(C_i) = \bigcup_{u \in C_i} N_3(u) = \{a \in V_3 | \exists u \in C_i, \{u, a\} \in E_{T''}\}$$

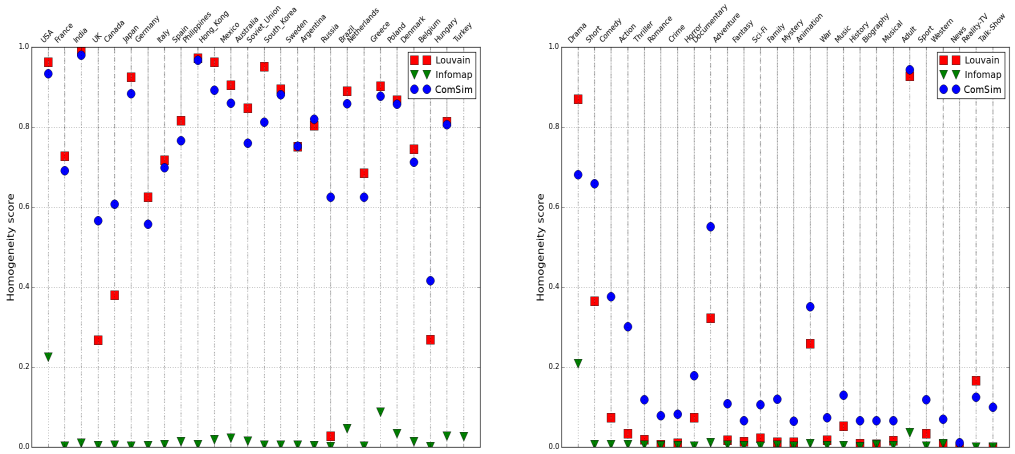
$$d_3(u) = |\{a \in V_3 | \{u, a\} \in E_{T''}\}|$$

$$d_3(C_i) = \textstyle\sum_{u \in C_i} d_3(u)$$

Ainsi, pour chaque attribut présent dans une communauté, on calcule la fraction de films de cette communauté ayant cet attribut parmi l'ensemble des attributs présents dans la communauté.

Un score d'homogénéité élevé indique donc que les attributs liés aux films d'une communauté sont répartis de façon homogène. Toutefois, nous souhaitons obtenir un score d'homogénéité pour chaque attribut d'une catégorie et ainsi déterminer si les algorithmes de détection de communautés regroupent les films par attributs. Pour cela, nous calculons, pour un attribut donné a_i, la moyenne des scores d'homogénéité parmi l'ensemble des communautés ayant au moins un film avec cet attribut a_i (voir Figure 5.6).

La Figure 5.6(b) montrent que ComSim est l'algorithme qui crée les communautés les plus homogènes pour la catégorie Genres. Sur la Figure 5.6(a) même si Louvain paraît



(a) Pays (b) Genres

FIGURE 5.6: Scores d'homogénéité normalisés par attribut pour les catégories Pays et Genres.

meilleur que COMSIM pour la plupart des attributs, en vérité sa moyenne est de $0,7084$ alors que COMSIM a une moyenne de $0,7406$. Cet écart peut être expliqué par le fait que COMSIM identifie plus de communautés de petite taille que Louvain, ce qui tend à créer une plus forte homogénéité. À l'inverse, Infomap identifie plutôt des communautés de très grande taille, ce qui signifie par conséquent que ces communautés ne sont pas homogènes car elles sont constituées de films avec des attributs très différents.

Nous observons également que certains attributs, comme "USA", "Inde" et "Hong-Kong" de la catégorie Pays (Figure 5.6(a)) ou "Drama" et "Adult" de la catégorie Genres (Figure 5.6(b)) sont attachés à des communautés très homogènes. En effet, il est fréquent que les acteurs Américains, Indiens ou d'Hong-Kong collaborent dans des films avec d'autres acteurs de leurs pays. Il en va de même pour la catégorie Drames et Adultes. En d'autres termes, les communautés détectées par COMSIM et Louvain semblent être constituées de films qui ont tendance à partager des attributs similaires. Globalement, ces communautés sont homogènes dans ce sens et permettent de mettre en perspective des mécanismes sous-jacents liés à l'organisation des films et à la collaboration entre acteurs par exemple.

On peut alors se demander comment expliquer les différences entre les algorithmes et leurs résultats. COMSIM et Louvain semblent être comparables en termes de densité interne et d'homogénéité et génèrent tout deux une nouvelle structure grâce à la fonction de similarité θ, ce qui n'est pas le cas de Infomap. Louvain et Infomap ont comme point commun d'identifier des communautés de grande taille, à l'inverse de COMSIM qui identifie des communautés de petite taille. En prenant en compte ces différents points, il serait intéressant d'établir un lien entre les résultats obtenus et les caractéristiques

des algorithmes. Une première hypothèse est que la fonction de similarité θ joue un rôle important dans l'homogénéité des communautés car elle est responsable de la projection du graphe, et donc de la modification de sa structure.

5.3 Discussion

Dans cette étude, nous proposons ComSim, un nouvel algorithme pour la détection de communautés dans les graphes unipartis, bipartis et k-partis. Cette approche génère une partition de nœuds en s'appuyant sur la similarité entre les nœuds du réseau pour créer un graphe pondéré. À partir de ce dernier, ComSim cherche à trouver et maximiser les cycles dans les relations entre les nœuds. Ceci définit les cœurs de communautés qui sont enrichis avec d'autres nœuds lors de la deuxième étape de l'algorithme.

Nous avons implémenté et appliqué cet algorithme sur 3 jeux de données bipartis et nous avons comparé les partitions générées avec celles proposées par trois algorithmes standards de détection de communautés, Louvain, Infomap et lpBRIM. Les résultats montrent que, sur des petits réseaux pour lesquels nous connaissons les bonnes partitions, ComSim est l'algorithme qui génère les meilleures communautés.

De plus, ComSim a prouvé être capable de passer à l'échelle sur un jeux de données d'Internet avec un temps d'exécution proche de Louvain. Nous avons pour cela étudié les partitions générées par ComSim, Louvain et Infomap. Nous montrons que les communautés détectées par ComSim sont plus équilibrées en termes de taille, tout en conservant des indicateurs qualitatifs raisonnables et comparables à ceux de Louvain par exemple. Afin d'affiner l'évaluation sur ce jeu de données, nous avons cherché à montrer que ComSim permet aussi de détecter des communautés homogènes en utilisant des attributs disponible sur les nœuds. À l'aide d'une métrique baptisée score d'homogénéité, nous avons montré que ComSim tend à regrouper les nœuds qui ont des attributs similaires.

Par ailleurs, il aurait été possible de confronter ComSim avec d'autres algorithmes de la littérature, par exemple biSBM [LCJ14] même s'il n'est pas totalement adapté à ce contexte car il nécessite de fournir un nombre de communautés attendues.
En fin de compte, cette étude propose une approche pertinente pour la détection de communautés en se focalisant sur des propriétés généralement peu utilisées. Ceci pourrait mener à une étude plus approfondie pour un travail futur.

Le présent travail a donné lieu à une communication pendant la journée d'étude Thyrex (Fondements théoriques des réseaux mulitplexes) et à un article accepté à la conférence ComplexNetworks 2017.

Chapitre 6

Structure communautaire des réseaux multi-couches

Sommaire

6.1 Représentation et résolution du problème 78
6.2 Données 79
6.2.1 Génération de réseaux artificiels 79
6.3 Développement d'algorithmes de détection de communautés 82
6.3.1 Propriétés souhaitables 82
6.3.2 Modularité multi-couche 82
6.3.3 Algorithmes d'optimisation 85
6.4 Evaluation de Q_M 86
6.4.1 Résultats expérimentaux 87
6.5 Évaluation sur les réseaux artificiels 88
6.5.1 Protocole expérimental 88
6.5.2 Algorithmes en compétition 89
6.5.3 Évaluation 89
6.6 Évaluation sur des réseaux réels 91
6.6.1 Yelp 91
6.6.2 Meetup 93
6.7 Discussion 94

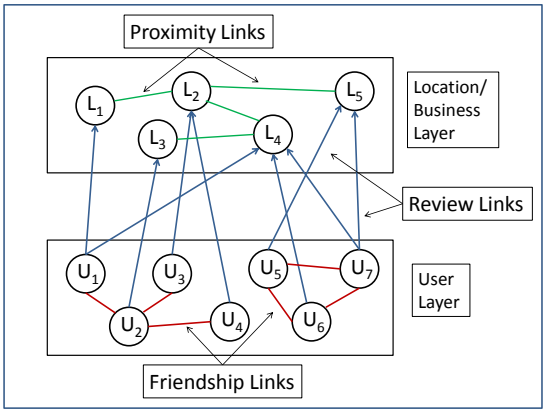


FIGURE 6.1: Illustration d'un réseau (Yelp) multi-couches.

Une communauté est définie comme un groupe de sommets plus densément connectés en interne que vers le reste du réseau (voir Chapitre 3). Le but de la détection de communautés est d'identifier de tels groupes. Une littérature abondante existe sur le sujet [For10]. Cependant, de nombreux réseaux réels tels que des réseaux de communication, sociaux, d'infrastructures ou biologiques, peuvent être modélisés par des réseaux multi-couches [BPP+10]. Un réseau multi-couches est composé de plusieurs réseaux interdépendants où chaque couche représente un type d'interaction. De plus, un sommet dans une couche dépend des sommets dans les autres couches. Par exemple, un réseau social basé sur les lieux tel que Yelp peut être représenté par un réseau multi-couches (voir Figure 6.1) dans lequel une couche contient les clients connectés par des liens sociaux et une seconde couche contient des lieux connectés par des liens de proximité (géographique, thématique ou autre). Les liens de couplage, ou *liens bipartis*, connectent des clients aux lieux qu'ils ont visités.

La détection de communautés dans les réseaux multi-couches est un problème de recherche important. Les communautés de ces réseaux peuvent permettre d'identifier des groupes cohésifs et de révéler des interactions complexes entre des sommets de types différents et des liens hétérogènes. Elles peuvent aussi servir à résoudre des problèmes de fouille de données tels que la prédiction, la recommandation, la recherche contextuelle, etc. [LSC+09].

La détection de communautés dans les réseaux multi-couches pose de nouvelles questions car les communautés cherchées peuvent contenir un seul type de sommets ou plusieurs. La plupart des études récentes se sont concentrées sur les réseaux multiplexes [MRM+10, KM15] dans lesquels toutes les couches possèdent les mêmes sommets mais chaque couche correspond à un type d'interaction. Dans ce contexte, certaines approches proposent de nouvelles métriques pour mesurer la qualité des communautés [MRM+10] alors que d'autres utilisent des marches aléatoires [KM15] ou des

techniques de recherche de motifs fréquents [BPC13] pour obtenir des groupes structurellement similaires. En général, la plupart des approches mentionnées transforment le problème en celui de détection de communautés dans un réseau mono-couche en se basant sur le fait que les liens entre les couches ne connectent que les copies d'un même sommet. Dans le cas multi-couches, le fait que les sommets soient différents sur chaque couche et que les liens entre ces couches (appelés liens de couplage) aient une signification plus large empêche d'utiliser les méthodes développées dans le cas multiplexe.

Malgré tout, ce problème n'est pas sans solution et différentes études s'y sont attaquées en utilisant des méthodes telles que les processus de Dirichlet [STH+14], la factorisation de tenseurs [LSC+09], le clustering de sous-espaces [DFVN14], la factorisation de matrices non négatives [CZG+13], etc. Cependant, la plupart de ces approches ont des limitations. Tout d'abord, certaines ne s'appliquent que sur des réseaux multi-couches spécifiques (par exemple, en étoile [STH+14]). Certaines encore imposent de détecter des communautés contenant différents types de sommets [LSC+09] ce qui introduit une limitation. D'autre part, le nombre de communautés à détecter doit souvent être fixé a priori pour la plupart d'entre elles [LSC+09, DFVN14] ce qui limite leur capacité à découvrir l'ensemble réel des communautés. Enfin, aucune méthodologie pour créer des réseaux avec des communautés artificielles n'est disponible, ce qui est pourtant souvent essentiel pour évaluer ce type d'algorithmes.

Des travaux récents se sont tournés vers le développement d'une modularité pour les réseaux hétérogènes. Par exemple, la modularité composite [LLMW14] évalue la modularité d'un réseau avec différents types de relations en intégrant les modularités des réseaux associés à chaque type de relation. Cependant, cette modularité composite ne peut trouver que des communautés contenant un type de sommets. A l'opposé, la modularité proposée dans [STL+15] pour des réseaux de gènes et d'interactions chimiques ne parvient pas à utiliser les liens de couplage pour caractériser les communautés. Pour résumer, l'état de l'art ne parvient pas à l'heure actuelle à proposer des méthodes qui (a) n'ont pas de paramètre externe à fixer tel que le nombre de communautés et (b) ne sont pas dédiées à trouver des communautés ne contenant qu'un type de sommet ou au contraire des sommets de tous les types. Développer une modularité évitant ces problèmes peut donc être un premier pas dans cette direction.

Dans ce chapitre, nous proposons un algorithme de détection de communautés pour les réseaux multi-couches capable de détecter des communautés comprenant à la fois un ou plusieurs types de sommets selon la structure du réseau et sans connaissance a priori du nombre de communautés. Notre travail est structuré comme suit : nous définissons tout d'abord la notion de réseau multi-couches et spécifions formellement le problème de la détection de communautés (Section 6.1). Ensuite, nous proposons

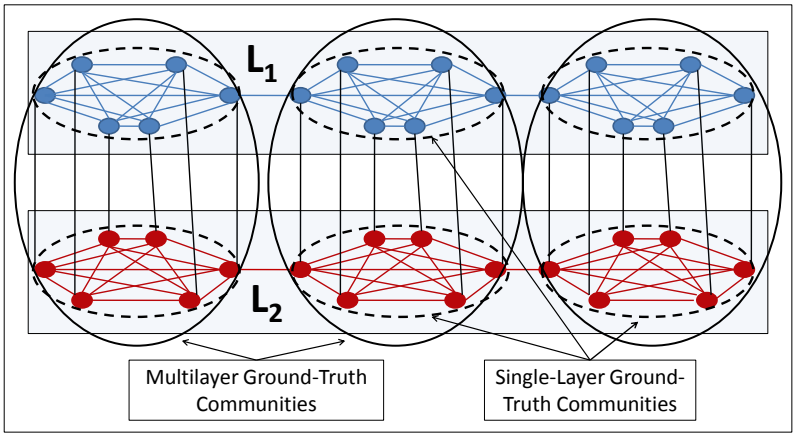


FIGURE 6.2: Configurations avec deux types de communautés, communauté multi-couches et communautés mono-couches.

et évaluons une méthodologie pour construire des réseaux multi-couches synthétiques avec une vérité de terrain (Section 6.2). La contribution principale est la proposition d'une modularité Q_M pour caractériser la qualité de communautés dans ce contexte. Nous proposons deux algorithmes simples pour l'optimiser **GN-**Q_M et **Louvain-**Q_M qui intègrent l'index de modularité Q_M (Section 6.3). Puis nous évaluons la pertinence de la modularité (Section 6.4) et les performances des algorithmes contre l'état de l'art avec des expériences contrôlées sur des réseaux synthétiques (Section 6.5) ainsi que sur des données empiriques (Yelp et Meetup) afin de montrer que les algorithmes proposés obtiennent de meilleures performances que l'état de l'art (Section 6.6).

6.1 Représentation et résolution du problème

Un réseau multi-couches est un couple $\mathcal{G} = (\mathcal{G}_\mathcal{U}, \mathcal{G}_\mathcal{B})$ où $\mathcal{G}_\mathcal{U} = \{L_i : i \in \{1, 2, \ldots, M\}\}$ est un ensemble de M réseaux unipartis (appelés couches de $\mathcal{G}$) et $\mathcal{G}_\mathcal{B} = \{L_{ij} : i, j \in \{1, 2, \ldots, M\}, i \neq j\}$ est un ensemble de réseaux bipartis contenant des sommets des couches et des liens entre eux. Pour chaque couche $L_i = (V_i, E_i)$, V_i et E_i sont respectivement les ensembles de sommets et les liens intra-couches présents dans L_i. De même, L_{ij} est un triplet (V_i, V_j, E_{ij}) où $\{E_{ij} \subseteq \{V_i \times V_j\} : \ i, j \in \{1, 2, \ldots, M\}, i \neq j\}$ est l'ensemble des liens de couplage entre les couches L_i et L_j.

Définition : une communauté C dans un réseau multi-couches $\mathcal{G}$ est définie comme un groupe cohésif $(\mathcal{C}_\mathcal{U}, \mathcal{C}_\mathcal{B})$ de $\mathcal{G}$ contenant des sommets d'une ou plusieurs couches ainsi que tous les liens ayant leurs deux extrémités dans le groupe. Formellement, $\mathcal{C}_\mathcal{U}$ et $\mathcal{C}_\mathcal{B}$ sont définis par $\mathcal{C}_\mathcal{U} = \{L_i^C = (V_i^C, E_i^C) : V_i^C \subseteq V_i, E_i^C = \{E_i \cap (V_i^C \times V_i^C)\}, i \in$

$\{1, 2, \ldots, M\}\}$ *et* $\mathcal{C}_{\mathcal{B}} = \{L_{ij}^C = (V_i^C, V_j^C, E_{ij}^C) : V_i^C \subseteq V_i, V_j^C \subseteq V_j, E_{ij}^C = \{E_{ij} \cap (V_i^C \times V_j^C)\}, i, j \in \{1, 2, \ldots, M\}, i \neq j\}$.

Les communautés dans un réseau multi-couches $\mathcal{G}$ sont de deux types (voir Figure. 6.2) : (a) des communautés multi-couches contenant des sommets de plusieurs couches ; pour lesquels $\mathcal{C}_{\mathcal{B}} \neq \emptyset$; (b) des communautés mono-couches pour lesquelles $\mathcal{C}_{\mathcal{B}} = \emptyset$.

Le problème de la détection de communautés multi-couches consiste à partitionner le réseau $\mathcal{G}$ en un ensemble de groupes disjoints $C_1, C_2, \ldots, C_K$ qui couvre les sommets de $\mathcal{G}$ de sorte que chaque groupe C_i soit composé de sommets fortement connectés entre eux et peu vers l'extérieur.

Les problèmes principaux sont (a) de gérer des réseaux multi-couches contenant différents types de liens et de sommets avec des densités variables et (b) de détecter à la fois des communautés mono-couches et multi-couches sans paramètre additionnel. Le problème peut bien entendu être généralisé au cas des communautés recouvrantes, dynamiques, etc. comme dans le cas des réseaux unipartis.

6.2 Données

6.2.1 Génération de réseaux artificiels

Nous proposons ici une méthodologie pour générer des réseaux multi-couches aléatoires avec vérité de terrain en utilisant le modèle LFR [LFR08]. Différents paramètres sont utilisés pour générer différents types de réseaux. Le réseau contient M couches, la couche L_i ayant N_i sommets ($N_i = |V_i|$) avec un degré moyen $\langle k_i \rangle$. Le paramètre α régule la proportion de liens de couplage vs liens intra-couches. La génération suit trois phases :

Phase A. Communautés mono-couches : Les communautés de chaque couche sont générées avec LFR. Chaque couche L_i a N_i sommets et les distributions de degrés et de taille de communautés suivent une loi de puissance d'exposants γ_i et β_i respectivement. Le paramètre de mélange μ_i de LFR est fixé de sorte à avoir $|\mathcal{C}_i|$ communautés dans la couche L_i.

Phase B. Communautés multi-couches : Les communautés $x_i \in \mathcal{C}_i$ de L_i et $x_j \in \mathcal{C}_j$ de L_j créent la communauté multi-couches x_{ij}. Si $|\mathcal{C}_i|$ et $|\mathcal{C}_j|$ correspondent au nombre de communautés dans les couches L_i et L_j respectivement, $|\mathcal{C}_c| = min\{|\mathcal{C}_i|, |\mathcal{C}_j|\}$ est le nombre maximum de communautés multi-couches. Nous construisons $(|\mathcal{C}_c| \times \alpha)$

communautés multi-couches en combinant des communautés mono-couches L_i et L_j. Chaque communauté x_{ij} peut ainsi contenir une ou plusieurs communauté mono-couche.

Phase C. Liens de couplage : Enfin, nous créons des liens de couplage entre les couches L_i et L_j avec une densité d_{ij}. La fraction p représente le paramètre de mélange pour les communautés multi-couches. Nous distribuons $(N_i \times N_j \times d_{ij})$ liens de couplage entre les couches L_i et L_j, puis nous recâblons ces liens de sorte qu'une fraction p de ces liens soient dans les communautés multi-couches et une fraction $1 - p$ connecte des communautés différentes.

Afin de savoir si les réseaux artificiel sont pertinents, nous procédons en trois étapes :

1. Nous générons des réseaux artificiels en faisant varier μ, α et p.
2. Nous appliquons les méthodes de l'état de l'art pour détecter des communautés multi-couches [LSC+09, LLMW14] sur ces réseaux en comparant leurs résultats à la vérité de terrain.
3. Nous concluons que les réseaux artificiels sont pertinents uniquement si les communautés détectées à l'étape 2 sont cohérentes avec la vérité de terrain générée à l'étape 1.

Les algorithmes utilisés dans l'étape 2 sont :

- MetaFac [LSC+09] : cet algorithme détecte des communautés via une factorisation de tenseurs et nécessite de connaître le nombre de communautés a priori[1]. De plus, il ne peut détecter que des communautés contenant au moins un sommet de chaque couche.
- CompMod [LLMW14] : cet algorithme maximise la *modularité composite* qui est une combinaison des modularités des différentes couches.

Nous validons les résultats en comparant les communautés obtenues et les communautés de la vérité de terrain en utilisant l'information mutuelle normalisée (NMI) [DDGDA05]. La validation de l'étape 3 est faite de la manière suivante :

Variation de μ : sur les figures. 6.3(a) et 6.3(b), nous augmentons le paramètre de mélange μ (aussi noté *mu* dans certaines figures) pour dégrader la qualité des communautés. Quand μ augmente, la NMI diminue pour les deux algorithmes, indépendamment de α et p, ce qui signifie que les communautés de terrain se dégradent comme espéré.

[1] Dans nos expériences nous faisons varier ce nombre de 2 à 50 et indiquons les résultats correspondant à la plus grande similarité avec la vérité de terrain.

Variation de p : de même les figures 6.3(c) et 6.3(d) s'intéressent à p qui régule la cohésion des communautés multi-couches. Comme attendu, la NMI augmente avec p pour les deux algorithmes. La pente est plus prononcée pour des valeurs de α plus élevées du fait de l'augmentation du nombre de communautés multi-couches.

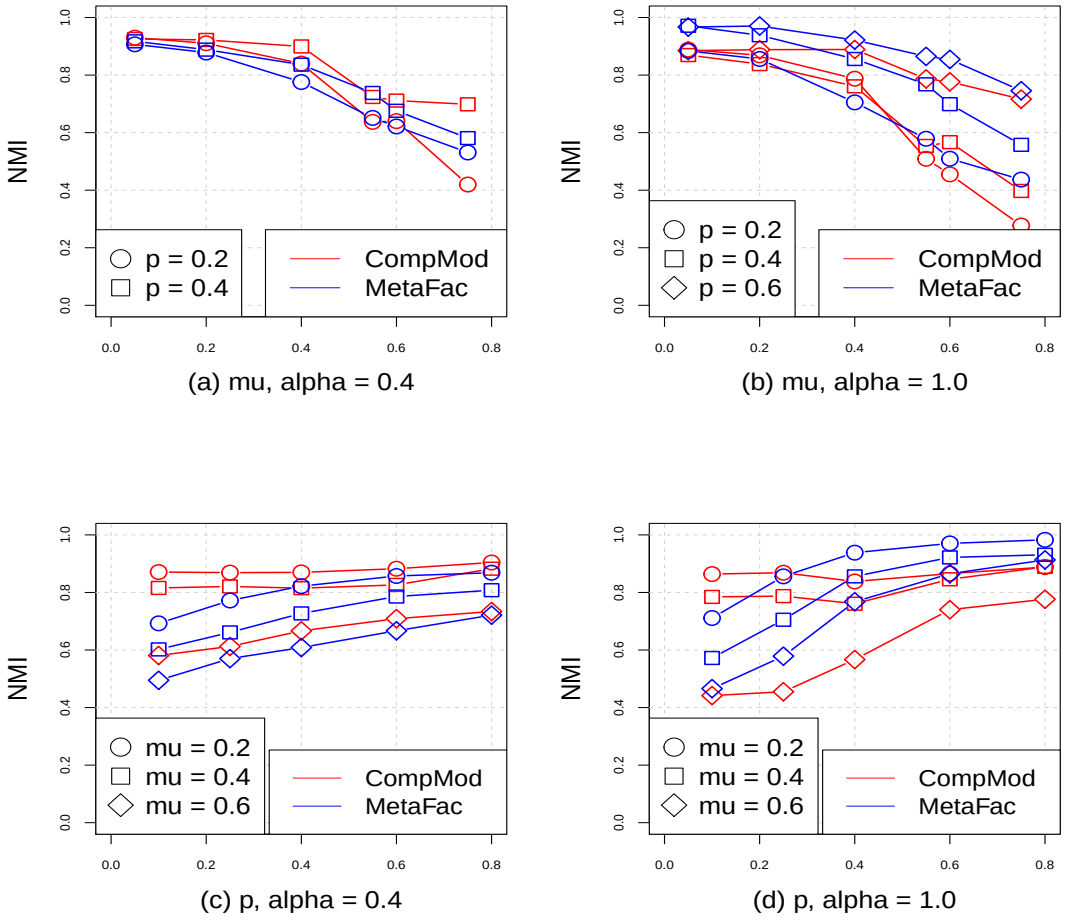


FIGURE 6.3: Variations de NMI en fonction de μ et p pour 'CompMod' et 'MetaFac' sur des réseaux à 2 couches avec 100 sommets sur chaque couche, générés avec un degré maximum $k^i_{max} = 10$, un degré moyen $\langle k_i \rangle = 6$ et une densité des liens de couplage $d = 0.07$.

Dans les deux cas la pertinence des communautés de terrain est en accord avec l'intuition, le modèle semble donc suffisamment pertinent pour les études suivantes.

6.3 Développement d'algorithmes de détection de communautés

Dans cette section nous allons tout d'abord proposer une modularité spécifique aux communautés multi-couches Q_M puis nous montrerons qu'il est possible d'adapter les algorithmes classiques à Q_M pour obtenir des algorithmes efficaces pour la détection de communautés multi-couches.

6.3.1 Propriétés souhaitables

Comme indiqué dans la Section 6.1, nous pouvons observer deux types de communautés dans les réseaux multi-couches (voir Figure. 6.2), (a) des communautés multi-couches et (b) des communautés mono-couche. Les propriétés souhaitables pour une communauté multi-couches $C = (\mathcal{C}_\mathcal{U}, \mathcal{C}_\mathcal{B})$ de $\mathcal{G}$ (où $\mathcal{C}_\mathcal{B} \neq \emptyset$) sont :

Propriété^{X1} : le groupe de sommets dans chaque couche et inter-couche (L_i^Cs et L_{ij}^Cs) devrait être cohésif.

Propriété^{X2} : les liens de couplage dans $L_{ij}^C \in \mathcal{C}_\mathcal{B}$ devraient être nombreux entre L_i^C (i.e. V_i^C) et L_j^C (i.e. V_j^C).

De même, une communauté mono-couche idéale $C = (\mathcal{C}_\mathcal{U}, \mathcal{C}_\mathcal{B})$ de $\mathcal{G}$ (où $\mathcal{C}_\mathcal{B} = \emptyset$) devrait avoir les propriétés suivantes :

Propriété^{S1} : C devrait être cohésive dans la couche L_i à laquelle elle appartient (i.e. $\mathcal{C}_\mathcal{U} = L_i^C \subseteq L_i$).

Propriété^{S2} : les sommets dans L_i^C (i.e. V_i^C) devraient être peu connectés aux autres couches L_j.

6.3.2 Modularité multi-couche

La modularité classique de Newman-Girvan [New06b] $Q = \frac{1}{2m}\sum_{i,j}(A_{ij} - P_{ij})\delta(\psi_i, \psi_j)$ où m est le nombre de liens du réseau, $i, j \in V_1 \cup V_2$ sont les paires de sommets du réseau, ψ_i est la communauté de i et $\delta(\psi_i, \psi_j)$ vaut 1 si et seulement si $\psi_i = \psi_j$. A_{ij} est la matrice d'adjacence de $\mathcal{G}$. Enfin P_{ij} est la probabilité d'existence d'un lien entre i et j si les liens sont placé aléatoirement (généralement appelé modèle nul).

Soit un réseau à deux couches $\mathcal{G} = \{\{L_1, L_2\}, \{L_{12}\}\}$ où $L_1 = (V_1, E_1)$ & $L_2 = (V_2, E_2)$ sont les couches et $L_{12} = (V_1, V_2, E_{12})$ est le graphe biparti connectant ces couches.

Dans un réseau multi-couches les liens d'une couche L_i sont intrinsèquement différents de ceux d'une autre couche L_j, ce qui se reflète dans les propriétés des couches. Cette observation nous incite à dissocier les modèles des différentes couches et de proposer la modularité suivante :

$$Q_M = \frac{1}{2m}\sum_{ij}\{(A_{ij} - P_{ij})\delta(\psi_i, \psi_j)\} \text{ où}$$
$$P_{ij} = \begin{cases} P_{ij}^1 & si\ i \in V_1\ et\ j \in V_1 \\ P_{ij}^2 & si\ i \in V_2\ et\ j \in V_2 \\ P_{ij}^{12} & si\ (i \in V_1\ et\ j \in V_2)\ \vee\ (i \in V_2\ et\ j \in V_1) \end{cases} \tag{6.1}$$

Dans la suite nous calculons les termes P_{ij}^1, P_{ij}^2 et P_{ij}^{12} séparément pour chaque type de communauté puis nous les assemblons pour définir Q_M.

6.3.2.1 Communautés multi-couches

Toute communauté multi-couches C est composée de trois parties, celles correspondant aux deux couches (L_1^C, L_2^C) et celle de l'inter-couche (L_{12}^C). La modularité classique permet de calculer P_{ij}^1, P_{ij}^2 directement : $P_{ij}^1 = (h_i * h_j)/2\,|E_1|$ (pour L_1^C) and $P_{ij}^2 = (h_i * h_j)/2\,|E_2|$ (pour L_2^C) où h_i (respectivement h_j) est le nombre de liens du sommet i (respectivement j) à l'intérieur d'une couche (soit L_1^C ou L_2^C). Pour la partie inter-couche (dite de couplage ou bipartie) L_{12}^C, la probabilité d'existence d'un lien de couplage entre i et j dépend des degrés de couplage c_i et c_j : $P_{ij}^{12} = (c_i * c_j)/\,|E_{12}|$ (comme dans [Bar07]). Ces modèles respectent la Propriété^{X1} introduite dans la Section 6.3.1.

Chaque communauté multi-couches C est représentée par $\{\{L_1^C, L_2^C\}, \{L_{12}^C\}\}$ où L_1^C et L_2^C sont les sous-groupes avec les liens de E_1 et E_2 respectivement et L_{12}^C de E_{12}. En substituant P_{ij} dans Équation 6.1 on obtient :

$$Q_M^C = \left[\frac{1}{3}\left\{\frac{1}{2\,|E_1|}\sum_{i,j\in V_1\cap C}(A_{ij} - \frac{(h_i * h_j)}{2\,|E_1|}) + \right.\right.$$
$$\frac{1}{|E_{12}|}\sum_{i\in V_1\cap C, j\in V_2\cap C}(A_{ij} - \frac{(c_i * c_j)}{|E_{12}|}) + \tag{6.2}$$
$$\left.\left.\frac{1}{2\,|E_2|}\sum_{i,j\in V_2\cap C}(A_{ij} - \frac{(h_i * h_j)}{2\,|E_2|})\right\}\right]$$

Dans le cas des communautés multi-couches, si $i \in C$ n'est connecté avec aucune autre couche (le degré de couplage $c_i = 0$), nous utilisons son degré intra-couche h_i au lieu

de c_i pour calculer P_{ij}^{12}. Cela permet de pénaliser les sommets des communautés multi-couches C qui ne sont connectés qu'à des sommets de la même couche. Cette modification du modèle nul P_{ij}^{12} satisfait la propriété Propriété^{X2}. En conséquence, l'Équation 6.2 peut être écrite comme :

$$Q_M^C = \left[\frac{1}{3}\left\{\frac{1}{2\left|E_1\right|}\sum_{i,j\in V_1\cap C}\left(A_{ij}-\frac{(h_i*h_j)}{2\left|E_1\right|}\right)+\right.\right.$$
$$\frac{1}{2\left|E_1\right|+2\left|E_2\right|+\left|E_{12}\right|}\sum_{i\in V_1\cap C, j\in V_2\cap C}\left(A_{ij}-\frac{(c'_i*c'_j)}{2\left|E_1\right|+2\left|E_2\right|+\left|E_{12}\right|}\right)+ \quad (6.3)$$
$$\left.\left.\frac{1}{2\left|E_2\right|}\sum_{i,j\in V_2\cap C}\left(A_{ij}-\frac{(h_i*h_j)}{2\left|E_2\right|}\right)\right\}\right]$$

où pour chaque sommet i, $c'_i = c_i$ si $c_i > 0$ et $c'_i = h_i$ sinon.

6.3.2.2 Communautés mono-couches

Dans une communauté mono-couche C tous les sommets appartenant soit à L_1 soit à L_2, les modèles nuls valent $P_{ij}^1 = (h_i*h_j)/2\left|E_1\right|$ et $P_{ij}^2 = (h_i*h_j)/2\left|E_2\right|$ d'après [New06b]. Ces modèles satisfont la Propriété^{S1}. Pour chaque communauté mono-couche C, on substitue P_{ij} dans Équation 6.1 pour obtenir :

$$Q_M^C = \left[\frac{1}{3}\left\{\frac{1}{2\left|E_1\right|}\sum_{i,j\in V_1\cap C}\left(A_{ij}-\frac{(h_i*h_j)}{2\left|E_1\right|}\right)\right\}\right] \quad (6.4)$$

Il peut y avoir de nombreux sommets dans C connectés à d'autres couches par des liens de couplage en contradiction avec la Propriété^{S2}. Pour pénaliser cela pour un sommet i dans une communauté C avec un degré de couplage c_i on ajoute c_i et h_i pour estimer le modèle nul ce qui donne :

$$Q_M^C = \left[\frac{1}{3}\left\{\frac{1}{2\left|E_1\right|+\left|E_{12}\right|}\sum_{i,j\in V_1\cap C}\left(A_{ij}-\frac{(h_i+c_i)*(h_j+c_j)}{2\left|E_1\right|+\left|E_{12}\right|}\right)\right\}\right] \quad (6.5)$$

Enfin, en combinant tout, nous obtenons

$$Q_M = \frac{1}{3}\sum_{k=1}^{n_C}\Bigg[\Bigg\{\frac{1}{2|E_1| + \theta_{C_k} * |E_{12}|}\sum_{i,j\in V_1\cap C_k}\Big(A_{ij} - \frac{(h_i + \theta_{C_k} * c_i) * (h_j + \theta_{C_k} * c_j)}{2|E_1| + \theta_{C_k} * |E_{12}|}\Big) + \frac{1}{2|E_1| + 2|E_2| + |E_{12}|}\sum_{i\in V_1\cap C_k, j\in V_2\cap C_k}\Big(A_{ij} - \frac{(c'_i * c'_j)}{2|E_1| + 2|E_2| + |E_{12}|}\Big) + \frac{1}{2|E_2| + \theta_{C_k} * |E_{12}|}\sum_{i,j\in V_2\cap C_k}\Big(A_{ij} - \frac{(h_i + \theta_{C_k} * c_i) * (h_j + \theta_{C_k} * c_j)}{2|E_2| + \theta_{C_k} * |E_{12}|}\Big)\Bigg\}\Bigg] \tag{6.6}$$

où n_C est le nombre total de communautés et θ_{C_k} indique le type de la communauté C_k. θ_{C_k} vaut 1 si C_k est mono-couche et 0 sinon. Pour une communauté mono-couche, au plus un des termes de Équation 6.6 est non-nul.

Les calculs précédents ont été effectués avec deux couches mais ils se généralisent sans problème à plus de couches. Par exemple, si le réseau a L couches et $\frac{L(L-1)}{2}$ relations de couplage, il y aurait autant de termes de modularité mono-couches ou inter-couches.

6.3.3 Algorithmes d'optimisation

Nous nous basons sur deux algorithmes de l'état de l'art pour détecter des communautés sur des réseaux classiques, Girvan-Newman [NG04] et Louvain [BGLL08] qui optimisent tous deux la modularité classique [New06b]. Nous substituons la modularité classique par notre nouvelle modularité Q_M pour obtenir **GN-Q_M** et **Louvain-Q_M** respectivement. Bien que les algorithmes originaux ne soient pas capables de distinguer les différents types de sommets et liens dans le réseau multi-couches, du fait de l'adaptabilité de Q_M, **GN-Q_M** et **Louvain-Q_M** devraient être capables de détecter les différents types de communautés. D'une certaine manière, Q_M fonctionne comme un correctif pouvant transformer n'importe quel algorithme d'optimisation de la modularité au cas multi-couches.

Algorithme 3 : GN-Q_M

Input : Un réseau multi-couches $\mathcal{G}$ tel que défini Section 6.3.2 où $E = E_1 \cup E_2 \cup E_{12}$.
Output : Q_M maximum de $\mathcal{G}$ et communautés.
Calculer la centralité d'intermédiarité de tous les liens
while $|E| > 1$ **do**
  Supprimer le lien $e \in E$ de centralité maximale de E
  $currQ$ = valeur courante Q_M de $\mathcal{G}$
  Sauvegarder la partition courante et $currQ$
  Mettre à jour les centralités des liens restants
Retourner la partition de Q_M maximum
return

Algorithme 4 : Louvain-Q_M

Input : Un réseau multi-couches $\mathcal{G}$ tel que défini Section 6.3.2 où $V = V_1 \cup V_2 \cup V_{12}$.
Output : Q_M maximum de $\mathcal{G}$ et communautés.
while *Vrai* **do**
  Placer chaque sommet de $\mathcal{G}$ dans une communauté
  Sauvegarder Q_M pour sa partition
  while *Au moins un sommet a été déplacé* **do**
    foreach $i \in V$ **do**
      c = communauté voisine de i qui maximise Q_M
      if *c résulte en une amélioration strictement positive* **then**
        Déplacer i de sa communauté à la communauté c
  if *La valeur de Q_M est plus élevée qu'au précédent passage.* **then**
    Retourner la partition de $\mathcal{G}$
    Générer un nouveau réseau $\mathcal{G}$ avec les communautés détectées
  else
    break
return

6.4 Evaluation de Q_M

Dans cette section nous comparons Q_M à d'autres indices en utilisant notre générateur de réseaux artificiels présenté Section 6.2.1 avec différents paramètres.

Dans la littérature, peu d'indices de modularité ont été proposés pour les réseaux multi-couches [LLMW14, STL+15]. En fait, comme 'CompMod' [LLMW14] ne peut évaluer que des communautés mono-couches, la seule approche à laquelle nous pouvons réellement nous comparer est 'mQ' [STL+15]. Pour un réseau bi-couches $\mathcal{G} = \{\{L_1, L_2\}, \{L_{12}\}\}$ où $L_1 = (V_1, E_1)$ et $L_2 = (V_2, E_2)$ sont les deux couches et $L_{12} = (V_1, V_2, E_{12})$ est le réseau biparti connectant L_1 et L_2, mQ vaut

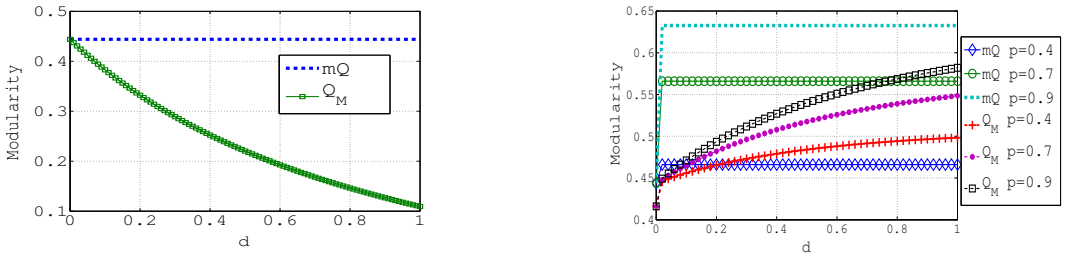
$$mQ = \frac{1}{3}\sum_{k=1}^{n_C}\Bigg\{\underbrace{\left(\frac{\left|E_1^{C_k}\right|}{|E_1|} - \left(\frac{h_1^{C_k}}{2\,|E_1|}\right)^2\right)}_{Terme\ pour\ L_1} + \underbrace{\left(\frac{\left|E_{12}^{C_k}\right|}{|E_{12}|} - \frac{r_{12}^{C_k} * s_{12}^{C_k}}{|E_{12}|^2}\right)}_{Terme\ pour\ les\ liens\ de\ couplage} + \underbrace{\left(\frac{\left|E_2^{C_k}\right|}{|E_2|} - \left(\frac{h_2^{C_k}}{2\,|E_2|}\right)^2\right)}_{Terme\ pour\ L_2}\Bigg\} \tag{6.7}$$

où chaque communauté C_k est représentée par $\{\{L_1^{C_k}, L_2^{C_k}\}, \{L_{12}^{C_k}\}\}$; $L_1^{C_k}$ et $L_2^{C_k}$ sont les sous-groupes avec $E_1^{C_k}$ et $E_2^{C_k}$ liens de E_1 et E_2 respectivement et que $L_{12}^{C_k}$ contient $E_{12}^{C_k}$ liens de E_{12} ; n_C est le nombre de communautés ; $h_i^{C_k}$ est la somme des degrés de tous les sommets $L_i^{C_k}$ de L_i ; $r_{12}^{C_k}$ est la somme des degrés des $L_1^{C_k}$ sommets de L_{12} et $s_{12}^{C_k}$ celle des $L_2^{C_k}$ sommets de L_{12}. Pour résumer cet indice combine la modularité de Newman-Girvan [NG04] et celle de Barber pour les réseaux bipartis [Bar07].

Afin de comparer mQ et Q_M, nous construisons un réseau bi-couches $\{\{L_1, L_2\}, \{L_{12}\}\}$ (voir Figure 6.2) contenant trois groupes cohésifs (des cliques) dans chaque couche. Chacune de ces cliques contient 100 sommets (il y a donc 300 sommets par couche) et chaque couche contient $14,852$ liens internes (les trois cliques et deux liens pour les connecter). Nous considérons deux configurations comme illustré Figure 6.2 - (i) **Config A :** des communautés mono-couches (donc six communautés), (ii) **Config B :** des communautés bi-couches contenant un groupe de chaque couche (donc trois communautés).
Les liens de couplage connectant deux couches sont la clé des réseaux multi-couches et permettent notamment de distinguer ces derniers des réseaux multiplexes. Dans cette évaluation nous nous concentrons donc sur ces derniers et jouons sur leur nombre pour réguler la structure du réseau. Cela se fait grâce aux paramètres d_{ij} et p, ie. la densité entre deux couches et la fraction des liens de couplage à l'intérieur de communautés multi-couches (comme discuté Section 6.2.1).

6.4.1 Résultats expérimentaux

Dans notre expérience nous faisons varier la densité de liens de couplage d de 0 à 1, pour chaque configuration (A et B), avec différentes valeurs de p. Intuitivement ajouter des liens de couplage devrait diluer les communautés mono-couches de A et donc diminuer leur modularité et à l'inverse cela doit renforcer les communautés de B et donc augmenter la modularité.



(a) mQ vs. Q_M pour la Config A en ajoutant des liens de couplage.

(b) mQ vs. Q_M pour la Config B en ajoutant des liens de couplage et en faisant varier p.

FIGURE 6.4

6.4.1.1 Config A

Sur la Figure 6.4(a), la courbe de mQ montre que cette fonction n'est pas sensible à l'augmentation de d. Plus précisément, les liens de couplage ne contribuent aucunement dans l'Équation 6.7 (le terme de couplage s'élimine pour les communautés mono-couches). Au contraire, Q_M pénalise les liens de couplage connectés à des communautés mono-couches (voir les termes $(h_i + \theta_{C_k} * c_i)$ de l'Équation 6.6) d'où la diminution de Q_M

quand d augmente (voir Figure 6.4(a)). Ce résultat s'ajoute à la propriété souhaitée Propriété^S, introduite Section 6.3.1.

6.4.1.2 Config B

Pour cette configuration, mQ n'arrive pas à capturer le comportement souhaité (voir Propriété^X) lors de l'augmentation des liens de couplage, sauf au début quand d passe de 0 à une valeur non nulle. Au contraire elle reste constante lors de l'ajout de liens de couplages sans aucun lien avec p (voir Figure 6.4(b)). Comme observé Équation 6.7, l'ajout de liens de couplage n'affecte que le terme de couplage de mQ mais l'ajout de liens modifie le numérateur et le dénominateur quasiment de la même manière ce qui élimine l'effet sur mQ. Sur toutes les courbes correspondant à Q_M, le comportement souhaité est obtenu grâce à la pénalisation introduite dans le modèle nul de l'Équation 6.3.

Comme attendu, la valeur de la modularité augmente avec p pour mQ et Q_M du fait qu'une plus grande valeur de p augmente la cohésion des communautés multi-couches.

6.5 Évaluation sur les réseaux artificiels

Nous évaluons ici la performance des algorithmes de détection de communautés multi-couches proposés **GN-Q_M** et **Louvain-Q_M** dans un environnement contrôlé. Tout d'abord, nous présentons le protocole expérimental de génération du réseau multi-couches artificiel avec des communautés de terrain et la mesure d'évaluation utilisée. Ensuite, nous décrivons les algorithmes en compétition et, enfin, nous mettons en évidence l'efficacité des algorithmes proposés à différents points de vue.

6.5.1 Protocole expérimental

Nous générons les réseaux artificiels à deux couches ($\mathcal{G} = \{\{L_1, L_2\}, \{L_{12}\}\}$) avec communautés implantées en suivant le modèle proposé dans la Section 6.2.1. Nous fixons le nombre de sommets $|V_i|$ dans chaque couche L_i à 100 avec le degré maximum $k^i_{max} = 10$ et le degré moyen $\langle k_i \rangle = 6$. Les autres paramètres du modèle (μ, α, p & d) sont réglés et ajustés en fonction des besoins. Nous appliquons l'information mutuelle normalisée (NMI) [DDGDA05] pour mesurer la similarité entre les communautés détectées et la vérité de terrain.

Le réseau artificiel contient 30 communautés de terrain mono-couches (et pas de multi-couches) quand $\alpha = 0$ et 15 communautés de terrain multi-couches (sans mono-couche) quand $\alpha = 1$.

6.5.2 Algorithmes en compétition

Nous mettons en compétition les trois classes d'algorithmes suivantes pour évaluer la performance de **GN-Q_M** et **Louvain-Q_M**.

6.5.2.1 Méthode avec mQ

Nous utilisons la modularité multi-couches mQ proposée dans [STL+15] avec les algorithmes classiques Grivan-Newman et Louvain [NG04], [BGLL08]. Nous appelons ces algorithmes **GN-mQ** et **Louvain-mQ** respectivement.

6.5.2.2 Approches à base de fusion

Nous appliquons la méthode de Louvain [BGLL08] pour détecter des communautés sur les couches individuelles puis tentons de fusionner ces communautés pour créer des communautés multi-couches. Pour cela, nous fusionnons une communauté C_T d'une couche T avec la communauté C_B de B à laquelle elle est la plus connectée si la densité de connexion est supérieure à un seuil fixé[2]. Si la densité est inférieure au seuil nous conservons la communauté mono-couche.

6.5.2.3 Algorithmes de l'état de l'art

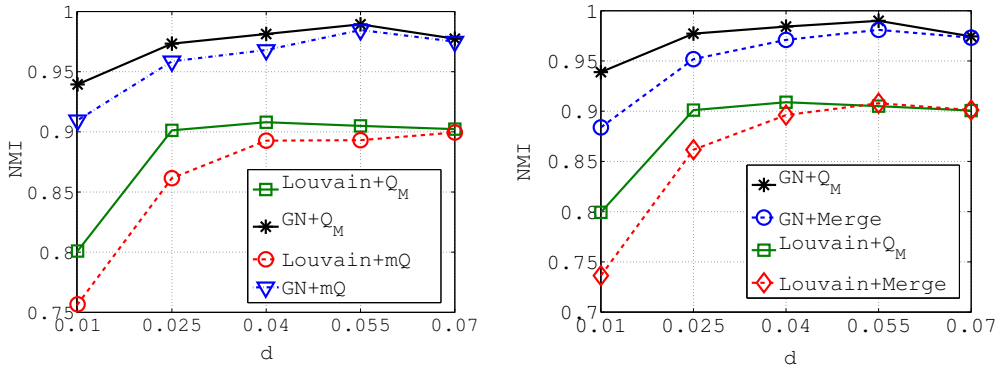
Nous utilisons également 'MetaFac' [LSC+09] et 'CompMod' [LLMW14] déjà présentés dans la Section 6.2.

6.5.3 Évaluation

Les méthodes **GN-Q_M** et **Louvain-Q_M** obtiennent des résultats proches de **GN-mQ** et **Louvain-mQ**. Cependant, la différence se fait lorsque la densité de couplage d diminue. Sur la Figure 6.5(a), nous observons que **GN-Q_M** et **Louvain-Q_M** donnent de meilleurs résultats en cas de densité faible. Cela vient du fait que mQ ne parvient pas à gérer la cohésion des communautés multi-couches si d est faible comme expliqué dans la Figure 6.4(b).

Les méthodes à base de fusion donnent également des bons résultats du fait qu'elles optimisent indépendamment chacune des couches. Cependant, les méthodes proposées

[2]La fusion est effectuée si le rapport entre le nombre de liens de couplage entre C_T et C_B et le nombre total de liens de couplage connectés à C_T ou C_B est au moins Th en faisant varier Th de 0.1 à 1.0 et ne gardant que le meilleur résultat.



(a) Différentes valeurs de d pour $p = 0.4$, $\alpha = 0.6$, $\mu = 0.05$ pour les méthodes avec mQ

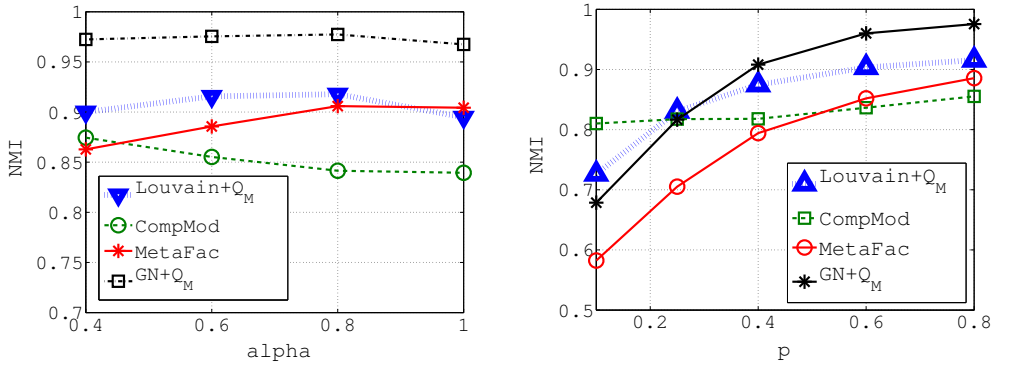
(b) Différentes valeurs de d pour $p = 0.4$, $\alpha = 0.6$, $\mu = 0.05$ pour les approches à base de fusion

FIGURE 6.5: NMI entre les vérités de terrain et les communautés identifiées pour différentes valeurs de d.

GN-Q_M et **Louvain-Q_M** obtiennent de meilleurs résultats si d est faible (voir Figure 6.5(b)).

Concernant les méthodes de l'état de l'art, **(a) Effet de α :** Le paramètre α régule la proportion de communautés multi-couches dans le réseau artificiel. Sur la Figure 6.6(a), nous observons que **GN-Q_M** et **Louvain-Q_M** sont peu sensibles à α et donc les performances de ces deux algorithmes sont globalement indépendantes de la proportion de communautés mono-couches. Au contraire, la qualité de $MetaFac$ augmente avec α et $CompMod$ fait l'inverse : $MetaFac$ a ainsi un biais clair vers la détection de communautés multi-couches et $CompMod$ vers les communautés mono-couches. **(b) Effet de p :** ce paramètre gère la cohésion des liens de couplage dans les communautés multi-couches. Nous observons (Figure 6.6(b)) que nos algorithmes sont meilleurs en cas de communautés multi-couches moyennement à fortement cohésives ($p > 0.3$). Cependant, pour $p < 0.3$, $CompMod$ donne des résultats légèrement meilleurs dû à la dégradation des communautés multi-couches et à son biais sur les communautés mono-couches.

Nous résumer, nous affirmons que (a) les algorithmes proposés offrent un comportement équilibré dans différents scénario pour d, p, etc. et (b) ils peuvent détecter simultanément des communautés mono-couches et multi-couches sans biais apparent (invariance sur α) vers l'une des deux. Bien que **GN-Q_M** et **Louvain-Q_M** soient peu à l'aise si p est faible ce ne sont jamais les plus mauvaise approches. $CompMod$ donne de bons résultats dans ce cas du fait de son biais au contraire de $MetaFac$.



(a) Différent α pour $p = 0.8$, $\mu = 0.4$ et $d = 0.04$ (b) Différent p pour $\mu = 0.4$, $\alpha = 0.6$ et $d = 0.04$

FIGURE 6.6: NMI entre les vérités de terrain et les communautés identifiées pour différentes valeurs de α et p.

6.6 Évaluation sur des réseaux réels

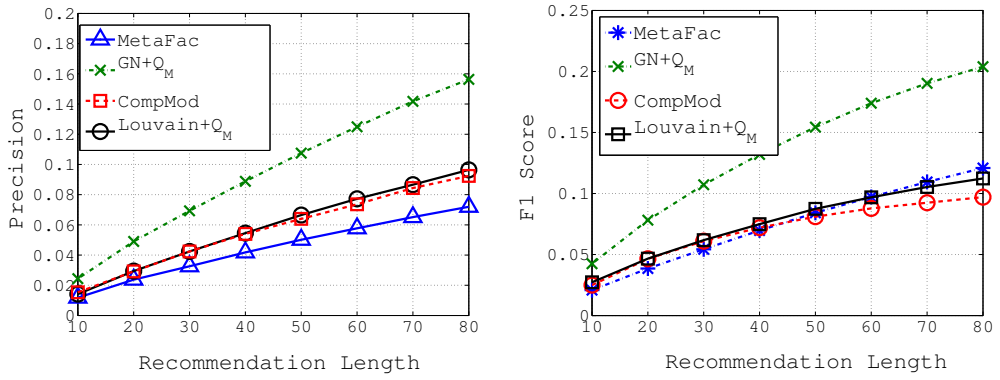
Dans cette section nous évaluons les performances de **GN-Q_M** et **Louvain-Q_M** sur plusieurs réseaux réels. Pour cela nous utilisons deux jeux de données, 'Yelp' et 'Meetup', présentons notre méthodologie de validation et comparons les algorithmes.

6.6.1 Yelp

Les données sont issues de Yelp [GPM15], une plateforme de réseau social basée sur des lieux (LBSN), contenant des informations sur les visiteurs (dont leurs connexions et leur lieu de résidence), les lieux et les conseils et avis postés par les visiteurs. Nous supposons que v a visité L s'il a écrit un conseil ou donné un avis sur L. Nous nous limitons à la ville de "Las Vegas" pour laquelle nous disposons de 13601 lieux et 173697 visiteurs. Pour simplifier les calculs nous nous limitons aux lieux situés à moins de 5km du centre de la ville et aux visiteurs ayant visité ces lieux. Ensuite nous ne gardons que la composante connexe la plus grande du réseau social des visiteurs ce qui nous laisse 244 visiteurs et 1627 lieux visités par au moins un de ces visiteurs.

Le réseau multi-couches associé est $\mathcal{G}_{yelp} = \{\{L_U, L_L\}, \{L_{UL}\}\}$, où $L_U = (V_U, E_U)$ est la couche visiteurs ; $L_L = (V_L, E_L)$ est la couche des lieux connectés par proximité géographique (un lien existe si la distance est inférieure à $200m$) et $L_{UL} = (V_U, V_L, E_{UL})$ connecte les visiteurs aux lieux qu'ils ont visités (voir Figure 6.1) .

Contrairement aux données artificielles, obtenir une vérité de terrain sur des données réelles pose très souvent problème. Nous évaluons donc la performance des différents algorithmes de manière indirecte en utilisant une application à la recommandation. Dans



(a) Précision (b) score-F1

FIGURE 6.7: Précision et score-F1 (moyenné sur tous les visiteurs) des communautés obtenues pour différents algorithmes et différentes recommandations.

les plateformes de réseaux sociaux basés sur des lieux, la recommandation de lieux est un problème classique [GPM15] : pour un visiteur $v \in V_U$, un ensemble de lieux potentiels $L \subseteq V_L$ lui sont recommandés d'après son profil. Nous appliquons un algorithme de filtrage collaboratif basé sur les similarités des lieux [RIS+94] pour obtenir K recommandations de lieux pour chaque visiteur v et nous considérons cela comme la vérité de terrain. Ensuite nous appliquons les algorithmes de détection de communautés multi-couches sur $\mathcal{G}_{yelp}$ afin d'obtenir K' communautés disjointes $C_1, C_2, \ldots, C_{K'}$. Comme indiqué Section 6.1, chaque C_i est $\{\{L_U^{C_i}, L_L^{C_i}\}, \{L_{UL}^{C_i}\}\}$ où $L_U^{C_i} = (V_U^{C_i}, E_U^{C_i})$, $L_L^{C_i} = (V_L^{C_i}, E_L^{C_i})$ et $L_{UL}^{C_i} = (V_U^{C_i}, V_L^{C_i}, E_{UL}^{C_i})$. Nous supposons que pour un visiteur $v \in V_U^{C_i}$, l'ensemble $V_L^{C_i}$ correspond aux lieux à recommander selon l'algorithme de détection de communautés et nous confrontons cet ensemble $V_L^{C_i}$ à la vérité de terrain obtenue par filtrage collaboratif.

Supposons que pour un visiteur v, L_C soit l'ensemble des lieux recommandés via la détection de communautés et L_R ceux recommandés par le filtrage collaboratif. Nous évaluons la qualité en utilisant (a) la précision et (b) le score-F1 avec $\frac{|L_C \cap L_R|}{|L_C|}$ et $\frac{2|L_C \cap L_R|}{|L_C|+|L_R|}$ respectivement (voir Section 3.3.2). Nous n'utilisons pas le rappel ($\frac{|L_C \cap L_R|}{|L_R|}$) pour éviter le biais vers les communautés de grande taille.

La Figure 6.7 présente la précision et le score F1 (moyenné sur tous les visiteurs) en faisant varier le nombre de recommandations K pour les différents algorithmes. En principe l'augmentation de K incrémente le numérateur ($|L_C \cap L_R|$) des deux métriques ce qui améliore les résultats. A la fois **GN-Q_M** et **Louvain-Q_M** (et en particulier **GN-Q_M**) donnent de meilleurs résultats que les autres algorithmes pour pratiquement toutes les valeurs de k.

6.6.2 Meetup

Meetup est un réseau social basé sur les événements (EBSN) qui permet de créer des groupes en ligne ou d'organiser des événements hors ligne. Les données ont été obtenues via un crawler [PGPM16] sur la ville de Chicago pour une période de 20 mois (de août 2015 à mars 2017). Ce jeu de données contient 5727 groupes Meetup, 342773 utilisateurs et 31719 événements gérés par ces groupes. Le jeu de données indique aussi le moment exact où un utilisateur a rejoint un groupe. Nous avons de plus les données sur le profil des groupes et des utilisateurs qui sont caractérisés par un ensemble de tags (20 pour les membres et 56 pour les groupes) tels que 'web design', 'foodie', 'cycling', etc. Afin de pouvoir exploiter ces données nous avons filtré les groupes avec plus de 30 membres. Puis nous n'avons retenu que les groupes ayant plus de 10 membres qui l'ont rejoint avant le 30 novembre 2016 (i.e. durant les premiers 80% de la période considérée) et au moins 4 membres qui l'ont rejoint dans les derniers 4 mois. Au final nous avons 49 groupes et 1194 membres.

Nous construisons un réseau multi-couches $\mathcal{G}_{meetup} = \{\{L_M, L_G\}, \{L_{MG}\}\}$ contenant les groupes et les membres. $L_M = (V_M, E_M)$ est la couche des membres connectés par similarité et $L_G = (V_G, E_G)$ est la couche des groupes également connectés par similarité. Dans les deux cas les similarités sont évaluées à l'aide du coefficient de Jaccard calculé sur le recouvrement de leurs tags. Nous connectons les 33% des paires dans L_M et L_G. $L_{MG} = (V_M, V_G, E_{MG})$ représentent les connexions biparties entre membres et groupes si le membre a rejoint le groupe avant le 30 novembre 2016.

Pour évaluer les algorithmes nous utilisons un algorithme de recommandation de groupes. Ce problème qui consiste à recommander un ensemble de groupes $g_S \subseteq V_G$ à un utilisateur x est bien connu [PK16]. Comme précédemment nous commençons par calculer K communautés disjointes $C_1, C_2, \ldots, C_K$, chaque C_i pouvant être exprimée par $\{\{L_M^{C_i}, L_G^{C_i}\}, \{L_{MG}^{C_i}\}\}$ où $L_M^{C_i} = (V_M^{C_i}, E_M^{C_i})$, $L_G^{C_i} = (V_G^{C_i}, E_G^{C_i})$ and $L_{MG}^{C_i} = (V_M^{C_i}, V_G^{C_i}, E_{MG}^{C_i})$. Pour un groupe $g \in V_G$, nous définissons $B_g \subseteq V_M$ et $A_g \subseteq V_M$ comme les membres ayant rejoint g durant la période d'apprentissage (avant le 30 novembre) et celle de test (après cette date). Pour un groupe $g \in V_G^{C_i}$, $(V_M^{C_i} - B_g)$ est l'ensemble des utilisateurs auquel il faut recommander de joindre le groupe g dans la période de test.

Étant donné M_g l'ensemble des utilisateurs recommandés par l'algorithme de détection de communautés et A_g la vérité de terrain sur les groupes effectivement rejoints nous calculons (a) la précision et (b) le score-F1 de la recommandation (voir Section 3.3.2). La Figure 6.8 montre ces deux valeurs (moyennées sur tous les groupes) pour les algorithmes proposés et l'état de l'art. À nouveau, **GN-Q_M** est le meilleur en termes de précision et **Louvain-Q_M** le meilleur en termes de score-F1.

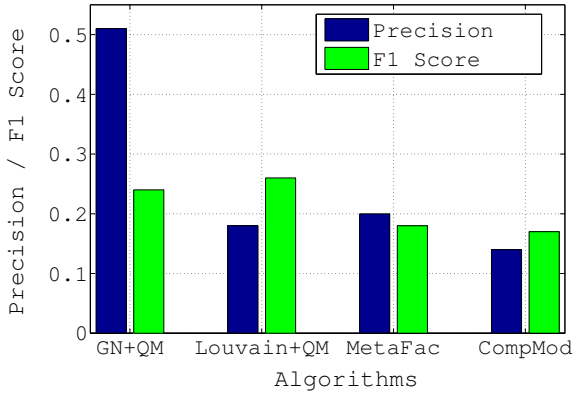


FIGURE 6.8: Précision et score-F1 (moyennés sur tous les groupes) des communautés obtenues sur Meetup N/W.

6.7 Discussion

L'intérêt principal a été de développer Q_M, un nouvel indice de modularité pour évaluer la qualité des communautés dans les réseaux multi-couches. L'expérimentation a montré que cette modularité améliore les définitions de l'état de l'art concernant la modularité multi-couche [LLMW14, STL+15] et réagit comme attendu dans différents scénarios. Nous avons démontré l'utilité de Q_M en développant des algorithmes de détection de communautés multi-couches **GN-Q_M** et **Louvain-Q_M**, en substituant la modularité par Q_M dans les algorithmes classiques.

Afin d'examiner la modularité Q_M et d'évaluer les algorithmes proposés dans un environnement contrôlé, nous avons développé une méthodologie permettant de générer des réseaux multi-couches artificiels incluant des communautés. Les résultats observés de nos algorithmes ont été meilleurs que ceux des techniques de détection de communautés présentées par l'état de l'art sur un large éventail de paramètres caractérisant les réseaux. En particulier, contrairement aux autres algorithmes, leurs performances restent quasi constantes quel que soit le nombre de communautés présentes dans les réseaux. Ainsi, la découverte de communautés par ces algorithmes permet des applications pratiques pour la recommandation de lieux dans Yelp ou de groupes dans Meetup, ce qui montre la pertinence de notre approche sur des jeux de données réels.

Le présent travail a donné naissance à un article accepté dans la conférence IEEE DSAA'17 (*Discovering Community Structure in Multilayer Networks*) et sa version française à la conférence MARAMI'17.

Chapitre 7

Conclusion

Nous avons abordé dans cette thèse la problématique de la caractérisation des communautés et de leur détection dans les réseaux bipartis. Nos études s'intéressent plus largement à tout formalisme faisant intervenir une structure bipartie. Nos contributions se sont orientées dans trois directions.

Tout d'abord, nous nous sommes intéressés à l'analyse du recouvrement dans la structure bipartie des réseaux réels. Nous avons étudié les propriétés liées au recouvrement dans plusieurs jeux de données issus de réseaux sociaux d'Internet en utilisant différentes métriques capturant dans la structure bipartie la densité locale, c'est-à-dire une densité qui est fonction du voisinage du sommet considéré. Pour cela, nous nous sommes appuyés sur deux métriques classiques et avons proposé deux nouvelles métriques pour affiner l'analyse. Nous avons montré que le coefficient de redondance est plus adapté que le coefficient de clustering pour capturer les recouvrements dans les réseaux étudiés. De plus, l'utilisation du coefficient de dispersion et de monopole ont permis de dévoiler que les nœuds de fort degré sont souvent impliqués dans les recouvrements détectés par le coefficient de redondance.

Notre deuxième contribution consiste en un algorithme de détection de communautés appelé ComSim qui est compatible avec tout graphe présentant une structure bipartie. La méthode développée repose sur les relations de voisinage entre deux ensembles de sommets bipartis. Un des deux ensembles est projeté afin de générer un graphe dont les arêtes sont pondérées par la force des similarités calculées préalablement entre les sommets du graphe biparti initial. Dans ce nouveau graphe pondéré, l'algorithme recherche des cycles en se déplaçant de voisin en voisin et en sélectionnant les voisins avec le poids maximal. Dès qu'un cycle est détecté de cette manière, il est considéré comme une communauté qui pourra alors, lors d'une deuxième phase, être agrémentée de sommets

n'intervenant dans aucun cycle. Nous avons évalué la qualité des communautés détectées par cette méthode sur des réseaux bipartis empiriques dont nous connaissons la vérité de terrain et sur lesquels nous avons confronté les algorithmes de l'état de l'art. Nous avons montré que ComSim est l'algorithme le plus efficace pour retrouver les communautés de la vérité de terrain. Pour démontrer la pertinence de notre méthode, nous l'avons également appliqué à un grand réseau provenant d'Internet afin d'évaluer les propriétés structurelles des communautés détectées, notamment à travers des mesures de connectivé interne et d'homogénéité des attributs. Par ailleurs, notre méthode est capable de détecter des communautés dont la taille est relativement petite, sans corrélation notable avec la taille du réseau.

Enfin, nous nous sommes intéressés à la détection de communautés dans les réseaux multi-couches, parfois appelés réseaux de réseaux. Chaque couche de ce type de réseau peut être représentée par un graphe uniparti et les liens entre deux couches comme un graphe biparti. Dans ce contexte, nous avons proposé une généralisation de la modularité classique qui est capable de détecter à la fois des communautés mono-couches, composées de sommets provenant d'une seule couche, et des communautés dites multi-couches, composées de sommets provenant de plusieurs couches. Nous avons également proposé deux algorithmes utilisant notre modularité comme fonction de qualité. Nous avons montré sur des réseaux artificiels ainsi que sur des réseaux réels que cette méthode est pertinente et permet de révéler des structures complexes liées à ce contexte.

Nos travaux ouvrent plusieurs perspectives. Tout d'abord, il reste des nombreuses pistes à explorer dans l'étude du recouvrement. Nous avons montré que les recouvrements sont capturables dans le voisinage des nœuds à travers la redondance et la dispersion des liens. Cependant, l'analyse pourrait être approfondie en étudiant d'autres aspects. En premier lieu, développer des métriques sollicitant de nouvelles propriétés structurelles permettrait d'affiner l'analyse et de mieux décrire les caractéristiques liées au recouvrement. De plus, il serait judicieux d'évaluer le comportement des métriques dans un environnement contrôlé, par exemple à l'aide de réseaux artificiels présentant plusieurs degrés de recouvrements (en faisant varier les degrés de recouvrement par exemple). En deuxième lieu, il serait intéressant d'analyser les recouvrements d'un algorithme de détection de communautés recouvrantes. Dans cette vision, le résultat de la détection est vue comme un graphe biparti, dans lequel les sommets $\top$ représentent les communautés détectées, les sommets $\bot$ représentent les sommets d'origine, et les liens reliant les sommets $\bot$ contenus dans les communautés de $\top$. Ceci permettrait par exemple d'évaluer les communautés recouvrantes détectées et de montrer leur pertinence face à ce type de détection.

La définition informelle d'une communauté donnée par "un groupe de nœuds fortement liés à l'intérieur et peu liés à l'extérieur" se base sur un critère topologique qui s'apparente à la densité interne de ce groupe. Ceci a du sens lorsqu'on s'intéresse à partitionner le réseau en communautés mais a moins de sens lorsqu'on cherche à trouver des communautés recouvrantes ou plus généralement lorsqu'on cherche à rassembler les nœuds selon un certain profil de connexion. Par conséquent, nous souhaitons décrire une communauté comme étant un ensemble de nœuds qui forment un groupe cohérent selon un ou plusieurs critères. En effet, il peut y avoir d'autres propriétés topologiques, autres que la densité, qui pourraient permettre de détecter de tels groupes. Une approche favorable consisterait à étudier les propriétés topologiques des communautés de terrain afin de développer un algorithme adapté à ce qui a été observé. L'enjeu est alors de déterminer ou "détecter" les propriétés topologiques caractérisant les communautés de terrain observées. L'utilisation de techniques d'apprentissage automatique est envisageable, grâce aux réseaux de neurones artificiels par exemple. De plus, cette approche permettrait d'identifier plusieurs types de communautés selon le choix du "critère détecte" ou "critère appris". Ainsi, ce dernier pourrait servir de fonction qualité dans un algorithme d'optimisation.

Afin de poursuivre conjointement l'étude du recouvrement et de la détection de communautés, plusieurs pistes restent ouvertes. Tout d'abord, nous pensons que les métriques caractérisant les recouvrements, comme la redondance par exemple, pourraient détecter des communautés présentant un fort recouvrement. Dans cette optique, il serait possible de proposer une version bipartie de Louvain qui pourrait intégrer, comme fonction de qualité, le coefficient de redondance, le coefficient de dispersion ou tout autre métrique capturant les recouvrements. Idéalement, cette fonction de qualité devrait être capable de mesurer les recouvrements, dans le sous-graphe induit par une communauté, avec une sensibilité telle qu'un déplacement d'un sommet d'une communauté à une autre soit détecté finement. Parallèlement, ces métriques pourraient être adaptées sous la forme de mesures de similarité diadique et ainsi être compatibles avec la méthode ComSim présentée Chapitre 5. Par ailleurs, cette dernière pourrait être adaptée à l'identification de communautés recouvrantes. De plus, il serait possible d'étendre ComSim pour une détection de communautés sur plusieurs ensembles de nœuds.

Bibliographie

[ABFX08] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(Sep) :1981–2014, 2008.

[ABHD13] Tim Althoff, Damian Borth, Jörn Hees, and Andreas Dengel. Analysis and forecasting of trending topics in online media streams. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 907–916. ACM, 2013.

[AP15] Alessia Amelio and Clara Pizzuti. Is normalized mutual information a fair measure for comparing community detection methods ? In *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*, pages 1584–1585. IEEE, 2015.

[ARS02] James Abello, Mauricio Resende, and Sandra Sudarsky. Massive quasi-clique detection. *LATIN 2002 : Theoretical Informatics*, pages 598–612, 2002.

[BA99] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439) :509–512, 1999.

[Bar07] Michael J Barber. Modularity and community detection in bipartite networks. *Physical Review E*, 76(6) :066102, 2007.

[BDG+06] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Görke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. Maximizing modularity is hard. *arXiv preprint physics/0608255*, 2006.

[BGLL08] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics : theory and experiment*, 2008(10) :P10008, 2008.

[Bol98] Béla Bollobás. Random graphs. In *Modern Graph Theory*, pages 215–252. Springer, 1998.

[BPC13] Michele Berlingerio, Fabio Pinelli, and Francesco Calabrese. Abacus : frequent pattern mining-based community discovery in multidimensional networks. *DMKD*, 27(3) :294–320, 2013.

[BPP+10] Sergey V Buldyrev, Roni Parshani, Gerald Paul, H Eugene Stanley, and Shlomo Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291) :1025–1028, 2010.

[BPW+13] Danielle S Bassett, Mason A Porter, Nicholas F Wymbs, Scott T Grafton, Jean M Carlson, and Peter J Mucha. Robust detection of dynamic community structure in networks. *Chaos : An Interdisciplinary Journal of Nonlinear Science*, 23(1) :013142, 2013.

[Bur76] Ronald S Burt. Positions in networks. *Social forces*, 55(1) :93–122, 1976.

[CCG14] Romain Campigotto, Patricia Conde Céspedes, and Jean-Loup Guillaume. A generalized and adaptive method for community detection. *arXiv preprint arXiv :1406.2518*, 2014.

[CDGS04] Hyuk Cho, Inderjit S Dhillon, Yuqiang Guan, and Suvrit Sra. Minimum sum-squared residue co-clustering of gene expression data. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pages 114–125. SIAM, 2004.

[CG12] Gennaro Cordasco and Luisa Gargano. Label propagation algorithm : a semi-synchronous approach. *International Journal of Social Network Mining*, 1(1) :3–26, 2012.

[CL02] Fan Chung and Linyuan Lu. Connected components in random graphs with given expected degree sequences. *Annals of combinatorics*, 6(2) :125–145, 2002.

[CLN13] Marek Ciglan, Michal Laclavík, and Kjetil Nørvåg. On community detection in real-world networks and the importance of degree assortativity. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1007–1015. ACM, 2013.

[CNM04] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6) :066111, 2004.

[CST+00] Andrea Califano, Gustavo Stolovitzky, Yuhai Tu, et al. Analysis of gene expression microarrays for phenotype classification. In *Ismb*, volume 8, pages 75–85, 2000.

[CT13] Émilie Coupechoux and Fabien Tarissan. Un modèle pour les graphes bipartis aléatoires avec redondance. In *4ème Journées Modèles et l'Analyse des Réseaux : Approches Mathématiques et Informatique (MARAMI'13)*, 2013.

[CZG+13] Wei Cheng, Xiang Zhang, Zhishan Guo, Yubao Wu, Patrick F. Sullivan, and Wei Wang. Flexible and robust co-regularized multi-domain graph clustering. In *Proceedings of KDD '13*, pages 320–328, NY, USA, 2013. ACM.

[DDGDA05] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics : Theory and Experiment*, 2005(09) :P09008, 2005.

[DFVN14] Xiaowen Dong, Pascal Frossard, Pierre Vandergheynst, and Nikolai Nefedov. Clustering on multilayer graphs via subspace analysis on grassmann manifolds. *IEEE Transactions on Signal Processing*, 62(4) :905–918, 2014.

[DGG41] Allison Davis, Burleigh B. Gardner, and Mary R. Gardner. *Deep South ; a Social Anthropological Study of Caste and Class*. The University of Chicago Press, Chicago, 1941.

[Dhi01] Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM, 2001.

[DMM03] Inderjit S Dhillon, Subramanyam Mallela, and Dharmendra S Modha. Information-theoretic co-clustering. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98. ACM, 2003.

[DWWW08] Nan Du, Bai Wang, Bin Wu, and Yi Wang. Overlapping community detection in bipartite networks. In *Proceedings of the 2008 IEEE/-WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 176–179. IEEE Computer Society, 2008.

[DZLH06] Chris Ding, Ya Zhang, Tao Li, and Stephen R Holbrook. Biclustering protein complex interactions with a biclique finding algorithm. In *Data*

Mining, 2006. ICDM'06. Sixth International Conference on, pages 178–187. IEEE, 2006.

[EB13] Martin G Everett and Stephen P Borgatti. The dual-projection approach for two-mode networks. *Social Networks*, 35(2) :204–210, 2013.

[EC07] Kevin Emamy and Richard Cameron. Citeulike : a researcher's social bookmarking service. *Ariadne*, (51), 2007.

[FB07] Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1) :36–41, 2007.

[FCMR08] Maurizio Filippone, Francesco Camastra, Francesco Masulli, and Stefano Rovetta. A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1) :176–190, 2008.

[For10] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3) :75–174, 2010.

[Fre03] Linton C Freeman. *Finding social groups : A meta-analysis of the southern women data*. na, 2003.

[GdMC10] Benjamin H Good, Yves-Alexandre de Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4) :046106, 2010.

[GL04] Jean-Loup Guillaume and Matthieu Latapy. Bipartite structure of all complex networks. *Information processing letters*, 90(5) :215–221, 2004.

[GL06] Jean-Loup Guillaume and Matthieu Latapy. Bipartite graphs as models of complex networks. *Physica A : Statistical Mechanics and its Applications*, 371(2) :795–813, 2006.

[GLD00] Gad Getz, Erel Levine, and Eytan Domany. Coupled two-way clustering analysis of gene microarray data. *Proceedings of the National Academy of Sciences*, 97(22) :12079–12084, 2000.

[GN02] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12) :7821–7826, 2002.

[GPM15] Saurabh Gupta, Sayan Pathak, and Bivas Mitra. Complementary usage of tips and reviews for location recommendation in yelp. In *PAKDD*, pages 720–731. Springer, 2015.

[Gre97] Elna C Green. *Southern strategies : Southern women and the woman suffrage question*. Univ of North Carolina Press, 1997.

[Gre10] Steve Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10) :103018, 2010.

[GSPA07] Roger Guimerà, Marta Sales-Pardo, and Luís A Nunes Amaral. Module identification in bipartite and directed networks. *Physical Review E*, 76(3) :036102, 2007.

[HA85] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1) :193–218, 1985.

[HDC+01] Ingrid Hedenfalk, David Duggan, Yidong Chen, Michael Radmacher, Michael Bittner, Richard Simon, Paul Meltzer, Barry Gusterson, Manel Esteller, Mark Raffeld, et al. Gene-expression profiles in hereditary breast cancer. *New England Journal of Medicine*, 344(8) :539–548, 2001.

[HLL83] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels : First steps. *Social networks*, 5(2) :109–137, 1983.

[Huf52] David A Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9) :1098–1101, 1952.

[iCS01] Ramon Ferrer i Cancho and Richard V Solé. The small world of human language. *Proceedings of the Royal Society of London B : Biological Sciences*, 268(1482) :2261–2265, 2001.

[Joh67] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3) :241–254, 1967.

[JTA+00] Hawoong Jeong, Bálint Tombor, Réka Albert, Zoltan N Oltvai, and A-L Barabási. The large-scale organization of metabolic networks. *arXiv preprint cond-mat/0010278*, 2000.

[Kan13] Rushed Kanawati. Détection de communautés dans les grands graphes d'interactions (multiplexes) : état de l'art. 2013.

[KGMI+12] Stephen Kelley, Mark Goldberg, Malik Magdon-Ismail, Konstantin Mertsalov, and Al Wallace. Defining and discovering communities in social networks. In *Handbook of Optimization in Complex Networks*, pages 139–168. Springer, 2012.

[KKKS08] Jussi M Kumpula, Mikko Kivelä, Kimmo Kaski, and Jari Saramäki. Sequential algorithm for fast clique percolation. *Physical Review E*, 78(2) :026109, 2008.

[KM15] Zhana Kuncheva and Giovanni Montana. Community detection in multiplex networks using locally adaptive random walks. In *Proceedings of ASONAM'15*. ACM, 2015.

[KN11] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83(1) :016107, 2011.

[Lan95] Ken Lang. Newsweeder : Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339, 1995.

[LAV10] Anna Lázár, Dániel Abel, and Tamás Vicsek. Modularity measure of networks with overlapping communities. *EPL (Europhysics Letters)*, 90(1) :18001, 2010.

[LBA11] Pierre Latouche, Etienne Birmelé, and Christophe Ambroise. Overlapping stochastic block models with application to the french political blogosphere. *The Annals of Applied Statistics*, pages 309–336, 2011.

[LC13] Conrad Lee and Pádraig Cunningham. Benchmarking community detection methods on social media data. *arXiv preprint arXiv :1302.0739*, 2013.

[LCJ14] Daniel B Larremore, Aaron Clauset, and Abigail Z Jacobs. Efficiently inferring community structure in bipartite networks. *Physical Review E*, 90(1) :012805, 2014.

[LFK09] Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3) :033015, 2009.

[LFR08] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical review E*, 78(4) :046110, 2008.

[LGH05] Pedro G Lind, Marta C Gonzalez, and Hans J Herrmann. Cycles and clustering in bipartite networks. *Physical review E*, 72(5) :056127, 2005.

[LHP+16] Jian-Guo Liu, Lei Hou, Xue Pan, Qiang Guo, and Tao Zhou. Stability of similarity measurements for bipartite networks. *Scientific reports*, 6 :18653, 2016.

[LLMW14] Xin Liu, Weichu Liu, Tsuyoshi Murata, and Ken Wakita. A framework for community detection in heterogeneous multi-relational networks. *Advances in Complex Systems*, 17(06) :1450018, 2014.

[LMDV08] Matthieu Latapy, Clémence Magnien, and Nathalie Del Vecchio. Basic notions for the analysis of large two-mode networks. *Social networks*, 30(1) :31–48, 2008.

[LS13] Jianwu Li and Yulong Song. Community detection in complex networks using extended compact genetic algorithm. *Soft Computing*, 17(6) :925–937, 2013.

[LSC+09] Yu-Ru Lin, Jimeng Sun, Paul Castro, Ravi Konuru, Hari Sundaram, and Aisling Kelliher. Metafac : community discovery via relational hypergraph factorization. In *Proceedings of SIGKDD'09*, pages 527–536. ACM, 2009.

[LSH08] Sune Lehmann, Martin Schwartz, and Lars Kai Hansen. Biclique communities. *Physical Review E*, 78(1) :016108, 2008.

[LWZZ16] Zhenping Li, Rui-Sheng Wang, Shihua Zhang, and Xiang-Sun Zhang. Quantitative function and algorithm for community detection in bipartite networks. *Information Sciences*, 367 :874–889, 2016.

[LZ11] Linyuan Lü and Tao Zhou. Link prediction in complex networks : A survey. *Physica A : statistical mechanics and its applications*, 390(6) :1150–1170, 2011.

[MA84] Leslie C Morey and Alan Agresti. The measurement of classification agreement : An adjustment to the rand statistic for chance agreement. *Educational and Psychological Measurement*, 44(1) :33–37, 1984.

[MDP+11] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[Mei07] Marina Meilă. Comparing clusterings—an information based distance. *Journal of multivariate analysis*, 98(5) :873–895, 2007.

[Mel14] David Melamed. Community structures in bipartite networks : A dual-projection approach. *PloS one*, 9(5) :e97823, 2014.

[MGH11] Aaron F McDaid, Derek Greene, and Neil Hurley. Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint arXiv :1110.2515*, 2011.

[MO04] Sara C Madeira and Arlindo L Oliveira. Biclustering algorithms for biological data analysis : a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(1) :24–45, 2004.

[MRM+10] Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980) :876–878, 2010.

[Mur09] Tsuyoshi Murata. Detecting communities from bipartite networks based on bipartite modularities. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 4, pages 50–57. IEEE, 2009.

[New01] Mark EJ Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2) :404–409, 2001.

[New03a] Mark EJ Newman. Mixing patterns in networks. *Physical Review E*, 67(2) :026126, 2003.

[New03b] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2) :167–256, 2003.

[New06a] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3) :036104, 2006.

[New06b] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23) :8577–8582, 2006.

[New09] Mark EJ Newman. Random graphs with clustering. *Physical review letters*, 103(5) :058701, 2009.

[NG04] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2) :026113, 2004.

[NMMC08] Vincenzo Nicosia, Giuseppe Mangioni, Michele Malgeri, and Vincenza Carchiolo. Extending modularity definition for directed graphs with overlapping communities. Technical report, 2008.

[NP03] Mark EJ Newman and Juyong Park. Why social networks are different from other types of networks. *Physical Review E*, 68(3) :036122, 2003.

[NS01] Krzysztof Nowicki and Tom A B Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455) :1077–1087, 2001.

[NSW01] Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical review E*, 64(2) :026118, 2001.

[NWS02] Mark EJ Newman, Duncan J Watts, and Steven H Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99(suppl 1) :2566–2572, 2002.

[Ops13] Tore Opsahl. Triadic closure in two-mode networks : Redefining the global and local clustering coefficients. *Social Networks*, 35(2) :159–167, 2013.

[PC16] Subhadeep Paul and Yuguo Chen. Null models and modularity based community detection in multi-layer networks. *arXiv preprint arXiv :1608.00623*, 2016.

[PDFV05] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *arXiv preprint physics/0506133*, 2005.

[PGPM16] Soumajit Pramanik, Midhun Gundapuneni, Sayan Pathak, and Bivas Mitra. Predicting group success in meetup. In *ICWSM*, pages 663–666, 2016.

[PK16] Sanjay Purushotham and C.-C. Jay Kuo. Personalized group recommender systems for location- and event-based social networks. *ACM Trans. Spatial Algorithms Syst.*, 2(4) :16 :1–16 :29, November 2016.

[PKVS12] Symeon Papadopoulos, Yiannis Kompatsiaris, Athena Vakali, and Ploutarchos Spyridonos. Community detection in social media. *Data Mining and Knowledge Discovery*, 24(3) :515–554, 2012.

[PL05] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *ISCIS*, volume 3733, pages 284–293, 2005.

[PMB10] Yongjin Park, Cristopher Moore, and Joel S Bader. Dynamic networks from hierarchical bayesian graph clustering. *PloS one*, 5(1) :e8118, 2010.

[PPDSLP14] Arnau Prat-Pérez, David Dominguez-Sal, and Josep-Lluis Larriba-Pey. High quality, scalable and parallel community detection for large real graphs. In *Proceedings of the 23rd international conference on World wide web*, pages 225–236. ACM, 2014.

[PPZ09] Mahendra Piraveenan, Mikhail Prokopenko, and Albert Y Zomaya. Local assortativity and growth of internet. *The European Physical Journal B-Condensed Matter and Complex Systems*, 70(2) :275–285, 2009.

[RA04] Garry Robins and Malcolm Alexander. Small worlds among interlocking directors : Network structure and distance in bipartite graphs. *Computational & Mathematical Organization Theory*, 10(1) :69–94, 2004.

[RAB09] Martin Rosvall, Daniel Axelsson, and Carl T Bergstrom. The map equation. *The European Physical Journal-Special Topics*, 178(1) :13–23, 2009.

[RAK07] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3) :036106, 2007.

[Ran71] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336) :846–850, 1971.

[RB06] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(1) :016110, 2006.

[RCC+04] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9) :2658–2663, 2004.

[RCY+11] Karl Rohe, Sourav Chatterjee, Bin Yu, et al. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4) :1878–1915, 2011.

[Rho93] Stephen A Rhoades. The herfindahl-hirschman index. *Fed. Res. Bull.*, 79 :188, 1993.

[RIS+94] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens : an open architecture for collaborative filtering of netnews. In *Proceedings of CSCW'94*, pages 175–186. ACM, 1994.

[RSM+02] Erzsébet Ravasz, Anna Lisa Somera, Dale A Mongru, Zoltán N Oltvai, and A-L Barabási. Hierarchical organization of modularity in metabolic networks. *science*, 297(5586) :1551–1555, 2002.

[SABA14] Fatiha Souam, Ali Aïtelhadj, and Riadh Baba-Ali. Dual modularity optimization for detecting overlapping communities in bipartite networks. *Knowledge and information systems*, 40(2) :455–488, 2014.

[ŠB11] Lovro Šubelj and Marko Bajec. Robust network community detection using balanced propagation. *The European Physical Journal B-Condensed Matter and Complex Systems*, 81(3) :353–362, 2011.

[SCCH09] Huawei Shen, Xueqi Cheng, Kai Cai, and Mao-Bin Hu. Detect overlapping and hierarchical community structure in networks. *Physica A : Statistical Mechanics and its Applications*, 388(8) :1706–1712, 2009.

[SG02] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research*, 3(Dec) :583–617, 2002.

[SPGMA07] Marta Sales-Pardo, Roger Guimera, André A Moreira, and Luís A Nunes Amaral. Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences*, 104(39) :15224–15229, 2007.

[STH+14] Yizhou Sun, Jie Tang, Jiawei Han, Cheng Chen, and Manish Gupta. Co-evolution of multi-typed objects in dynamic star networks. *IEEE Transactions on Knowledge and Data Engineering*, 26(12) :2942–2955, 2014.

[STL+15] Jianglong Song, Shihuan Tang, Xi Liu, Yibo Gao, Hongjun Yang, and Peng Lu. A modularity-based method reveals mixed modules from chemical-gene heterogeneous network. *PLoS ONE*, 10(4) :1–16, 04 2015.

[SWB51] Claude E Shannon, Warren Weaver, and Arthur W Burks. The mathematical theory of communication. 1951.

[TL10] Lei Tang and Huan Liu. Community detection and mining in social media. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 2(1) :1–137, 2010.

[TM67] Jeffrey Travers and Stanley Milgram. The small world problem. *Phychology Today*, 1 :61–67, 1967.

[TNG14] Fabien Tarissan and Raphaëlle Nollez-Goldbach. The network of the international criminal court decisions as a complex system. In *Iscs 2013 : Interdisciplinary symposium on complex systems*, pages 255–264. Springer, 2014.

[TQM+13] Fabien Tarissan, Bruno Quoitin, Pascal Mérindol, Benoit Donnet, Jean-Jacques Pansiot, and Matthieu Latapy. Towards a bipartite graph modeling of the internet topology. *Computer Networks*, 57(11) :2331–2347, 2013.

[TSS02] Amos Tanay, Roded Sharan, and Ron Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(suppl_1) :S136–S144, 2002.

[VEB09] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison : is a correction for chance necessary ? In *Proceedings of the 26th annual international conference on machine learning*, pages 1073–1080. ACM, 2009.

[WH04] Fang Wu and Bernardo A Huberman. Finding communities in linear time : a physics approach. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2) :331–338, 2004.

[WS98] Duncan J Watts and Steven H Strogatz. Collective dynamics of'small-world'networks. *nature*, 393(6684) :440, 1998.

[WS05] Scott White and Padhraic Smyth. A spectral clustering approach to finding communities in graphs. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 274–285. SIAM, 2005.

[XCL+15] Yongcheng Xu, Ling Chen, Bin Li, et al. Density-based modularity for evaluating community structure in bipartite networks. *Information Sciences*, 317 :278–294, 2015.

[XSL11] Jierui Xie, Boleslaw K Szymanski, and Xiaoming Liu. Slpa : Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*, pages 344–349. IEEE, 2011.

[YL13] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale : a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM, 2013.

[YL15] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1) :181–213, 2015.

[Yul25] G Udny Yule. A mathematical theory of evolution, based on the conclusions of dr. jc willis, frs. *Philosophical transactions of the Royal Society of London. Series B, containing papers of a biological character*, 213 :21–87, 1925.

[YWWY03] Jiong Yang, Haixun Wang, Wei Wang, and Philip Yu. Enhanced biclustering on expression data. In *Bioinformatics and Bioengineering, 2003. Proceedings. Third IEEE Symposium on*, pages 321–327. IEEE, 2003.

[ZHD+01] Hongyuan Zha, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu. Bipartite graph partitioning and data clustering. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 25–32. ACM, 2001.

[ZRMZ07] Tao Zhou, Jie Ren, Matúš Medo, and Yi-Cheng Zhang. Bipartite network projection and personal recommendation. *Physical Review E*, 76(4) :046115, 2007.

[ZWL+08] Peng Zhang, Jinliang Wang, Xiaojia Li, Menghui Li, Zengru Di, and Ying Fan. Clustering coefficient and community structure of bipartite networks. *Physica A : Statistical Mechanics and its Applications*, 387(27) :6869–6875, 2008.

[ZZGZ11] Weihua Zhan, Zhongzhi Zhang, Jihong Guan, and Shuigeng Zhou. Evolutionary method for finding communities in bipartite networks. *Physical Review E*, 83(6) :066120, 2011.