UNIVERSITÉ DU QUÉBEC À MONTRÉAL

UNE NOUVELLE APPROCHE DE DÉTECTION DE COMMUNAUTÉS DANS LES GRAPHES BIPARTIS

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

KHALED NOURI

MARS 2019

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.07-2011). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

REMERCIEMENTS

Je voudrais remercier tout d'abord, Dieu, pour être mon meilleur confident et pour me permettre de réaliser ce mémoire de maîtrise.

Le travail présenté dans ce mémoire a été réalisé au laboratoire LATECE, département d'informatique, faculté des sciences de l'Université du Québec À Montréal (UQAM) sous la supervision de Monsieur Mohamed Bouguessa. De ce fait, je voudrais présenter mes remerciements à mon directeur de recherche. J'ai vraiment profité avec lui pendant longtemps du savoir et du savoir-faire dont j'ai pu bénéficier au cours de nombreuses discussions. Je voudrais également lui témoigner ma gratitude pour sa patience, sa disponibilité et son soutien qui m'a été précieux afin de mener mon travail à bon port.

Je remercie également la faculté des sciences de l'UQAM ainsi l'Ambassade de l'Algérie à Ottawa de m'avoir attribuer la bourse d'exemption des frais de scolarité majorés au cours de mon cursus universitaire.

De même, je remercie mes parents, mes frères, mes sœurs, ainsi que toute ma famille et amis en Algérie et au Canada. En particulier, je remercie mon cher frère Fayçal, ma chère mère Djamila de m'avoir soutenir moralement, financièrement ou de toute autre manière.

Afin de n'oublier personne, mes vifs remerciements s'adressent à tous ceux qui m'ont aidée à la réalisation de ce modeste mémoire.

À la mémoire de mon père, que Dieu bénisse son âme.

TABLE DES MATIÈRES

LISTE DES TABLEAUX iv

LISTE DES FIGURES v

RÉSUMÉ vii

CHAPITRE I
INTRODUCTION 1

1.1 Mise en contexte 1

1.2 Motivations 8

1.3 Contributions 9

1.4 Organisation du mémoire 10

CHAPITRE II
REVUE DE LA LITTÉRATURE 12

2.1 Les approches basées sur la projection 12

2.1.1 La projection pondérée 13

2.1.2 La projection non pondérée 15

2.2 Les approches fondées sur les graphes homogènes 16

2.3 Les approches fondées sur le modèle de liaison similaire 27

2.4 Conclusion 34

CHAPITRE III
APPROCHE PROPOSÉE 36

3.1 Notations et concepts 37

3.2 La première phase : le partitionnement initial 38

3.3 La deuxième phase : raffinement du clustering 45

3.4 Conclusion 49

CHAPITRE IV
ÉVALUATION DE L'APPROCHE PROPOSÉE 50

4.1 Cadre expérimental 50
4.1.1 Algorithmes sélectionnés pour la comparaison 50
4.1.2 Critères d'évaluation 52
4.2 Expérimentations sur les graphes synthétiques 55
4.2.1 Génération des graphes bipartis synthétiques 56
4.2.2 Résultats et discussions 60
4.3 Expérimentations sur les graphes réels 65
4.3.1 Partie 1 : Expérimentations sur un graphe benchmark 66
4.3.2 Partie 2 : Expérimentations sur des graphes réels non étiquetés 72
4.4 Conclusion 77
CHAPITRE V
CONCLUSION ET PERSPECTIVES 78

LISTE DES TABLEAUX

Tableau Page

4.1 Le partitionnement prédéfini du graphe *Southern Women*. 68

4.2 Les valeurs de NMI pour le graphe *Southern Women*. 70

4.3 Statistiques sur les graphes réels non étiquetés. 73

4.4 Valeurs de la *couverture*. 75

4.5 Valeurs de la *densité de modularité* M_D. 75

LISTE DES FIGURES

Figure		Page
1.1	Exemple d'un réseau homogène avec trois communautés.	3
1.2	Un exemple simple d'un graphe biparti.	4
1.3	Un exemple de partitionnement d'un graphe biparti en communautés.	6
1.4	Un graphe biparti divisé en quatre communautés.	7
2.1	Un exemple illustratif de la projection pondérée.	14
2.2	Un exemple illustratif de la projection non pondérée.	15
2.3	Un exemple de partitionnement des approches fondées sur les graphes homogènes.	17
2.4	Les différentes étapes de l'algorithme *LPBrim*	21
2.5	Un exemple du processus d'étiquetage de *LPAb*.	23
2.6	Un exemple d'une matrice bi-adjacence d'un graphe biparti.	28
2.7	Un partitionnement d'un graphe biparti basant sur le modéle de liaison similaire.	30
3.1	Illustration de la transformation d'un graphe biparti en données transactionnelles.	40
4.1	Les matrices bi-adjacences de 6 réseaux bipartis synthétiques avec différents paramètres d'entrées.	59
4.2	Résultats obtenus sur des graphes synthétiques avec le paramètre de mixage *mxp* égal à 0 et différents pourcentages de nœuds non discriminants (n_{nd}).	61
4.3	Résultats obtenus sur des graphes synthétiques avec le paramètre de mixage *mxp* égal à 20 et différents pourcentages de nœuds non discriminants (n_{nd}).	62

4.4 Résultats obtenus sur des graphes synthétiques avec le paramètre de mixage *mxp* égal à 40 et différents pourcentages de nœuds non discriminants (n_{nd}). 64

4.5 Le graphe biparti *Southern Women*. 67

4.6 Structure de communautés identifiée par notre approche sur le réseau *Southern Women*. 69

4.7 Structure de communautés identifiée par l'algorithme *BiLouvain* sur le réseau *Southern Women*. 70

4.8 Structure de communautés identifiée par l'algorithme *Adaptive Brim* sur le réseau *Southern Women*. 71

4.9 Structure de communautés identifiée par l'algorithme *LPBrim* sur le réseau *Southern Women*. 71

4.10 Structure de communautés identifiée par l'algorithme *LPAb+* sur le réseau *Southern Women*. 72

4.11 Structure de communautés identifiée par l'algorithme *LBiSBM* sur le réseau *Southern Women*. 72

RÉSUMÉ

Beaucoup de données du monde réel peuvent être représentées par des graphes hétérogènes qui sont composés de plus d'un type de nœuds et de liens. À titre d'exemple de graphes hétérogènes, on trouve les graphes bipartis. Contrairement à la représentation standard de graphes où les nœuds et les liens sont de même type, les graphes bipartis sont caractérisés par la présence de deux types de nœuds reliés par un seul type de liens. L'analyse des graphes bipartis est l'un des nouveaux défis apparus en forage de données. Parmi les problématiques fondamentales étudiées dans ce domaine, on trouve la détection de communautés dont le but consiste à identifier des groupes de nœuds densément connectés entre eux et faiblement liés avec les nœuds d'autres groupes. Ces communautés ont souvent des caractéristiques similaires, ou des centres d'intérêt commun non connus *à priori*.

La problématique de détection de communautés a été bien étudiée dans le cadre des graphes homogènes. Cependant, dans le contexte des graphes bipartis, cette problématique demeure une question de recherche ouverte. Cela est dû, d'une part, au fait que la structure topologique du graphe biparti est différente de celle du graphe ordinaire/homogène, et d'autre part, l'absence d'une définition universellement acceptée quant à la notion de communauté dans les graphes bipartis. Un certain nombre d'approches a été proposé afin d'identifier des communautés dans les graphes bipartis. Toutefois, ces travaux existants souffrent d'au moins un des problèmes suivants : (1) la perte d'information à la suite de la transformation du graphe biparti en graphes homogènes (2) la sensibilité à la présence des nœuds non discriminants qui cachent la structure de communautés (3) la nécessité de spécifier manuellement le nombre de communautés ainsi que la restriction liée au fait que les deux types de nœuds du graphe biparti doivent avoir le même nombre de communautés.

Afin de pallier les limites des approches existantes (mentionnées ci-dessus), nous présentons dans le cadre de ce mémoire, une nouvelle approche de détection de communautés dans les graphes bipartis. L'approche proposée se base sur une stratégie de détection de communautés qui relate la pertinence des nœuds combinée à une heuristique d'optimisation de la modularité bipartie afin d'identifier les structures de communautés finales. L'efficacité de l'approche proposée est comparée à d'autres méthodes récemment présentées sur des réseaux autant réels que synthétiques. Les résultats de cette comparaison montrent que notre approche offre une performance meilleure et parfois compétitive par rapport aux autres approches.

Mots clés : Graphes bipartis, Détection de communautés, Clustering.

CHAPITRE I

INTRODUCTION

1.1 Mise en contexte

Un réseau d'information est un graphe [1] dont les nœuds sont des entités et les arcs sont des liens qui les relient. De nos jours, on trouve ce type de réseaux dans différents domaines d'applications, tels que les réseaux sociaux comme Facebook et les réseaux de publications comme DBLP. Motivé par la grande quantité de données générées par ces applications, le forage de réseaux d'informations fait l'objet d'un intérêt croissant dans les milieux académiques et industriels. La détection d'anomalies, l'identification de communautés et l'étude de l'évolution du réseau dans le temps sont quelques exemples de problèmes étudiés dans l'analyse de réseaux d'informations. Dans le cadre de ce mémoire, nous nous intéressons à la détection de communautés.

La détection de communautés (aussi appelée clustering de graphe) est une tâche de regroupement de nœuds en plusieurs groupes (ou clusters) en tenant compte des relations qui existent entre les nœuds dans le graphe. Les nœuds d'une même communauté doivent être densément connectés entre eux et faiblement liés avec

1. Dans ce mémoire, les termes réseau et graphe sont utilisés de façon interchangeable pour désigner le même concept.

les nœuds qui forment les autres communautés du graphe (Newman, 2004a) (Lancichinetti et Fortunato, 2009). Ces communautés ont souvent, d'une part, des affinités particulières, ou d'autre part, des caractéristiques similaires, ou encore, des centres d'intérêt commun non connus *à priori*. Les structures de communautés existent dans de nombreux réseaux réels et, dans la plupart du temps, portent une signification concrète en terme d'organisation. À titre d'exemple, dans les réseaux d'interactions protéine-protéine, les communautés correspondent généralement à des ensembles de protéines qui collaborent à une même fonction cellulaire. De ce fait, les algorithmes de détection de communautés présentent un outil précieux qui permet de prédire des fonctions cellulaires pour de nombreuses protéines autrement non caractérisées ((Brun *et al.*, 2003) (Brun *et al.*, 2004)). Dans le Web, les groupes de pages ou de sites fortement connectés traitent souvent des thèmes similaires (Kleinberg et Lawrence, 2001). Dans les réseaux sociaux, les communautés peuvent représenter des groupes d'individus ayant des intérêts communs ou des activités communes (Moody et White, 2003). Dans les systèmes de recommandation, la structure de communautés peut être exploitée pour recommander à une personne les produits les mieux évalués par les membres de sa communauté. Ces produits peuvent être également regroupés en communautés selon les motifs de leurs achats, ce qui permet à nouveau de recommander à un client des produits similaires à ce qu'il a consommé dans le passé. Pour des fins d'illustration, la figure 1.1 montre un petit réseau constitué de trois communautés. Notons que la densité des nœuds dans chaque communauté est plus élevée par rapport à la densité externe entre les communautés.

La plupart des études existantes sur la détection des communautés (Blondel *et al.*, 2008) ; (Raghavan *et al.*, 2007) ; (Pons et Latapy, 2005) et (Newman, 2004b) se concentrent principalement sur les graphes homogènes qui ne contiennent qu'un

seul type de nœuds et de liens (cf. figure 1.1). Or, plusieurs systèmes du monde réel sont naturellement décrits comme des graphes hétérogènes avec plusieurs types de nœuds et de liens. Comme exemple de graphes hétérogènes, on trouve les graphes bipartis qui contiennent deux types de nœuds et un seul type de liens. En effet, les graphes bipartis peuvent être utilisés pour modéliser plusieurs applications telles que le site de vente de produits *Amazon*. Les nœuds dans ce réseau sont des utilisateurs et des produits, alors que les liens (entre utilisateurs et produits) relatent les opérations d'achat.

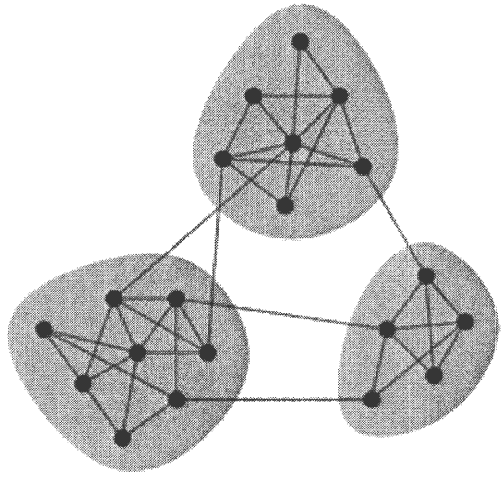


Figure 1.1: Exemple d'un réseau homogène avec trois communautés.

Cette représentation (c.-à-d., en forme de graphe biparti) pose des défis à la tâche de détection de communautés, car la représentation standard (c.-à-d., par des graphes homogènes) est incapable de modéliser les différents types de nœuds et de liens du graphe biparti. En outre, plusieurs méthodes de détection de communautés existantes sont inadéquates pour ce genre de graphes, puisqu'elles ne prennent pas en compte la diversité des nœuds et des liens dans le graphe. Pour pallier ce point, nous nous intéressons, dans le cadre de ce mémoire, à la détection de communautés dans les graphes bipartis.

En théorie des graphes, un graphe est dit biparti si on peut diviser son ensemble de nœuds en deux parties V_1 et V_2, de sorte qu'il n'y ait aucun lien entre les éléments de V_1 et aucun lien entre les éléments de V_2. Autrement dit, les nœuds ayant le même type ne sont jamais connectés et les liens existent seulement entre les nœuds de différents types. La figure 1.2 montre un exemple simple d'un graphe biparti. Dans cette figure, les nœuds de type V_1 représentent les utilisateurs, alors que les nœuds de type V_2 représentent les produits. Les liens dans ce graphe désignent l'opération d'achat de produits par les utilisateurs. À noter que l'utilisation des graphes bipartis n'est pas limitée à relater les interactions d'achats utilisateurs-produits. Un graphe biparti est une modélisation graphique riche qui peut représenter les interactions entre différentes entités (par exemple : les relations de publications co-auteurs-livres dans les réseaux de publications comme DBLP, les relations d'annotations/tags usagers-photos dans les sites de partage de photos et vidéos comme *Flickr*, etc.).

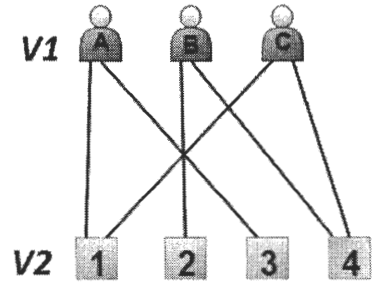
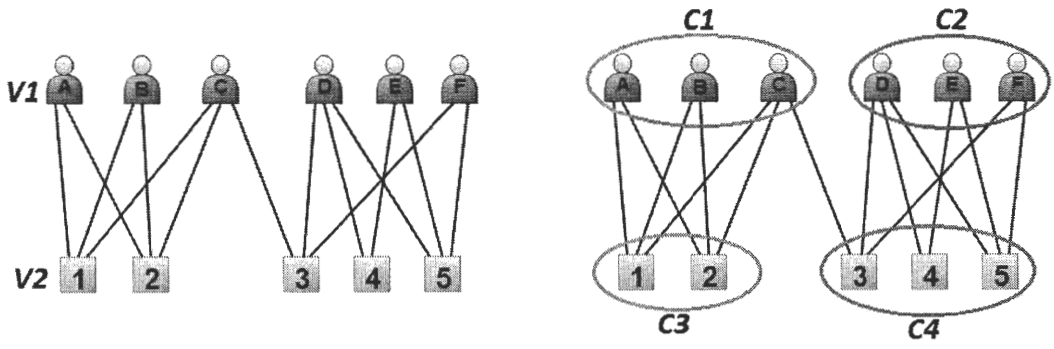


Figure 1.2: Un exemple simple d'un graphe biparti.

La détection de communautés dans les graphes bipartis a retenu une attention remarquable dans la dernière décennie. Toutefois, il convient de noter qu'une définition universellement acceptée quant à la notion de communauté dans un graphe biparti est loin d'être établie. Un certain nombre de travaux adoptent simplement la définition standard d'une communauté dans un graphe simple (un graphe homo-

gène), et ce, dans le contexte des graphes bipartis. Par exemple, Melamed (2014) ; Alzahrani et Horadam (2016) simplifient la tâche de détection en projetant un graphe biparti vers des graphes homogènes. Puis, ils appliquent les algorithmes traditionnels (*Louvain* (Blondel *et al.*, 2008) ; *Label Propagation* (Raghavan *et al.*, 2007) ; *Walktrap* (Pons et Latapy, 2005) et *Fast Greedy* (Newman, 2004b)) pour identifier des structures de communautés. Quelques travaux ((Barber, 2007) ; (Liu et Murata, 2010a) et (Liu et Murata, 2010b)) exploitent les graphes bipartis avec leurs formats de base, c.-à-d., sans l'utilisation de la projection, mais définissent toujours une communauté dans un graphe biparti comme celle dans un graphe simple, à savoir, un groupe de nœuds densément connectés entre eux et faiblement liés avec les nœuds des autres groupes. Cette définition implique que chaque communauté détectée se compose par les deux types de nœuds. En outre, il y a une seule correspondance entre les communautés, c.-à-d., chaque communauté de type V_1 doit être connectée avec une et une seule communauté de type V_2 et vice versa.

D'autres travaux (Suzuki et Wakita, 2009) ; (Murata, 2009) et (Pesantez et Kalyanaraman, 2017) ont défini une communauté dans un graphe biparti comme un ensemble de nœuds de même type qui ont un intérêt commun. Autrement dit, un groupe de nœuds de type V_1 qui se connecte d'une manière dense et similaire avec d'autres groupes de nœuds de type V_2. À partir de cette définition, les auteurs ont proposé des extensions/généralisations d'approches de détection de communautés dans les graphes homogènes pour les adapter au contexte des graphes bipartis.



(a) Un graphe biparti. (b) La partition bipartite.

Figure 1.3: Un exemple de partitionnement d'un graphe biparti en communautés.

La figure 1.3(a) présente un exemple simple d'un graphe biparti. Tel qu'illustré dans la figure 1.3(b), ce graphe est divisé en quatre communautés, à savoir, $C_1 = \{A, B, C\}$; $C_2 = \{D, E, F\}$; $C_3 = \{1, 2\}$ et $C_4 = \{3, 4, 5\}$. Les nœuds de la communauté C_1 sont densément connectés avec les nœuds de sa communauté correspondante C_3 et vice versa. De même, les nœuds de la communauté C_2 sont densément connectés avec les nœuds de sa communauté correspondante C_4 et vice versa. Il convient de noter ici que la communauté correspondante (appelée aussi le $co-cluster$) d'une communauté C_1 de type V_1 est une communauté C_2 de type V_2 de sorte que C_1 partage le nombre maximum de liens avec C_2.

Comme mentionné ci-dessus, le graphe de la figure 1.3 présente un simple exemple d'un graphe biparti avec une structure de communautés relativement claire. Toutefois, dans certaines applications et selon la topologie du graphe sous investigation, les structures de communautés risquent d'être difficilement identifiables. En effet, les graphes bipartis peuvent être caractérisés par la présence de

deux types de nœuds : (1) nœuds discriminants et (2) nœuds non discriminants pour la détection de communautés. Par nœuds discriminants, nous faisons référence aux nœuds qui contribuent à la formation des communautés. Les nœuds non discriminants, ce sont les nœuds qui ne contribuent pas à distinguer les différentes communautés qui existent dans le graphe.

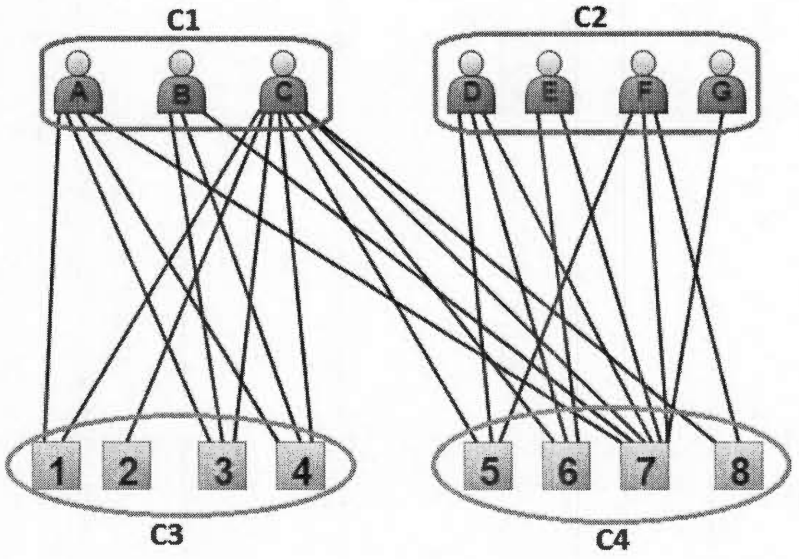


Figure 1.4: Un graphe biparti divisé en quatre communautés.

Pour des fins de clarification, la figure 1.4 illustre un graphe biparti qui relate les interactions d'achats utilisateurs-produits. Dans ce graphe, nous remarquons que l'utilisateur C achète tous les produits existants, alors que le produit 7 a été acheté par tous les utilisateurs existants. Ces deux nœuds, c.-à-d. les nœuds C et 7, sont normalement considérés comme moins importants dans la phase de *clustering*, car ils sont densément connectés avec toutes les communautés, donc, ils ne nous aident pas à discriminer entre les communautés dans le graphe. En outre, on trouve que l'utilisateur G achète un seul produit et le produit 2 a été acheté par un seul utilisateur. Ces deux nœuds, c.-à-d. les nœuds G et 2, sont aussi considérés

comme moins importants pour la phase de détection de communautés, puisqu'ils relatent des connexions éparses et ils ne nous aident pas à distinguer entre les communautés. En revanche, les nœuds A, B, D, E, F, 1, 3, 4, 5, 6 et 8 sont des nœuds pertinents pour la détection de communautés, parce qu'ils sont densément connectés seulement avec les membres de leur communauté correspondante.

1.2 Motivations

La détection de communautés est une problématique qui a été bien étudiée dans le contexte des réseaux homogènes caractérisés par la présence de liens et de nœuds de même type (Fortunato, 2010). Toutefois, dans le cadre des graphes bipartis, cette problématique reste ouverte, étant donné que les liens existent seulement entre les nœuds de différents types et, par conséquent, la définition classique de la communauté (Newman et Girvan, 2004) ne s'applique pas directement. Comme discuté dans la section précédente, la détection de communautés dans les graphes bipartis s'avère donc une tâche difficile pour laquelle un certain nombre de méthodes ont été proposées (Barber, 2007) ; (Liu et Murata, 2010b) et (Liu et Murata, 2010a). En dépit de ces travaux existants, la détection de communautés dans les graphes bipartis continue de défier les approches existantes. En effet, chacune des méthodes existantes souffre d'une ou de plusieurs des limitations suivantes :

1. Certaines approches exigent que l'utilisateur spécifie à l'avance le nombre de communautés dans le graphe biparti. Or, en pratique, il est rarement possible pour l'utilisateur de fournir le nombre exact de communautés. Spécifier le nombre de communautés comme paramètre d'entrées d'un algorithme requiert des connaissances *à priori* sur le graphe sous investigation. En réalité, de telles connaissances ne sont pas toujours disponibles. En outre, de nom-

breuses approches exigent que les deux types de nœuds d'un graphe biparti doivent avoir le même nombre de communautés. Ce scénario s'avère, cependant, irréaliste, car dans plusieurs cas, les deux types de nœuds peuvent avoir un nombre de communautés différent.

2. La vaste majorité des approches existantes traitent les nœuds du réseau d'une manière similaire sans prendre en compte l'importance et la contribution de chaque nœud à la formation de sa communauté. Cette démarche affecte le processus de détection de communautés, étant donné que, comme illustré par la figure 1.4, les graphes bipartis sont caractérisés par la présence de nœuds discriminants et non discriminants. En effet, une présence significative des nœuds non discriminants (principalement les nœuds qui se connectent massivement aux autres nœuds du graphe et les nœuds avec des connexions éparses) affecte les caractéristiques topologiques du réseau sous investigation, ce qui rend la tâche de clustering plus difficile.

3. Quelques approches utilisent des techniques de projection qui consistent à transformer un graphe biparti en un graphe simple dans le but de simplifier la tâche de détection de communautés et de pouvoir, par la même occasion, utiliser les algorithmes standards de détection de communautés. Cependant, cette stratégie de transformation risque d'entraîner une perte substantielle d'informations, étant donné que les caractéristiques structurelles du graphe biparti original sont altérées par le processus de projection/transformation.

1.3 Contributions

Afin de pallier les limitations mentionnées précédemment, nous proposons dans le cadre de ce mémoire une nouvelle approche de détection de communautés dans les graphes bipartis. L'approche proposée vise à identifier des structures de com-

munautés d'une façon automatique et sans aucune connaissance *à priori* telle que le nombre de communautés. Dans ce cadre, la contribution de notre travail est marquée par les éléments suivants :

1. Nous traitons le problème de détection de communautés dans les graphes bipartis comme un problème de clustering de données transactionnelles. Spécifiquement, nous représentons les interconnexions entre les nœuds comme des transactions. À partir de cette représentation, nous élaborons une stratégie de clustering de données transactionnelles combinée à une heuristique d'optimisation de la modularité bipartie afin d'identifier les communautés.

2. L'approche proposée est apte à détecter des communautés automatiquement et ne nécessite pas de paramétrage à spécifier par l'utilisateur, tel que le nombre de communautés. En outre, l'approche proposée dispose d'un mécanisme de pondération (*scoring*), incorporé dans une fonction objective, qui lui permet de distinguer les nœuds discriminants des nœuds non discriminants lors du processus de clustering.

3. Nous présentons une évaluation expérimentale de notre approche sur des graphes autant réels que synthétiques et en comparaison avec d'autres techniques récentes déjà existantes dans la littérature. Les résultats de cette évaluation montrent que notre approche offre une performance compétitive et parfois meilleure par rapport aux autres approches.

1.4 Organisation du mémoire

Ce mémoire est organisé comme suit : le chapitre 2 expose une revue des principaux algorithmes de détection de communautés dans les graphes bipartis. D'un point de vue spécifique, nous présentons les trois familles d'algorithmes avec

des approches existantes et nous discutons de leurs avantages et inconvénients. Le chapitre 3 décrit l'approche proposée d'une manière détaillée. Spécifiquement, dans ce chapitre, nous commençons par introduire la stratégie de traitement qui relate la pertinence des nœuds dans le graphe biparti. Ensuite, nous décrivons notre approche de modélisation en données transactionnelles ainsi que la procédure d'optimisation par le biais d'un algorithme de clustering pour l'identification des communautés dans les graphes bipartis. Le chapitre 4 présente une évaluation empirique de notre approche proposée. Précisément, dans ce chapitre, nous présentons des comparaisons de notre approche avec d'autres approches récentes sur des réseaux réels et synthétiques. Finalement, le chapitre 5 conclut ce mémoire.

CHAPITRE II

REVUE DE LA LITTÉRATURE

Ce chapitre présente une revue sur les approches de détection de communautés dans les graphes bipartis. Plus spécifiquement, il illustre les différentes techniques qui traitent cette problématique avec des exemples illustratifs de quelques approches récemment proposées, ainsi que leurs avantages et limites.

En général, les travaux qui abordent la problématique de détection de communautés dans les graphes bipartis peuvent être divisés en trois grandes familles : (1) les approches basées sur la projection (2) les approches fondées sur les graphes homogènes et (3) les approches fondées sur le modèle de liaison similaire. Dans ce qui suit, nous présentons les travaux représentatifs de chaque famille d'algorithmes.

2.1 Les approches basées sur la projection

Les approches basées sur la projection sont des approches naïves dont le but consiste principalement à simplifier la tâche de clustering en procédant à une transformation du graphe biparti vers des graphes homogènes, de sorte que chaque graphe généré contient des nœuds de même type. D'une manière spécifique, soit le graphe biparti $G = (V_1 \cup V_2, E)$, tel que V_1, V_2 représentent les deux sous-ensembles de nœuds et E désigne l'ensemble de liens dans G. V_1 est défini par $V_1 = \{u_1,$

$u_2,\ldots, u_p\}$, tandis que V_2 est donné par $V_2 = \{v_1, v_2,\ldots, v_q\}$. Supposons que le nœud v_i est connecté à la fois avec u_j et u_k. La projection est une transformation d'une connexion (u_j - v_i - u_k) en une connexion (u_j - u_k)[1] de sorte que le graphe obtenu contient seulement des nœuds ayant le type V_1. Par la suite, un algorithme de détection de communautés dans les graphes homogènes est appliqué sur la nouvelle représentation obtenue pour identifier les structures de communautés. En effet, la projection du graphe G peut se faire de deux différentes manières, à savoir, (1) la projection pondérée et (2) la projection non pondérée.

2.1.1 La projection pondérée

Dans cette stratégie d'approches, les graphes générés (suite à la projection du graphe biparti) sont pondérés par un poids dans chaque lien. Autrement dit, cette stratégie tient en compte le nombre de voisins en commun partagés entre les nœuds dans le graphe biparti initial. Spécifiquement, la projection pondérée du graphe biparti G sur le sous-ensemble de nœuds V_1 est un graphe homogène G_{V1} = (V_1, E_{V1}, ω), tel que E_{V1} représente l'ensemble de liens et ω désigne le poids de chaque lien dans G_{V1}. En effet, ce poids (ω) est mesuré en fonction du nombre de voisins en commun partagés entre chaque paire de nœuds ayant le même type dans G. Notons ici que deux nœuds de type V_1 sont connectés par le biais d'un lien dans E_{V1} si et seulement s'ils partagent au moins un nœud commun de type V_2 dans le graphe biparti original G.

La projection pondérée est généralement considérée comme plus représentative de l'information des liens dans le graphe biparti compte tenu du fait que deux nœuds de même type sont plus susceptibles d'être dans la même communauté s'ils ont de

---

1. Rappelons que, dans le graphe biparti original, les nœuds ayant le même type ne sont pas connectés entre eux.

nombreux voisins communs dans le graphe biparti original, et cette information est conservée par le poids ω.

À titre illustratif, soit le graphe biparti présenté par la figure 2.1(a). La projection pondérée de ce graphe sur les deux sous-ensembles de nœuds V_1 et V_2 est illustrée par les deux figures 2.1(b) et 2.1(c), respectivement. Ici, il convient de noter que les deux graphes générés (cf. figure 2.1(b) et figure 2.1(c)) peuvent être analysés par le biais d'un algorithme de clustering des graphes homogènes pondérés.

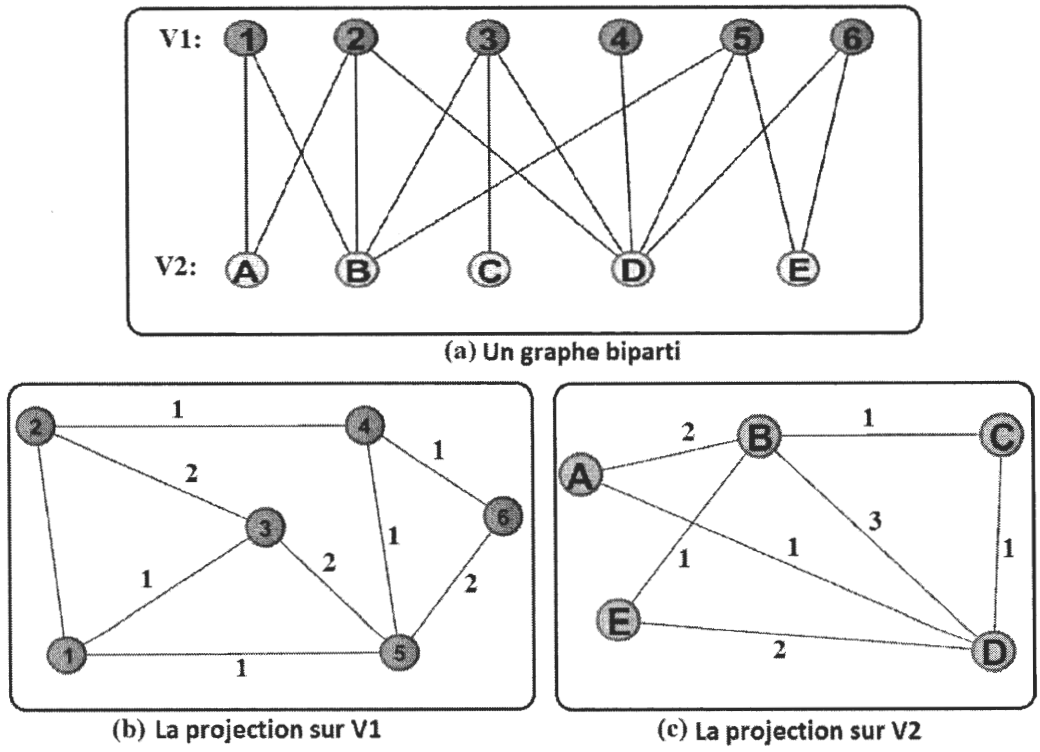


Figure 2.1: Un exemple illustratif de la projection pondérée.

2.1.2 La projection non pondérée

Cette stratégie d'approches est semblable aux techniques basées sur la projection pondérée dans le sens où les graphes générés contiennent des nœuds de même type. Toutefois, la différence réside dans le fait que les liens dans ces graphes (les graphes générés) ne sont pas pondérés. Autrement dit, le nombre de voisins communs partagé entre deux nœuds dans le graphe biparti original n'est pas pris en considération. D'une manière spécifique, la projection non pondérée du graphe biparti G sur le sous-ensemble de nœuds V_1 est un graphe homogène $G_{V1} = (V_1, E_{V1})$, où chaque deux nœuds dans G_{V1} sont connectés par le biais d'un lien dans E_{V1} si et seulement s'ils partagent au moins un nœud de type V_2 dans G.

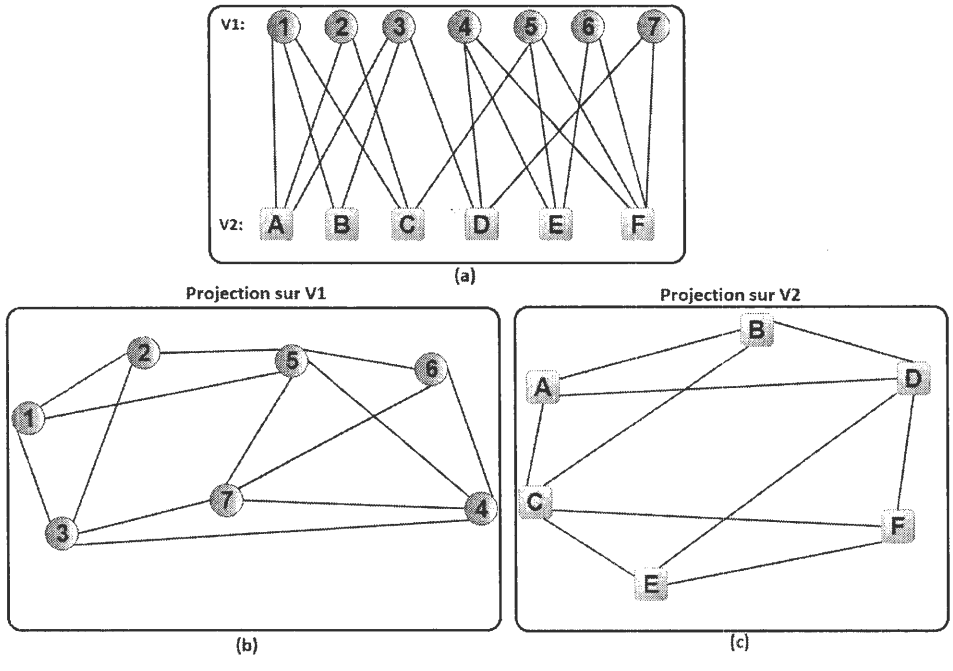


Figure 2.2: Un exemple illustratif de la projection non pondérée.

Prenons le graphe biparti de la figure 2.2(a) comme exemple. La projection non pondérée de ce graphe sur les deux sous-ensembles de nœuds V_1 et V_2 est illus-

trée par les deux figures (2.2(b) et 2.2(c)), respectivement. Les communautés ici peuvent être identifiées en utilisant un algorithme de clustering des graphes homogènes non pondérés sur les projections obtenues.

En terminant, il convient de noter que, bien que la stratégie de projection non pondérée semble être plus simple et facile comparativement à la stratégie de projection pondérée, elle ne relate pas la force de connexion entre les nœuds de même type dans le graphe biparti (cf. figure 2.2). Avec une approche pondérée, l'intensité de connexions entre les différents nœuds du graphe biparti est relatée à travers les poids des liens dans les graphes homogènes générés. Toutefois, l'utilisation de ces deux techniques de transformation (projection pondérée et non pondérée) risque de générer des graphes homogènes qui ne relatent pas nécessairement les caractéristiques spécifiques des graphes bipartis, à savoir, la correspondance entre les communautés.

2.2 Les approches fondées sur les graphes homogènes

Les techniques basées sur cette stratégie définissent une communauté dans un graphe biparti comme un ensemble de nœuds densément connectés entre eux et faiblement liés avec les nœuds de groupes différents. Cela implique que les communautés identifiées basées sur cette stratégie sont composées par les deux types de nœuds. En effet, cette définition a été inspirée de celle du graphe homogène. Pour des fins de clarification, la figure 2.3 (prise de (Liu et Murata, 2010b)) présente un exemple de partitionnement d'un graphe biparti en communautés selon cette stratégie. Ce graphe (celui de la figure 2.3) est divisé en trois communautés, de sorte que chacune contient des nœuds de différents types qui sont d'une part, densément connectés entre eux, et d'autre part, faiblement liés avec les nœuds d'autres communautés. Il convient de noter ici que les approches basées sur cette

stratégie supposent que les deux types de nœuds doivent avoir le même nombre de communautés. Cependant, dans plusieurs cas, les deux types de nœuds du graphe biparti sous investigation peuvent avoir un nombre de communautés différent.

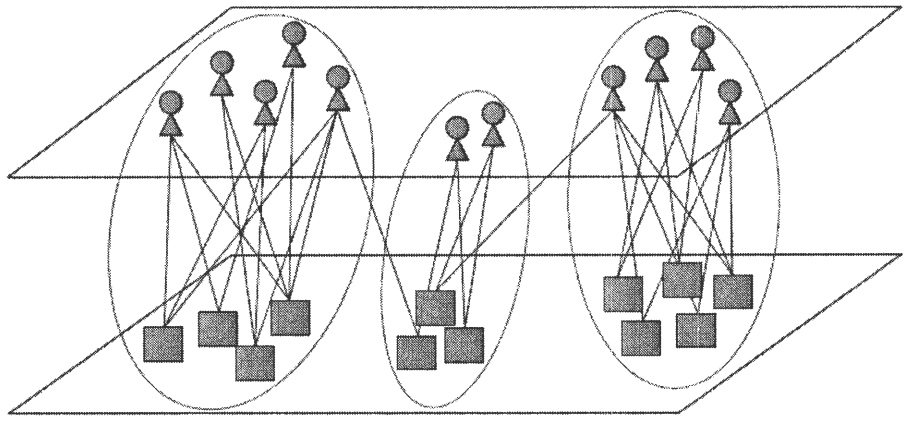


Figure 2.3: Un exemple de partitionnement des approches fondées sur les graphes homogènes.

Barber (2007) a introduit une extension de la modularité de Newman (2006) pour aborder la problématique de détection de communautés dans les graphes bipartis. Rappelons que la modularité est une mesure qui a pour but de caractériser l'efficacité d'une partition des nœuds d'un graphe au regard de la densité des liens au sein des communautés et du nombre de liens entre ces communautés. De ce fait, la modularité est définie par la proportion des liens du réseau qui relient des nœuds issus d'une même communauté, moins la proportion attendue dans un graphe équivalent, dans le sens où les nœuds ont la même distribution des degrés, mais les liens auraient été placés aléatoirement entre tous les nœuds du graphe. La modularité prend une valeur élevée si le nombre de liens intra communautés est important et le nombre de liens inter communautés est faible. Dans le cas contraire, elle prend une valeur faible.

La fonction de modularité proposée par Barber (2007) est considérée, à la fois, comme une métrique de qualité et une fonction objective dans plusieurs approches ((Liu et Murata, 2010b), (Barber et Clark, 2009)). Soit $G = (V_1 \cup V_2, E)$ un graphe biparti et p, q dénotent le nombre de nœuds dans V_1, V_2, respectivement. E représente l'ensemble de m liens dans G. La fonction objective de Barber (2007) est donc définie comme suivante :

$$Q_B = \frac{1}{m} \sum_{u=1}^{p} \sum_{v=1}^{q} \tilde{B}_{uv} \delta(s_u, t_v) \tag{2.1}$$

$\tilde{B}_{uv}$ représente la matrice de dimension $(p \times q)$, où chaque élément de cette matrice est défini par : $\tilde{B}_{uv} = \tilde{A}_{uv}$ -$\tilde{P}_{uv}$. Ici, $\tilde{A}_{uv}$ représente un élément de la matrice $\tilde{A}_{p*q}$ qui est égal à 1 si les nœuds u et v sont directement connectés et 0 sinon. $\tilde{P}_{uv} = \frac{d_u d_v}{m}$ est la probabilité qu'il existe un lien entre u et v. Dans ce cas, d_u, d_v représentent le nombre de voisins du nœud u et v, respectivement. $\delta(s_u, t_v)$ est le symbole de Kronecker, il retourne 1 si u et v appartiennent à la même communauté. Dans les autres cas, il retourne un 0. La modularité de Barber (2007) est élevée si les nœuds au sein de chaque communauté sont densément connectés entre eux et faiblement reliés avec les nœuds d'autres groupes.

Barber (2007) a également introduit un algorithme de détection de communautés dans les graphes bipartis, appelé *BRIM* (*Bipartite Recursively Induced modules*). C'est un algorithme itératif conçu pour maximiser la modularité bipartie définie par l'équation (2.1). *BRIM* est basé sur l'idée que les communautés dans les deux types de nœuds sont dépendantes, avec chaque partie étant mutuellement utilisée pour induire les nœuds des communautés de l'autre partie. Initialement, l'algorithme suppose que l'un des deux types de nœuds (supposons les nœuds de type V_2) est associé à des clusters par le biais d'un certain mécanisme. Puis, deux phases itératives se succèdent :

Lors de la première phase, chaque nœud $u \in V_1$ est associé à une communauté qui maximise l'équation suivante :

$$Q_{B1} = \frac{1}{m} \sum_{u=1}^{p} \left(\sum_{k=1}^{c} R_{uk} \tilde{T}_{uk} \right) \tag{2.2}$$

R est une matrice binaire de dimension $q \times c$ (q désigne le nombre de nœuds dans V_2 alors que c dénote le nombre de communautés) qui indique l'appartenance des nœuds de type V_2 à des communautés. La matrice $\tilde{T}$ est définie par : $\tilde{T} = T\tilde{B}$, avec $\tilde{B} = \tilde{A}$ - $\tilde{P}$. Ici, de manière similaire à R, T une matrice $p \times c$ (p étant le nombre de nœuds dans V_1) qui relate l'appartenance des nœuds de type V_1 à des communautés. Notons que chaque ligne de la matrice R (de même pour T) contient un seul 1 qui dénote l'appartenance d'un nœud à une communauté et tous les autres éléments sont 0. Intuitivement, la maximisation de l'équation (2.2) consiste à associer chaque nœud u à une communauté k de telle sorte que $\tilde{T}_{uk}$ prend la valeur maximale de la u$^{\text{éme}}$ ligne de la matrice $\tilde{T}$.

Une fois que tous les nœuds dans V_1 sont associés à leurs communautés, la deuxième phase est amorcée afin d'associer chaque nœud $v \in V_2$ à la communauté qui maximise l'équation suivante :

$$Q_{B2} = \frac{1}{m} \sum_{v=1}^{q} \left(\sum_{k=1}^{c} T_{vk} \tilde{R}_{vk} \right) \tag{2.3}$$

La matrice $\tilde{R}$ est définie par : $\tilde{R} = \tilde{B}R$. Similairement aux nœuds dans V_1, la maximisation de l'équation (2.3) consiste à associer chaque nœud v à une communauté k de telle sorte que $\tilde{R}_{vk}$ prend la valeur maximale de la v$^{\text{éme}}$ ligne de la matrice $\tilde{R}$.

Les deux phases de l'algorithme *BRIM* sont répétées itérativement jusqu'à ce qu'un maximum local de la modularité bipartie soit atteint. Notons que pour chaque itération, la modularité Q_B est garantie de ne jamais diminuer, parce qu'il est possible de garder la modularité obtenue et maintenir le précédent partitionnement de nœuds. Donc, l'algorithme *BRIM* trouvera toujours une partition avec une valeur maximale de la modularité Q_B. En général, la partition identifiée correspond à un maximum local de Q_B et pas un maximum global. Notons aussi que l'algorithme *BRIM* peut fonctionner avec n'importe quel nombre de communautés supposé initialement. Cependant, aucune contrainte n'existe pour s'assurer que chaque communauté détectée contient des nœuds.

En se basant sur l'algorithme *BRIM*, Barber (2007) a présenté également un autre algorithme appelé *Adaptive BRIM*. Le principe de ce dernier consiste initialement à associer tous les nœuds du graphe à une seule communauté ($k = 1$), donnant ($Q_B = 0$). Ensuite, le nombre de communautés k est doublé. À cet effet, la moitié des nœuds sont réaffectés aléatoirement aux communautés nouvellement définies, et une nouvelle solution localement optimale est trouvée en utilisant l'algorithme *BRIM*. Ce processus se poursuit, avec k étant doublé à plusieurs reprises tant que Q_B continue à augmenter. Une fois que Q_B diminue à mesure que k augmente, la valeur maximale de la modularité a été atteinte. Puis, une méthode de bi-section (division) est adoptée afin de chercher la valeur optimale de k, celle qui correspond à la division qui possède la plus grande valeur de la modularité bipartie Q_B par rapport aux autres divisions. Cette division est donc considérée comme la partition finale. *Adaptive BRIM* fonctionne bien dans les réseaux bipartis de petite taille. Toutefois, il trouve des difficultés avec les graphes à grande échelle. Cela est causé par le processus de recherche de la meilleure valeur de c qui prend beaucoup de temps.

Pour rendre *Adaptive BRIM* applicable aux réseaux à grande échelle, Liu et Murata (2010a) ont introduit un nouvel algorithme appelé *LPBrim*. Ce dernier (*LPBrim*) représente une combinaison de l'algorithme LPA (*Label Propagation Algorithm* – initialement conçu pour les graphes homogènes) (Raghavan *et al.*, 2007) et l'algorithme *BRIM*. Ce nouvel algorithme utilise *LPA* pour trouver une division initiale et ensuite *BRIM* afin d'identifier la structure de communautés finale. En général, *LPBrim* est capable d'identifier des communautés dans des graphes à grande échelle de manière précise et rapide. La figure 2.4 ci-dessous illustre les différentes étapes de l'algorithme *LPBrim*.

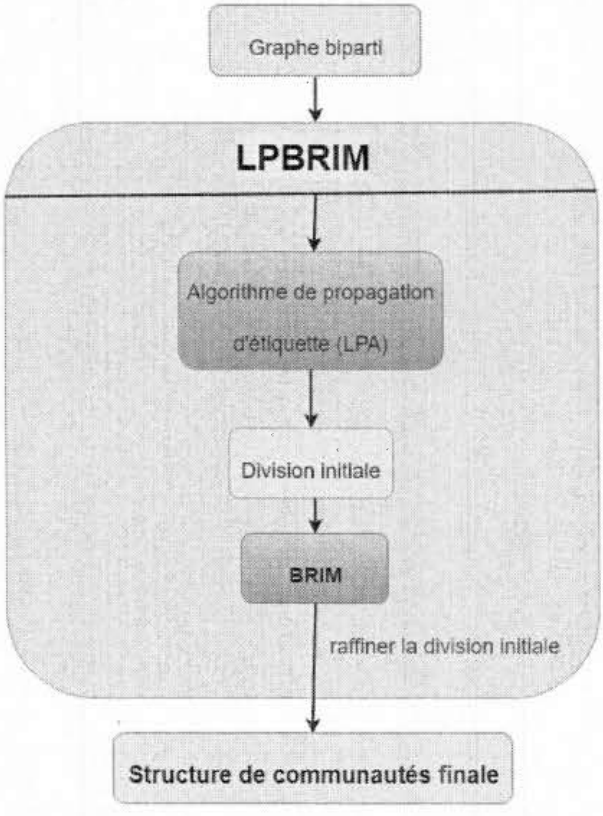


Figure 2.4: Les différentes étapes de l'algorithme *LPBrim* .

Dans un autre travail, Barber et Clark (2009) ont proposé une approche appelée *LPAb* (*Label Propagation Algorithm for Bipartite Graphs*), qui s'inspire de l'algorithme *LPA*. *LPAb* est un algorithme d'optimisation de la modularité bipartie de Barber (2007), donnée par l'équation (2.1). Au départ, chaque nœud est associé à une étiquette unique qui indique sa communauté initiale. Puis, l'algorithme implémente un processus itératif, de sorte qu'à chaque itération les nœuds mettent à jour leurs étiquettes dans une séquence aléatoire pour maximiser la modularité. Spécifiquement, chaque nœud $x \in (V_1 \vee V_2)$ met à jour son étiquette en utilisant l'une des deux fonctions suivantes :

Si $x \in V_1$:

$$s_x^{new} = \arg\max_s \left(\sum_{v=1}^{q} \tilde{B}_{xv} \delta(s, l_v) \right) \tag{2.4}$$

Si $x \in V_2$:

$$t_x^{new} = \arg\max_t \left(\sum_{u=1}^{p} \tilde{B}_{xv} \delta(s_u, t) \right) \tag{2.5}$$

s_x^{new}, t_x^{new} représentent les nouvelles étiquettes du nœud x. Dans le cas où deux ou plusieurs étiquettes maximiseraient la somme, la fonction arg max doit choisir aléatoirement une de ces étiquettes. Ce processus d'étiquetage est répété itérativement jusqu'à ce qu'un maximum local de la modularité bipartie soit atteint. Finalement, les communautés sont identifiées comme des groupes de nœuds partageant la même étiquette. Ici, il convient de noter que les nœuds dans V_1 utilisent uniquement les informations des nœuds dans V_2 pour mettre à jour leurs étiquettes. De même, les nœuds dans V_2 mettent à jour leurs étiquettes en utilisant seulement les informations sur les nœuds dans V_1.

Afin d'illustrer le principe de fonctionnement de l'algorithme *LPAb*, considérons le graphe biparti présenté par la figure 2.5(a). Supposons que les quatre nœuds colorés en rouge (étiquetés par 1, 2, 3 et 4) sont de type V_1 et les cinq nœuds

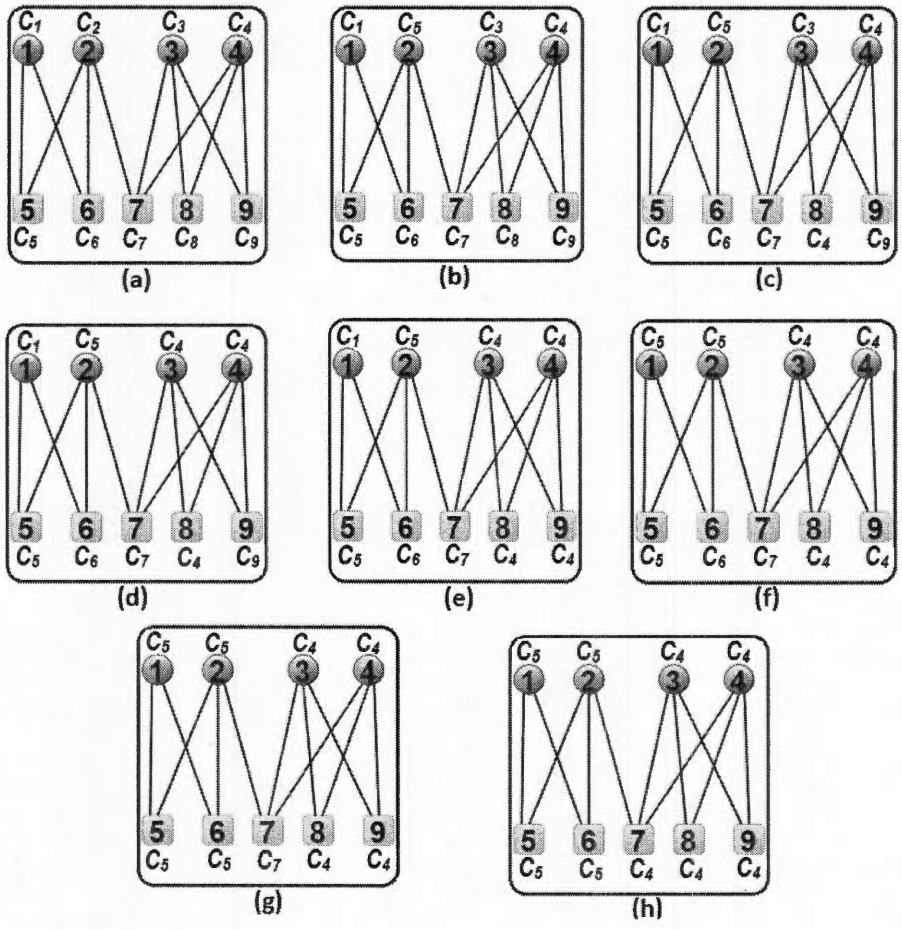


Figure 2.5: Un exemple du processus d'étiquetage de *LPAb*.

colorés en vert (étiquetés par 5, 6, 7, 8 et 9) sont de type V_2 (9 nœuds au total). Initialement, chaque nœud appartient à une seule communauté donnant 9 communautés (C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9), comme illustré dans la figure 2.5(a). L'algorithme commence par choisir un nœud d'une façon aléatoire. Supposons qu'il a choisi le nœud 2 qui porte l'étiquette (C_2). Comme ce nœud appartient à l'ensemble V_1, alors l'équation (2.4) est utilisée pour trouver la nouvelle étiquette de ce dernier (le nœud 2). Dans ce cas, seuls les nœuds 5 et 6 maximisent l'équation (2.4). Donc, le nœud 2 peut prendre l'une des deux étiquettes (C_5 ou C_6).

Supposons que l'algorithme a choisi d'associer le nœud 2 à la communauté C_5 (figure 2.5(b)). Par la suite, l'algorithme a choisi aléatoirement de traiter le nœud 8 qui se trouvait dans la communauté C_8. Ce dernier a deux voisins, à savoir, les nœuds 3 et 4, et chacun appartient à sa communauté initiale. Comme le nœud 8 est de type V_2, donc l'équation (2.5) est appliquée. Dans ce cas, les deux nœuds (3 et 4) maximisent l'équation (2.5). Ici, l'algorithme a choisi d'associer le nœud 8 à la communauté C_4 (figure 2.5(c)). Le prochain nœud à traiter est le nœud 3 qui possède trois voisins (7, 8 et 9). Dans ce cas, la maximisation de l'équation (2.4) donne deux possibilités, à savoir, les nœuds 8 et 9. Supposons que l'algorithme a choisi le nœud 8 qui appartient à la communauté C_4 (figure 2.5(d)). Le troisième nœud à traiter est le nœud 9. Ce nœud a deux voisins (les nœuds 3 et 4) qui appartiennent à la même communauté C_4. Donc, intuitivement, le nœud 9 prend l'étiquette C_4 (figure 2.5(e)).

Le prochain nœud à traiter est le nœud 1 qui a deux voisins (5 et 6) et avec chacun appartient à sa communauté initiale (c.-à-d., C_5 et C_6 respectivement). Ici, la maximisation de l'équation (2.4) donne deux possibilités, à savoir, la communauté C_5 ou la communauté C_6. Dans ce cas, l'algorithme a choisi d'associer le nœud 1 à la communauté C_5 (la figure 2.5(f)). En appliquant ce processus d'étiquetage sur tous les autres nœuds du graphe jusqu'à il n'y ait aucun changement de communauté entre les nœuds. À ce compte-là, l'algorithme s'arrête et les communautés sont identifiées comme des nœuds partageant la même étiquette. Dans notre exemple (la figure 2.5), et comme illustré dans la figure 2.5(h), deux communautés sont obtenues après ce processus d'étiquetage, à savoir, la communauté C_4 et la communauté C_5.

Comme l'algorithme *LPAb* est une extension de *LPA* (Raghavan *et al.*, 2007), il hérite les mêmes caractéristiques que ce dernier. De ce fait, il dispose de nombreuses qualités souhaitables, à savoir, la simplicité, la facilité de compréhension et la rapidité d'implémentation. En effet, le grand avantage de *LPAb* réside dans le fait qu'il est considéré comme l'algorithme d'optimisation de la modularité bipartie le plus rapide dans les graphes bipartis avec une complexité proche de $O(m)$ où m représente le nombre de liens dans le graphe biparti. En dépit de ces avantages souhaitables, la maximisation de la modularité bipartie Q_B (l'équation (2.1)) ne garantit pas forcément un partitionnement optimal du graphe. Cela s'explique par le fait que le maximum global de cette modularité donne une solution triviale où tous les nœuds possèdent la même étiquette, c.-à-d., que tous les nœuds appartiennent à la même communauté. De plus, la procédure d'optimisation de *LPAb* (l'équation (2.4) et (2.5)) ne produit que des changements locaux, ce qui rend la maximisation de la modularité bipartie (l'équation (2.1)) orientée vers un optimum local au lieu de l'optimum global. Toutefois, cette stratégie de recherche locale permet à *LPAb* d'éviter la solution triviale (c.-à-d., de mettre tous les nœuds dans une seule communauté) et de tirer des structures de communautés (Barber et Clark, 2009).

Pour de pallier les limites de *LPAb*, à savoir, l'incertitude causée par la séquence de mise à jour aléatoire et la convergence vers des optimums locaux, une nouvelle approche, appelée *LPAb+*, a été proposée par Liu et Murata (2010b). C'est une approche hybride combinant une extension de *LPAb* (appelée *LPAb'*) et l'algorithme MSG (Multistep greedy agglomerative algorithm) proposé dans (Schuetz et Caflisch, 2008a) et (Schuetz et Caflisch, 2008b). Le principe ici consiste à utiliser l'algorithme MSG pour conduire *LPAb* loin des maximums locaux. Plus spécifiquement, *LPAb+* est divisé en deux phases itératives correspondantes à *LPAb'* et

MSG, respectivement. Dans la phase 1 (c.-à-d., *LPAb'*), au lieu de mettre à jour les nœuds du graphe d'une façon aléatoire dans chaque étape, comme c'est le cas avec *LPAb*, les deux types de nœuds (V_1 et V_2) sont traités séparément. Au début, les nœuds dans V_1 mettent à jour leurs étiquettes en utilisant l'équation (2.4), puis les nœuds dans V_2 font la même chose par le biais de l'équation (2.5). Après ce processus de mise à jour, *LPAb'* converge vers *"a poor local maximum"* (Liu et Murata, 2010b) donnant une faible valeur de la modularité bipartie. Donc, les communautés sont identifiées comme des nœuds partageant la même étiquette. Il est à noter ici que l'avantage de *LPAb'* sur LPAb réside dans l'élimination du facteur d'incertitude causé par la séquence de mise à jour aléatoire. Quand la modularité bipartie ne peut plus être augmentée dans la phase 1 (*LPAb'*), un maximum local de la modularité est atteint. Cependant, cela peut ne pas être le maximum global. De ce fait, la phase 2 cherche à pousser *LPAb'* loin des maximums locaux en utilisant l'algorithme MSG. En effet, MSG est un algorithme d'optimisation de la modularité permettant la fusion de plusieurs paires de communautés dans chaque itération. Spécifiquement, chaque deux communautés (C_1, C_2) identifiées dans la phase 1 sont fusionnées, de telle sorte que la fusion de C_1 avec C_2 amène un changement positif maximal de la modularité.

Une fois la deuxième phase terminée, la phase 1 est appliquée de nouveau afin d'atteindre un autre maximum local. Ces deux phases sont répétées itérativement jusqu'à ce qu'il ne soit plus possible d'augmenter la modularité du graphe en fusionnant n'importe quelle paire de communautés. Finalement, les communautés sont identifiées comme des nœuds partageant la même étiquette. Un problème important à noter ici est que les résultats de *LPAb+* varient à chaque exécution faisant de *LPAb+* un algorithme indéterministe, ce qui nécessite plusieurs exécutions afin de trouver la meilleure valeur de la modularité.

2.3 Les approches fondées sur le modèle de liaison similaire

Contrairement aux approches précédentes, les approches fondées sur le modèle de liaison similaire visent en effet à introduire certaines caractéristiques liées aux graphes bipartis telles que la correspondance entre les communautés. De ce fait, ces approches définissent une communauté dans un graphe biparti comme un ensemble de nœuds, du même type, qui se connectent d'une manière dense et semblable avec d'autres groupes de nœuds de différent type. Cela implique qu'une communauté de type V_1 peut être connectée avec une ou plusieurs communautés de type V_2 et vice versa. De plus, les deux types de nœuds peuvent avoir un nombre de communautés différent.

Murata (2009) s'est penché sur cette stratégie d'approches en proposant une nouvelle extension de la modularité de Newman (2004a) pour la détection de communautés dans les graphes bipartis. Cette modularité mesure la fraction de liens dans le graphe biparti qui connectent les nœuds de chaque couple de communautés correspondantes, moins la valeur attendue de la même quantité avec des connexions aléatoires entre ces couples de communautés correspondantes.

Soit G un graphe biparti qui est défini par : $G = (V_1 \cup V_2, M)$, avec V_1, V_2 représentant les deux sous-ensembles de nœuds et M l'ensemble de liens dans G, soit $V = V_1 + V_2$. Supposons que N_{V1}, N_{V2} dénotent le nombre de communautés dans V_1 et V_2, respectivement, et C^{V1}, C^{V2} représentent l'ensemble de communautés dans V_1 et V_2, respectivement. C_l et C_m sont deux communautés qui appartiennent à l'ensemble ($C^{V1} = \{ C_1^{V1},.., C_{N_{V1}}^{V1} \}$, $C^{V2} = \{ C_1^{V2},.., C_{N_{V2}}^{V2} \}$). $A(i, j)$ la matrice bi-adjacence du graphe G où chaque élément de cette matrice est égal à 1 si les nœuds i et j sont directement connectés et vaut 0 dans le cas contraire. La matrice A dispose d'un bloc hors diagonale sous forme du bloc-matrice A suivant :

$$A = \left(\begin{array}{ccc|ccc} 0 & \cdots & 0 & e_{1,L^X}+1 & \cdots & e_{1,L^X}+L^Y \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & e_{L^X,L^X}+1 & \cdots & e_{L^X,L^X}+L^Y \\ \hline e_{L^X+1,1} & \cdots & e_{L^X+1,L^X} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ e_{L^X+L^Y,1} & \cdots & e_{L^X+L^Y,L^X} & 0 & \cdots & 0 \end{array}\right)$$

Figure 2.6: Un exemple d'une matrice bi-adjacence d'un graphe biparti.

Notons que les nœuds dans C_l et C_m sont de différents types, en d'autres termes $(C_l \in C^{V1} \wedge C_m \in C^{V2}) \vee (C_m \in C^{V1} \wedge C_l \in C^{V2})$. Soit e_{lm} la fraction de tous les liens qui connectent les nœuds de C_m avec les nœuds de C_l. e_{lm} est définie par l'équation suivante :

$$e_{lm} = \frac{1}{2M} \sum_{i \in V_l} \sum_{j \in V_m} A(i,j) \tag{2.6}$$

Soit a_l la fraction de tous les liens dans V_l :

$$a_l = \sum_m e_{lm} = \frac{1}{2M} \sum_{i \in V_l} \sum_{j \in V} A(i,j) \tag{2.7}$$

La modularité bipartie de Murata (2009) est donc définie par l'équation suivante :

$$Q_M = \sum_l Q_{M_l} = \sum_l (e_{lm} - a_l a_m), avec \quad m = \arg\max_k (e_{lk}) \tag{2.8}$$

En effet, l'équation (2.8) peut être divisée comme la suivante :

$$Q_M^{V_1 V_2} = \sum_{l \in V_1} (e_{lm} - a_l a_m), avec \quad m = \arg\max_{k \in V_2} (e_{lk}) \tag{2.9}$$

$$Q_M^{V_2V_1} = \sum_{l \in V_2} (e_{lm} - a_l a_m), avec \quad m = \arg\max_{k \in V_1}(e_{lk}) \tag{2.10}$$

Ici, $Q_M^{V_1V_2}$ représente la modularité bipartie pour la direction $V_1 \rightarrow V_2$ et $Q_M^{V_2V_1}$ désigne la modularité bipartie pour la direction $V_2 \rightarrow V_1$.

Finalement, Q_M peut être donnée comme suivante :

$$Q_M = Q_M^{V_1V_2} + Q_M^{V_2V_1} \tag{2.11}$$

Contrairement à la modularité de Barber (2007), définie par l'équation (2.1), la modularité de Murata (2009), définie par l'équation (2.8), supporte une correspondance d'une à plusieurs, c.-à-d., une communauté de type V_1 peut avoir une ou plusieurs communautés correspondantes dans V_2 ou le contraire. En outre, elle fournit un poids qui relate le degré de correspondance (la force de connexion) entre chaque communauté et son co-cluster [2]. Ce degré peut être utilisé pour analyser les caractéristiques des communautés dans le graphe biparti. Un autre avantage de cette mesure est qu'elle peut être également appliquée dans les graphes homogènes et que les résultats obtenus sont cohérents avec ceux de la modularité de Newman (2006). La figure 2.7 illustre un exemple de partitionnement du graphe biparti en communautés selon Murata (2009). Notons que dans cette figure, les deux types de nœuds ont un nombre de communautés différent. Cette modularité semble appropriée aux graphes bipartis, puisqu'elle prend en compte certaines caractéristiques liées aux graphes bipartis, à savoir, la correspondance entre les nœuds. Toutefois, aucun algorithme n'existe dans la littérature permettant de la maximiser.

---

2. Dans le reste de ce document, les termes communauté correspondante et *co-cluster* sont utilisés d'une manière interchangeable pour désigner le même concept.

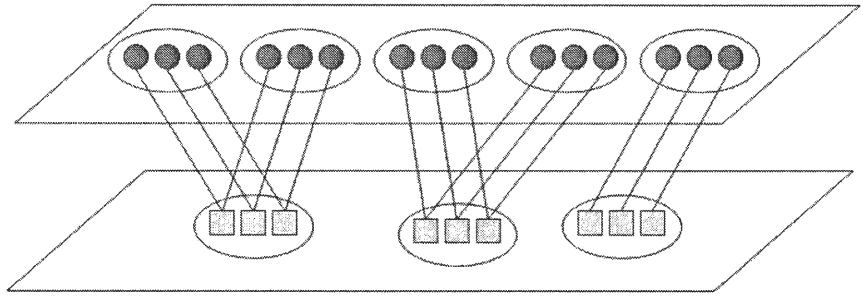


Figure 2.7: Un partitionnement d'un graphe biparti basant sur le modéle de liaison similaire.

Suzuki et Wakita (2009) ont proposé une version modifiée de la modularité de Murata (2009) pour tenir compte des correspondances à plusieurs facettes (des correspondances de plusieurs à plusieurs). L'idée se base sur le principe que chaque individu dans le réseau social a plus d'un seul intérêt, et peut donc être utilisé pour extraire des structures de communautés à plusieurs facettes. Par ce fait, Suzuki et Wakita ont remplacé la fonction arg max (définie par l'équation (2.9) et (2.10)) par un facteur de proéminence qui relate la force de connexion entre deux communautés de différents types. Ce facteur est défini par la proportion du nombre de liens entre la communauté C_m et la communauté C_l, et le nombre de toutes les arêtes que la communauté C_m possède. Suzuki et Wakita (2009) ont utilisé la première phase de l'algorithme *Louvain* (proposé par Blondel *et al.* (2008)), un algorithme itératif dédié aux graphes homogènes pour optimiser leur nouvelle version de la modularité. À titre de rappel, la première phase de l'algorithme *Louvain* se base principalement sur la procédure VM (*Vertex Mover* (Schuetz et Caflisch, 2008a)), et comme cette procédure produit juste des changements locaux basés sur le gain de la modularité qui résulte du processus de fusion de nœuds,

l'algorithme finit par produire une partition avec un nombre de communautés élevé et une faible valeur de modularité bipartie (plus faibles que celle de Murata (2009)).

Larremore *et al.* (2014) ont proposé un modèle appelé *BiSBM* (Bipartite Stochastic Block Model) qui s'inspire du modèle *SBM* (proposé initialement dans (Karrer et Newman, 2011)). Le principe consiste à maximiser une fonction de probabilité en déplaçant les nœuds entre les communautés. En effet, l'algorithme nécessite deux paramètres d'entrées, à savoir, N_{V1} qui représente le nombre de communautés dans V_1 et N_{V2} le nombre de communautés dans V_2. Initialement, l'algorithme commence par associer d'une manière aléatoire les nœuds de type V_1 aux groupes N_{V1} et les nœuds de type V_2 aux groupes N_{V2}. Puis, l'algorithme cherche la surface de probabilité en proposant de déplacer chaque nœud entre les communautés de ces voisins (ayant le même type). Après avoir proposé ces mouvements, à travers tous les nœuds et groupes admissibles, il choisit le mouvement qui maximisera la fonction de probabilité. Si aucune amélioration n'est possible, l'algorithme choisit le mouvement qui diminue le moins possible la fonction de probabilité car forcer les nœuds à se déplacer permet d'échapper aux optimums locaux. Quand tous les nœuds se sont déplacés, les états par lesquels l'algorithme a passé sont évalués et l'état avec le score le plus élevé est utilisé comme point de départ de la prochaine itération. Lorsqu'une itération complète passe sans aucune amélioration dans la fonction de probabilité, l'algorithme s'arrête et les communautés sont identifiées comme des nœuds ayant la même étiquette. À noter que l'algorithme doit être exécuté plusieurs fois et le score le plus élevé parmi les résultats obtenus est choisi. Comme mentionné ci-dessus, l'algorithme *BiSBM* est sensible à certains paramètres, notamment le nombre de communautés dans les deux types de nœuds (N_{V1} et N_{V2}). En effet, ces deux paramètres peuvent grande-

ment affecter la précision de cet algorithme si les valeurs fournies par l'utilisateur sont incorrectes. Un autre inconvénient réside dans le temps d'exécution excessif faisant de *BiSBM* un algorithme indéterministe.

Dans un travail récent, Pesantez et Kalyanaraman (2017) proposent une version modifiée de la modularité de Murata (2009), appelée *Murata+*, en ajoutant un terme négatif dans la fonction arg max. Ce terme est défini par le produit de la fraction de tous les liens reliant les deux communautés en question.

D'une manière spécifique, la fonction arg max est redéfinie comme suit :

Soit C_m la communauté correspondante de C_l :

$$C_m = \arg\max_m(e_{lm} - a_l a_m) \tag{2.12}$$

De même, C_l la communauté correspondance de C_m :

$$C_l = \arg\max_l(e_{lm} - a_l a_m) \tag{2.13}$$

En effet, le terme négatif $(a_l a_m)$ est ajouté dans la fonction arg max afin de rendre la correspondance entre les communautés symétriques, c.-à-d., que si une communauté C_l a choisi C_m comme sa communauté correspondante, alors C_m doit également choisir C_l comme sa communauté correspondante.

De plus, Pesantez et Kalyanaraman (2017) introduisent *BiLouvain*, une approche qui s'inspire de l'algorithme *Louvain* (proposé initialement dans (Blondel *et al.*, 2008)). Globalement, *BiLouvain* suit le même modèle d'optimisation élaboré initialement par *Louvain*, il diffère dans la fonction objective (en utilisant *Murata+*) et dans la Méthode de calcul du gain de modularité.

L'algorithme *BiLouvain* se base sur les étapes suivantes :

1. Soit le graphe biparti $G = (V_1 \cup V_2, M, \omega)$, initialise un ensemble de n_1+n_2 communautés, tel que $n_1 = |V_1|$ et $n_2 = |V_2|$, en plaçant chaque nœud dans sa propre communauté.

2. Dans chaque itération, tous les nœuds sont scannés linéairement (dans un ordre arbitraire). Pour chaque nœud x , il faut :

   (a) Acquérir une liste de communautés candidates auxquelles il peut éventuellement se fusionner. Supposons que $x \in V_1$ et $\Gamma(x)$ dénote l'ensemble de voisins de x qui appartient à V_2, alors :

   $\Gamma(x) = \{j/(x,j) \in M\}$

   Soit $\Gamma(x)'$ dénote l'ensemble de nœuds dans V_1 qui sont connectés avec les voisins de x, donc :

   $\Gamma(x)' = \{k/(k,j) \in M, \quad j \in \Gamma(x)\}$

   Par conséquent, l'ensemble des communautés candidates du nœud x est donné par :

   $Cand(x) = \bigcup_{k \in \Gamma(x)'} C(k)$

   De la même façon, on peut déterminer les communautés candidates de tous les nœuds dans V_2.

   (b) Évaluer le gain de modularité qui résulterait du scénario de fusion du nœud x avec chacune de ses communautés candidates ($Cand(x)$).

   (c) Finalement, fusionner le nœud x avec la communauté candidate qui maximise le gain de modularité, seulement si ce gain est positif, sinon le nœud x reste dans sa communauté.

3. Cette phase se termine, c.-à-d., phase 1 qui correspond à l'étape 2 ci-dessus,

lorsque le gain de la modularité entre deux itérations consécutives soit négligeable.

4. Une fois que la première phase est terminée, la phase 2 est commencée. De ce fait, un nouveau graphe $G' = (V_1' \cup V_2', M', \omega')$ est généré à travers une étape de compactage qui transforme chaque communauté en un nœud et les liens entre ces communautés sont remplacés par une seule connexion à laquelle on associe un poids qui relate le nombre de liens reliant les deux communautés en question. Après cette phase de compactage, la modularité globale du graphe est mesurée.

5. Le nouveau graphe compacté G' est entré dans la phase suivante (phase 1) et l'algorithme se termine lorsque deux phases consécutives entraînent une valeur de modularité globale négligeable (celle mesurée dans la phase 2).

2.4 Conclusion

Dans ce chapitre, nous avons présenté une vue d'ensemble sur le fonctionnement de quelques approches de détection de communautés dans les graphes bipartis. Nous avons également discuté de leurs avantages et limites. Comme mentionné dans l'i ntroduction, la plupart des approches existantes traitent les nœuds du graphe d'une manière similaire sans prendre en considération la contribution explicite de chaque nœud à la formation de communautés. À titre d'exemple, un nœud de type V_1 se connecte massivement à tous les nœuds de V_2 et vice versa. Ce type de connexions risque de bien cacher la structure de communautés, si le processus de détection se base sur un mécanisme qui traite les nœuds de la même façon sans faire la distinction explicite entre les nœuds discriminants et non discriminants. Certaines autres approches souffrent également de la perte d'informations causée par l'utilisation des techniques de projection du graphe biparti

vers des graphes homogènes, dont le but est de simplifier la tâche de détection de communautés. En outre, nous avons démontré aussi que d'autres approches traitent les graphes bipartis comme des graphes homogènes. Par conséquent, ces approches restreignent la correspondance entre les communautés à une correspondance unique, c.-à-d., une communauté de type V_1 doit être connectée avec une et une seule communauté de type V_2 et vice versa. Afin de pallier ces limites, nous présentons dans le chapitre suivant, un nouvel algorithme de détection de communautés dans les graphes bipartis.

CHAPITRE III

APPROCHE PROPOSÉE

Ce chapitre présente une nouvelle approche de détection de communautés dans les réseaux bipartis. L'approche proposée opère en deux phases. La première phase vise principalement à identifier un premier regroupement de nœuds par le biais d'un modèle de données transactionnelles. Spécifiquement, dans cette première phase, le graphe biparti en question est transformé en données transactionnelles, puis, un mécanisme de *scoring* qui permet de distinguer les nœuds discriminants des nœuds non discriminants, est incorporé dans un algorithme de clustering des données transactionnelles et appliqué sur les données transactionnelles obtenues de cette transformation. Comme nous allons le démontrer dans ce chapitre, cette transformation vers des données transactionnelles relate simplement les voisins de chaque nœud dans le graphe biparti en question et ne cause aucune perte d'information. La deuxième phase de notre approche est conçue spécifiquement pour raffiner le résultat de la première phase basée sur une nouvelle stratégie d'optimisation de la modularité bipartie pour identifier une structure de communautés définitive et relativement optimale. Il est à noter que l'approche proposée est applicable sur des graphes bipartis non orientés et non pondérés.

3.1 Notations et concepts

Avant de présenter notre approche, nous exposons tout d'abord quelques notations et définitions. Un graphe biparti est défini par le triplet $G = (V_1 \cup V_2, E)$, où V_1 est un ensemble de p nœuds, V_2 est un ensemble de q nœuds et E représente l'ensemble de liens dans le graphe G. Le nombre total de nœuds dans G est donné par $V = p + q$. La différence avec les réseaux classiques (homogènes) réside dans le fait que les nœuds dans V_1 et V_2 sont de différents types et les liens existent seulement entre ces deux types de nœuds (c.-à-d., entre V_1 et V_2).

L'objectif de notre approche est l'identification d'une partition de nœuds disjointe où chaque nœud $x \in (V_1 \vee V_2)$ ne peut appartenir qu'à une et une seule communauté (ou *cluster*) [1]. Nous croyons que ce genre de partitionnement permet de donner des structures de communautés qui sont d'une part, claires, et d'autre part, intuitives et facilement interprétables par l'utilisateur. Chaque communauté détectée par notre approche doit contenir des nœuds de même type qui sont densément connectés avec les nœuds de leur communauté correspondante et faiblement liés avec les nœuds des autres communautés. Rappelons que la communauté correspondante (appelée aussi le *co-cluster*) d'une communauté C_l de type V_1 est une communauté D_m de type V_2, de sorte que C_l partage le nombre maximum de liens avec D_m.

Nous utilisons C pour désigner les communautés identifiées dans V_1 et D pour désigner les communautés identifiées dans V_2. Soit $P_1 = \{ C_1, C_2, ..., C_{k_1} \}$ dénote un partitionnement de V_1 en k_1 communautés. De même, soit $P_2 = \{ D_1, D_2, ..., D_{k2} \}$ dénote un partitionnement de V_2 en k_2 communautés. Tout au long de ce mémoire,

---

1. Dans le reste de ce document, les termes communauté et *cluster* sont utilisés d'une façon interchangeable pour désigner le même concept.

nous supposons que k_1 ne doit pas forcément être égal à k_2.

Comme nous l'avons mentionné précédemment, notre approche opère en deux phases : la première phase vise à générer un premier partitionnement en s'appuyant sur le principe de clustering de données transactionnelles et la deuxième phase vise à raffiner le résultat de la première phase, en se basant sur un processus itératif qui permet de maximiser la modularité bipartie et ainsi identifier une structure de communautés relativement optimales. Les détails qui se rattachent à chaque phase sont présentés dans ce qui suit.

3.2 La première phase : le partitionnement initial

Dans cette section, nous présentons la stratégie de clustering de données transactionnelles que nous avons adoptée. Le point notable de cette démarche est que le rôle qu'un nœud joue dans la formation de communautés est considéré par le biais d'un mécanisme de *scoring*, incorporé dans une fonction objective. Avant de décrire le processus de clustering de données transactionnelles, nous devons tout d'abord procéder à la transformation du graphe biparti vers des données transactionnelles.

Un graphe biparti $G = (V_1 \cup V_2, E)$ avec $V_1 = \{ u_1, u_2,...,u_p \}$ représente l'ensemble de p nœuds de type V_1 et $V_2 = \{ v_1, v_2,...,v_q \}$ l'ensemble de q nœuds de type V_2 peut être transformé, et ce sans perte d'information, en un ensemble de données transactionnelles, de sorte que chaque transaction est définie par une liste d'items[2], et ces items relatent les voisins de chaque nœud dans le graphe biparti

2. Dans ce qui suit, les termes item et nœud sont utilisés d'une manière interchangeable pour dénoter le même concept.

sous investigation. En vue de faciliter l'analyse des résultats de cette transformation, nous divisons l'ensemble de données transactionnelles obtenu en deux ensembles de transactions (B_1 et B_2), tel que chaque ensemble relate les voisins de chaque type de nœuds (c.-à-d., V_1 et V_2).

D'une manière spécifique, la transformation du graphe biparti G en deux ensembles de données transactionnelles (B_1 et B_2) est donnée comme suit :

Le premier ensemble (B_1) est défini par :

S_{V_1}={ T_{u_1}, T_{u_2},...,T_{u_p} }

I_{V_2}={ v_1, v_2,...,v_q }

Tels que S_{V_1} dénote le sous-ensemble de p transactions qui représentent les nœuds dans V_1, et I_{V_2} le sous-ensemble de q items qui relatent les nœuds de type V_2. Chaque transaction dans S_{V_1} est définie par une liste d'items dans I_{V_2} qui relate les voisins du nœud en question.

Le deuxième ensemble (B_2) est défini par :

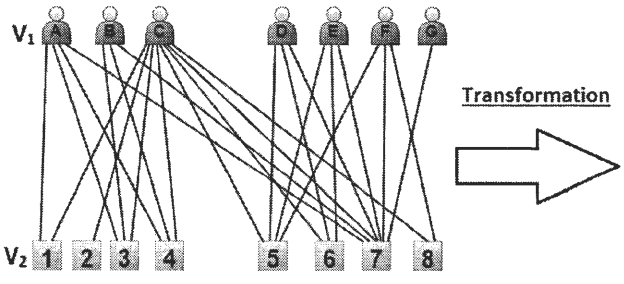
S_{V_2}={ T_{v_1}, T_{v_2},...,T_{v_q} }

I_{V_1}={ u_1, u_2,...,u_p }

Ici, S_{V_2} est le sous-ensemble de q transactions qui relatent les nœuds dans V_2 et I_{V_1} le sous-ensemble de p items qui représentent les nœuds de type V_1. Chaque transaction dans S_{V_2} est représentée par une liste d'items dans I_{V_1}, ces items dénotent les voisins du nœud en question.

Afin de faciliter la discussion, nous illustrons la transformation ci-dessus d'un graphe biparti vers des données transactionnelles, par un exemple, comme illustré

par la figure 3.1. Le graphe biparti dans la figure 3.1(a), peut être transformé en deux ensembles de données transactionnelles (B_1 et B_2) : le premier ensemble (B_1) relate les voisins des nœuds ayant le type V_1 (comme illustré par le tableau 3.1(b)) et le deuxième (B_2) relate les voisins des nœuds ayant le type V_2 (comme indiqué le tableau 3.1(c)). Chaque transaction dans ces deux tableaux (3.1(b) et 3.1(c)) est définie par une liste d'items qui représentent les voisins du nœud en question dans le graphe biparti de la figure 3.1(a). Rappelons que l'avantage de cette transformation (en données transactionnelles) sur la projection est qu'elle garde les mêmes interactions entre les nœuds et qu'il n'y a aucune perte d'information telle que la force de connexion entre les nœuds.



(a) Un graphe biparti.

L'ensemble B_1

Transactions	Items
T_A	{1, 3, 4, 7}
T_B	{3, 4, 7}
T_C	{1, 2, 3, 4, 5, 6, 7, 8}
T_D	{5, 6, 7}
T_E	{5, 6, 7}
T_F	{5, 7, 8}
T_G	{7}

(b) La transformation des nœuds de type V_1 en données transactionnelles.

L'ensemble B_2

Transactions	Items
T_1	{A, C}
T_2	{C}
T_3	{A, B, C}
T_4	{A, B, C}
T_5	{C, D, E, F}
T_6	{C, D, E}
T_7	{A, B, C, D, E, F, G}
T_8	{C, F}

(c) La transformation des nœuds de type V_2 en données transactionnelles.

Figure 3.1: Illustration de la transformation d'un graphe biparti en données transactionnelles.

Nous remarquons dans l'ensemble de données transactionnelles B_1 (le tableau 3.1(b)) que l'item 7 (le nœud 7 dans le graphe biparti) a une fréquence élevée dans l'ensemble de données, car il est apparu dans toutes les transactions. Un tel item (l'item 7) doit être considéré comme étant moins important dans la phase de clustering, parce qu'il ne nous aide pas à distinguer entre les *clusters*. Cette remarque est aussi applicable sur l'item C dans l'ensemble B_2 (le tableau 3.1(c)). D'autre part, nous voyons également dans le tableau 3.1(b) que l'item 2 apparaît juste une seule fois dans l'ensemble B_1. En effet, cet item ne contribue pas à la formation d'aucun *cluster*, parcequ'il a une faible occurrence dans l'ensemble de données transactionnelles B_1 et à cet effet, il n'aide pas à distinguer entre les communautés. Il convient de rappeler que les algorithmes de détection de communautés dans les graphes bipartis déjà existants ne prennent pas en compte ce genre de nœuds (les nœuds non discriminants) dans la phase de clustering, et ils les traitent d'une manière similaire avec les autres nœuds du graphe. Afin de tenir compte de la discrimination des nœuds dans le processus de clustering et d'identifier des communautés en nous basant sur la contribution des nœuds dans le graphe biparti, nous appliquons l'algorithme de clustering des données transactionnelles (présenté dans (Bouguessa, 2011)) sur les résultats obtenus de la transformation du graphe biparti en données transactionnelles (l'ensemble B_1 et B_2). Les détails de cet algorithme sont présentés dans ce qui suit.

L'objectif principal de l'algorithme de clustering de données transactionnelles présenté dans (Bouguessa, 2011) est de diviser l'ensemble de transactions S (ici S peut être soit S_{V_1} ou S_{V_2}), en nous basant sur la contribution des items, en un ensemble de clusters $C_T = \{ C_{T_1}, C_{T_2},...,C_{T_k} \}$ où k dénote le nombre de clusters dans C_T, et $\forall(C_{T_g}, C_{T_h}) \in C_T : C_{T_g} \cap C_{T_h} = \emptyset$, dans le cas où $g \neq h$. Les transactions groupées dans C_{T_j} (avec j = 1,..., k) doivent avoir un degré élevé

d'homogénéité par rapport à toutes les transactions dans (C_T - C_{T_j}).

La recherche d'un partitionnement $C_T = \{ C_{T_1}, C_{T_2},...,C_{T_k} \}$ dans l'ensemble de données transactionnelles S est guidée par la fonction objective suivante :

$$F(C_T) = \sum_{j=1}^{k} \left[\frac{r_j}{r} . Quality(C_{T_j}) \right] \tag{3.1}$$

Ici, r dénote le nombre de transactions dans S ($r = p \vee r = q$) et r_j le nombre de transactions dans C_{T_j}, le terme $\frac{r_j}{r}$ représente la contribution relative du cluster C_{T_j} à la partition C_T et $Quality(C_{T_j})$ mesure la qualité du cluster C_{T_j} basé sur la similarité entre les items des transactions qui appartiennent à C_{T_j}. D'une manière spécifique, $Quality(C_{T_j})$ est donnée par :

$$Quality(C_{T_j}) = \frac{1}{r_j} \sum_{m \in C_{T_j}} ntr(m, C_{T_j}) \cdot w(m, C_{T_j}) \cdot w(m, S) \tag{3.2}$$

Le terme $ntr(m, C_{T_j})$ dénote le nombre de transactions dans C_{T_j} contenant l'item m. Le poids $w(m, C_{T_j})$ mesure le degré de participation de l'item m dans C_{T_j}, $w(m, S)$ mesure l'importance de l'item m dans l'ensemble de données transactionnelles S. En effet, l'objectif principal d'utiliser ces deux poids ($w(m, C_{T_j})$ et $w(m, S)$) consiste à éviter de traiter d'une manière similaire les items dans C_{T_j} en fournissant plus de puissance discriminative pour les items discriminants qui nous aident à distinguer entre les clusters, et même de diminuer l'importance des items non discriminants, ceux qui ne contribuent pas à la formation des communautés.

Le poids $w(m, C_{T_j})$ est défini par :

$$w(m, C_{T_j}) = \frac{ntr(m, C_{T_j})}{r_j} \cdot \frac{ntr(m, C_{T_j})}{ntr(m, C_{T_j}) + ntr(m, (S - C_{T_j}))} \tag{3.3}$$

Dans l'équation (3.3), $ntr(m, (S - C_{T_j}))$ dénote le nombre de transactions qui n'appartiennent pas à C_{T_j} mais elles contiennent l'item m.

Le poids $w(m, S)$ est défini comme suit :

$$w(m, S) = \frac{ntr(m, S) \cdot (r - ntr(m, S) + 1)}{r} \tag{3.4}$$

Dans l'équation (3.4), $ntr(m, S)$ représente le nombre de transactions dans S qui contient l'item m. La valeur de $w(m, S)$ est fixée durant le processus de clustering, car elle mesure l'importance de l'item m dans l'ensemble de données transactionnelles S.

L'objectif derrière la multiplication de ces deux poids ($w(m,C_{Tj})$ et $w(m, S)$) dans la fonction $Quality(C_{Tj})$ (donnée par l'équation (3.2)) est de renforcer le poids des items qui contribuent à la formation des clusters, en leur donnant une puissance discriminante élevée, et même de diminuer le poids des items non discriminants, ceux qui ne nous n'aident pas à distinguer entre les communautés, en les rendant négligeables durant le processus de clustering. En effet, la maximisation de l'équation (3.2) par ce mécanisme de *scoring* indique que les transactions dans C_{Tj} possèdent une plus grande similarité entre elles qu'ailleurs (dans S - C_{Tj}), ce qui implique à ce moment que la maximisation de la fonction $F(C_T)$ (donnée par l'équation (3.1)) indique que le partitionnement C_T a une bonne qualité. La fonction objective $F(C_T)$ peut être maximisée grâce à la procédure itérative suivante (algorithme 1) :

Algorithme 1 : Phase 1

Entrée : Une collection de données transactionnelles S ;

Résultat : $C_T = \{ C_{T_1}, C_{T_2},...,C_{T_k} \}$: le partitionnement de S en k *clusters* ;

début

1- Allouer pour chaque transaction T_j $(j = 1...r)$ dans S un cluster existant ou un nouveau cluster qui maximise $F(C_T)$;

2- *transaction_déplacée* $= faux$;

3- **pour** $T_j = 1$ **à** $r \in S$ **faire**

Déplacer T_j à un *cluster* existant ou un nouveau *cluster* pour maximiser $F(C_T)$;

si T_j *est déplacée* **alors**

transaction_déplacée $= vrai$;

fin

fin

3- **si** *transaction_déplacée* $== faux$ **alors**

allez à l'étape 5 ;

sinon

allez à l'étape 2 ;

fin

4- Retourner $C_T = \{ C_{T_1}, C_{T_2},...,C_{T_k}\}$;

fin

Au début de l'algorithme 1 (ligne 1), les transactions sont investiguées séquentiellement et en assignant chacune à un cluster déjà existant ou un nouveau cluster qui maximise la fonction objective $F(C_T)$ (l'équation (3.1)). En effet, l'objectif principal de cette étape est de construire, en premier lieu, une partition initiale C_T basée sur la fonction objectif $F(C_T)$. Par la suite, l'algorithme passe à une boucle itérative (ligne 3) pour améliorer les résultats obtenus précédemment (la partition initiale). De ce fait, l'affectation des clusters est raffinée de manière itérative jusqu'à ce qu'il ne soit plus possible d'augmenter la valeur de la fonction $F(C_T)$. Ce processus de raffinage sera terminé si aucune transaction dans S n'est

déplacée d'un cluster à un autre durant une itération. Autrement, une nouvelle passe est commencée.

Il est à noter que la maximisation de la fonction $F(C_T)$ via l'algorithme 1 est orientée vers un maximum local de $F(C_T)$. De ce fait, nous élaborons dans la phase 2, de notre approche, une procédure d'optimisation de la modularité bipartie *murata+* (présentée initialement dans (Pesantez et Kalyanaraman, 2017)) qui permet de fusionner les clusters obtenus de la phase 1 afin de maximiser la modularité et identifier une structure de communautés relativement optimale. Les détails de la phase 2 sont présentés dans ce qui suit.

3.3 La deuxième phase : raffinement du clustering

Avant de procéder à la phase 2, le graphe biparti en question doit être divisé en communautés selon le partitionnement obtenu dans la phase 1. Puis, nous appliquons notre stratégie d'optimisation de la modularité bipartie (*murata+*) sur le partitionnement de la phase 1. Il convient de rappeler ici que la modularité (*murata+*) (présentée par Pesantez et Kalyanaraman (2017)) est définie comme suit :

Supposons que C_l est une communauté de type V_1 et que D_m représente sa communauté correspondante dans V_2 et vice versa, e_{lm} la fraction de tous les liens qui connectent les nœuds de C_l avec les nœuds de D_m, a_l et a_m la fraction de tous les liens dans C_l et D_m, respectivement. Alors la modularité *murata+* est définie par :

$$Q_M^+ = \sum_C (e_{lm} - a_l a_m) + \sum_D (e_{ml} - a_l a_m) \tag{3.5}$$

Chaque communauté C_l choisit sa communauté correspondante D_m en se basant sur la fonction suivante :

$$C_l = \arg\max_m(e_{ml} - a_l a_m) \tag{3.6}$$

De même, chaque communauté D_m choisit sa communauté correspondante C_l en se basant sur la fonction suivante :

$$D_m = \arg\max_l(e_{lm} - a_l a_m) \tag{3.7}$$

En effet, l'avantage de choisir la modularité *murata+* (Pesantez et Kalyanaraman, 2017) réside dans les points suivants :

1. La modularité *murata+* est apte à détecter des communautés avec des correspondances d'un à plusieurs (c.-à-d., une communauté de type V_1 peut être connectée avec une ou plusieurs communautés de type V_2 ou vice versa). Cela signifie que les deux types de nœuds peuvent avoir un nombre de communautés différent.

2. La modularité *murata+* fournit un degré de correspondance entre chaque paire de communautés de différent type. Ce degré peut être utilisé pour analyser les caractéristiques des communautés dans le graphe biparti sous investigation.

3. La modularité *murata+* fournit des correspondances symétriques, c.-à-d., si une communauté C_l a choisi la communauté D_m comme son *co-cluster*, D_m doit forcément choisi C_l comme son *co-cluster*.

Rappelons que l'objectif principal de la phase 2 est de raffiner les résultats obtenus de la phase 1 et même d'identifier une structure de communautés relativement optimale. À cet effet, nous proposons une heuristique d'optimisation de

la modularité bipartie (*murata+*). Cette heuristique se base principalement sur la fusion de chaque paire de communautés de même type afin de maximiser la modularité bipartie. Pour une partition initiale de graphe en (k_1+k_2) communautés, nous pouvons avoir $\left(k_1*(k_1 - 1)/2 + k_2*(k_2 - 1)/2\right)$ combinaisons possibles. Afin de rendre notre algorithme plus rapide et ainsi maximiser la modularité, nous fusionnons juste les paires de communautés qui partagent au moins un nœud commun, car la fusion de deux communautés de même type qui ne partagent pas des nœuds communs entre elles ne va pas augmenter la modularité. Donc plusieurs combinaisons peuvent être évitées.

L'algorithme de la phase 2 (présenté ci-dessous) commence tout d'abord par calculer, pour chaque communauté, une liste de communautés candidates (ayant le même type). Supposons que $D_{V2} \in V_2$ dénote l'ensemble de toutes les communautés voisines de la communauté $C_l \in V_1$. Alors les communautés candidates de C_l sont toutes les communautés dans V_1 qui partagent au moins un nœud en commun avec D_{V2}. Par la suite, pour chaque combinaison possible, entre la communauté en question et l'une de ces communautés candidates, on estime la modularité bipartie du graphe et on choisit celle qui a la plus grande valeur de la modularité. Cette dernière combinaison est acceptée et la fusion de deux communautés aura lieu si la valeur de la modularité bipartie de cette fusion est supérieure à la valeur de la précédente. Ce processus itératif de fusion s'arrête si aucune combinaison possible ne peut augmenter la valeur de la modularité du graphe biparti. L'algorithme 2 résume la phase 2 (raffinement du résultat de la phase 1) de notre approche.

Algorithme 2 : phase 2

Entrée : $G = (V_1 \cup V_2, E)$: un graphe biparti ;

N : une liste de taille $|V_1 + V_2|$ contenant le partitionnement initial de la phase 1 où chaque case de cette liste indique l'appartenance d'un nœud à un cluster.

Résultat : $P_1 = \{C_1, C_2,...,C_{k_1}\}$: les communautés dans V_1.

$P_2 = \{ D_1, D_2,...,D_{k2} \}$: les communautés dans V_2.

Variables : $cluster_fus$: le cluster candidat qui maximise la modularité Q_M^+ ;

début

Deviser G en communautés selon N ;

C_N : une liste contenant l'index de chaque cluster dans N ;

Q_M^{+first} = calculer Q_M^+ pour G selon l'équation (3.5) ;

répéter

pour chaque cluster R dans C_N **faire**

$candidate_cluster$ = calculer les clusters candidats du cluster R ;

pour chaque cluster candidat $Candi_R$ dans $candidate_cluster$ **faire**

Q_M^{+new} = calculer Q_M^+ selons l'équation (3.5) pour que R et $Candi_R$ soient dans le même cluster ;

si $Q_M^{+new} > Q_M^{+first}$ **alors**

$Q_M^{+first} = Q_M^{+new}$;

$cluster_fus = Candi_R$;

fin

fin

si $Q_B^{+first} > Q_B^{+new}$ **alors**

fusionner R et $cluster_fus$;

fin

fin

jusqu'à ce qu'il ne soit plus possible d'augmenter la modularité Q_M^{+first} ;

Retourner P_1, P_2 ;

fin

3.4 Conclusion

Dans ce chapitre, nous avons présenté une nouvelle approche de détection de communautés dans les graphes bipartis. Cette approche est basée principalement sur deux phases, la première vise à regrouper les nœuds en communautés selon un mécanisme de pondération qui permet de distinguer les nœuds discriminants des nœuds non discriminants. La deuxième phase vise à combiner les résultats de la première phase à l'aide d'une nouvelle heuristique d'optimisation de la modularité bipartie afin de trouver une structure de communautés optimale. Une évaluation empirique, sur des graphes synthétiques et réels, de l'approche proposée est présentée dans le chapitre suivant.

CHAPITRE IV

ÉVALUATION DE L'APPROCHE PROPOSÉE

Ce chapitre présente une évaluation empirique détaillée de l'approche proposée sur des graphes bipartis synthétiques et réels. La performance de l'approche est comparée à cinq algorithmes de détection de communautés dans les graphes bipartis, à savoir *Adaptive Brim, LPBrim, LPAb+, BiSBM et BiLouvain.* Dans ce qui suit, nous commençons tout d'abord par présenter le cadre expérimental que nous avons adopté, puis nous décrivons les critères d'évaluation sélectionnés. Par la suite, nous présentons les résultats obtenus de notre évaluation.

4.1 Cadre expérimental

4.1.1 Algorithmes sélectionnés pour la comparaison

Afin d'illustrer la performance de la méthode proposée, cinq approches ont été sélectionnées pour la comparaison, telles que *LPAb+*[1], *LPBrim*[2], *Adaptive*

1. l'implémentation de l'algorithme *LPAb+* est disponible dans https ://github.com/sjbeckett/weighted-modularity-LPAwbPLUS.

2. l'implémentation de l'algorithme *LPBrim* est disponible dans https ://bimat.github.io/

BRIM[3], *BiLouvain*[4] *et BiSBM*[5]. Le choix de ces algorithmes est dicté par le fait qu'ils représentent, d'une part, les différents types d'algorithmes de détection de communautés dans les graphes bipartis (les approches fondées sur les graphes homogènes et les approches fondées sur le modèle de liaison similaire), et d'autre part, ils sont très connus dans la littérature et considérés comme étant les plus performants dans les graphes bipartis. Comme nous avons mentionné dans le chapitre 2, l'algorithme *BiSBM* nécessite que le nombre de clusters (ou communautés) doive être spécifié à l'avance par l'utilisateur. De ce fait, pour les graphes synthétiques, le nombre réel de communautés a été fourni pour cet algorithme. Par contre, nous n'avons pas considéré cet algorithme (*BiSBM*) pour les expérimentations sur les graphes réels, puisqu'il est impossible de spécifier le nombre de clusters pour les graphes réels (sauf le cas des graphes réels benchmarks qui ont une structure de communautés connue).

Comme indiqué précédemment, les approches fondées sur les graphes homogènes, à savoir, *LPAb+, LPBrim et Adaptive BRIM* donnent des communautés contenant les deux types de nœuds. Afin de faciliter la comparaison des résultats, nous avons divisé chaque communauté détectée par ces algorithmes en deux communautés de sorte que chaque communauté contient des nœuds de même type. Il convient de rappeler aussi que nous n'avons pas considéré la projection dans cette expérimentation, car elle souffre de la perte d'information.

---

3. l'implémentation de l'algorithme *Adaptive BRIM* est disponible dans https ://bimat.github.io/

4. L'implémentation de l'algorithme *BiLouvain* est disponible dans https ://github.com/paolapesantez/biLouvain

5. L'implémentation de l'algorithme *BiSBM* est disponible dans http ://danlarremore.com/bipartiteSBM

4.1.2 Critères d'évaluation

Afin d'évaluer la performance des approches sélectionnées pour la comparaison, nous avons considéré des critères d'évaluation internes et externes. Les critères internes sont utilisés lorsque la partition est connue à l'avance, c.-à-d., lorsque nous connaissons l'appartenance de chaque nœud à sa communauté (*ground truth*). Tandis que les critères externes sont utilisés quand nous n'avons aucune connaissance *à priori* sur le partitionnement du graphe sous investigation. Autrement dit, il n'y a pas une partition prédéfinie. Dans nos expérimentations, un critère d'évaluation interne est utilisé lorsque le *ground truth* est disponible (en particulier avec les graphes synthétiques et les graphes réels benchmarks). En l'absence du *ground truth* (ce qui est généralement le cas des données réelles), nous avons utilisé deux critères d'évaluation externes.

4.1.2.1 Critère interne : le NMI

Comme nous l'avons souligné, pour chaque réseau synthétique généré, notre approche proposée ainsi que les autres approches choisies pour la comparaison sont appliquées. Par la suite, les structures de communautés identifiées dans chaque exécution sont comparées avec celles prédéfinies (*ground truth*) par le générateur de réseaux que nous avons utilisé. Pour établir cette comparaison, entre les communautés détectées et les communautés prédéfinies, nous avons sélectionné le NMI (l'information mutuelle normalisée), car il est le plus utilisé dans la littérature. L'objectif de ce dernier consiste à mesurer la similarité entre deux partitions, à savoir, la partition détectée et la partition prédéfinie.

Pour deux partitions A et B, avec A qui représente la partition prédéfinie et B qui représente la partition dégagée du même réseau, n le nombre total de nœuds,

k_A et k_B représentent le nombre de clusters dans A et B, respectivement, le NMI est défini comme suit :

$$NMI(A,B) = \frac{-2\sum_{i=1}^{k_A}\sum_{j=1}^{k_B} N_{ij}\log(\frac{N_{ij}n}{N_iN_j})}{\sum_{i=1}^{k_A} N_i\log(\frac{N_i}{n}) + \sum_{j=1}^{k_B} N_j\log(\frac{N_j}{n})} \tag{4.1}$$

Dans l'équation (4.1), N représente la matrice de confusion avec N_{ij} qui indique le nombre de nœuds qui apparaissent à l'i^{eme} cluster de la partition A et le j^{eme} cluster de la partition B. N_i est le nombre de nœuds dans le cluster i de la partition A et N_j représente le nombre de nœuds dans le cluster j de la partition B.

Le NMI retourne généralement une valeur comprise entre 0 et 1. Si cette valeur est proche de 1, la partition est proche de la partition prédéfinie. Ce qui suggère une bonne performance de l'algorithme de détection en question. Une valeur de NMI proche de 0 suggère que l'algorithme de détection de communautés a échoué dans le fait d'identifier une structure de communautés proche du *ground truth.*

4.1.2.2 Critères externes

Contrairement aux critères internes qui sont utilisés sur des graphes où la partition prédéfinie doit être connue, les critères externes sont souvent utilisés dans le cas où la partition prédéfinie est manquante, autrement dit, dans le cas où nous n'avons aucune connaissance *à priori* sur l'appartenance des nœuds à leurs communautés. Afin de réaliser une expérimentation objective qui relate les points forts et faibles des algorithmes adoptés pour cette expérimentation, nous avons utilisé deux critères externes, à savoir, la *couverture* (*Coverage* en anglais) proposée par Brandes et Erlebach (2005) et la *densité de modularité* (*modularity density* en anglais) proposée par Xu et Chen (2014).

Soit $G = (V_1 \cup V_2, E)$ un graphe biparti et $P = \{ C_1, C_2,...,C_k\}$ un partitionnement de G en k communautés. En effet, l'objectif principal des critères externes est de donner un score qui relate la qualité du partitionnement P obtenu par l'algorithme en question sur le graphe biparti sous investigation.

Comme mentionnée dans (Brandes et Erlebach, 2005), la *couverture* mesure la densité interne au sein des communautés. Elle est définie par la fraction de tous les liens intra communautés par rapport au nombre total de liens dans le graphe biparti sous investigation. Les valeurs de cette mesure sont généralement comprises entre 0 et 1. Une valeur plus proche de 1 indique que les nœuds dans chaque cluster sont densément connectés entre eux, ce qui est considéré comme un bon partitionnement, tandis qu'une valeur proche de 0 désigne que les nœuds au sein des clusters sont faiblement connectés entre eux. D'une manière générale, la *couverture* du graphe G est définie par l'équation suivante :

$$Couverture(G) = \frac{\sum_{i=1}^{k} e_{C_i}}{E} \tag{4.2}$$

Dans l'équation (4.2), e_{C_i} dénote le nombre de liens intra communautés du cluster C_i et E le nombre total de liens dans G.

La deuxième mesure que nous avons adoptée pour cette expérimentation est la *densité de la modularité*, proposée par Xu et Chen (2014). Contrairement à la *couverture* qui prend en compte seulement la densité interne au sein des communautés, la *densité de la modularité* prend en considération à la fois la densité interne au sein des communautés, de même que la densité externe entre les communautés du graphe biparti. Cette métrique prend des valeurs élevées si les nœuds dans chaque communauté sont densément connectés entre eux et faiblement reliés

avec les nœuds d'autres communautés. Dans le cas contraire, elle prend des valeurs faibles. Plus spécifiquement, la *densité de la modularité* est définie par l'équation suivante :

$$M_D = \sum_{i=1}^{k} M_D(C_i) \tag{4.3}$$

$$M_D(C_i) = D_{in}(C_i) - D_{out}(C_i) \tag{4.4}$$

Dans l'équation (4.4), $D_{in}(C_i)$ représente la fraction de tous les liens intra communautés du cluster C_i par rapport au nombre total de nœuds dans G et $D_{out}(C_i)$ dénote la fraction de tous les liens inter communautés du cluster C_i par rapport au nombre total de nœuds dans G. De ce fait, $D_{in}(C_i)$ et $D_{out}(C_i)$ sont définies par les deux équations suivantes :

$$D_{in}(C_i) = \frac{e_{C_i}}{\mid V_1 \mid + \mid V_2 \mid} \quad , \quad D_{out}(C_i) = \frac{e_{C_i out}}{\mid V_1 \mid + \mid V_2 \mid} \tag{4.5}$$

Dans l'équation (4.5), e_{C_i} est le nombre de liens intra communautés du cluster C_i, et $e_{C_i out}$ le nombre de liens inter communautés du cluster C_i. $\mid V_1 \mid$ et $\mid V_2 \mid$ représentent le nombre de nœuds dans V_1 et V_2, respectivement.

4.2 Expérimentations sur les graphes synthétiques

Le principal aspect étudié dans ce type de graphes (les graphes synthétiques) est la performance de notre approche proposée en termes de capacité à découvrir la partition prédéfinie (où l'on connaît à l'avance l'appartenance de chaque nœud à sa communauté dans le graphe biparti sous investigation). Autrement dit,

nous évaluons la performance de notre approche ainsi que les cinq algorithmes sélectionnés pour la comparaison sur des graphes synthétiques générés avec une variété de densités de liens et une variété de nœuds non discriminants (les nœuds massivement connectés avec les autres nœuds du graphe et les nœuds avec des connexions éparses).

4.2.1 Génération des graphes bipartis synthétiques

Le modèle de génération de graphes bipartis synthétiques que nous avons utilisé représente une extension de celui de graphes standards (homogènes), défini initialement par Newman et Girvan (2004). Avant de procéder à la description de ce modèle, il convient de rappeler qu'un graphe biparti est défini par le triplet $G = (V_1 \cup V_2, E)$, tel que V_1 représente l'ensemble de p nœuds, V_2 l'ensemble de q nœuds et E l'ensemble de liens dans G. Les nœuds dans V_1 et V_2 sont de différents types et les liens existent seulement entre ces deux types de nœuds. Le modèle de génération de graphes bipartis que nous avons adopté utilise plusieurs paramètres, tels que, p et q qui dénotent le nombre de nœuds dans V_1 et V_2, respectivement, P_{in}, qui désigne la probabilité d'avoir un lien entre deux nœuds de différents types qui appartiennent à des communautés ayant le même *co-cluster* (ou communauté correspondante), P_{out}, qui représente la probabilité d'avoir un lien entre deux nœuds de différents types qui appartiennent à des communautés ayant un différent *co-cluster* ; k_1, k_2 dénotent le nombre de clusters dans V_1, V_2, respectivement. Afin de relater la présence des nœuds non discriminants (dans ce cas, nous faisons référence aux nœuds avec des connexions éparses), nous avons ajouté un nouveau paramètre appelé n_{nd} qui représente le pourcentage des nœuds non discriminants, c.-à-d., les nœuds qui ne contribuent pas à la formation des communautés dans V_1 et V_2.

Il convient également de noter que le modèle de génération de graphes se base également sur un paramètre important qui est le paramètre de mixage *mxp* (*mixing parameter* en anglais). L'objectif de ce dernier est de contrôler l'intensité de la structure de communautés pour qu'elle soit compatible avec des propriétés topologiques réalistes. Ce paramètre (*mxp*) représente la moyenne (pour tous les nœuds du graphe biparti) du rapport entre le nombre de voisins externes de chaque nœud (les voisins du nœud qui se trouvent à l'extérieur de son *co-cluser*) et son degré total (le nombre de tous les voisins du nœud en question). Une faible valeur de ce paramètre (*mxp*) indique que les communautés sont bien séparées (cf. figure 4.1(a) et 4.1(b)). Toutefois, avec une valeur élevée de *mxp*, les communautés se chevauchent de plus en plus et la structure de communautés sera difficilement identifiable (cf. figure 4.1(d)) à cause du grand nombre de nœuds non discriminants (dans ce cas, nous faisons référence aux nœuds massivement connectés avec les autres nœuds du graphe).

En regard des valeurs de paramètres fournis, les communautés sont générées en deux étapes :

1. En premier lieu, les nœuds non discriminants (avec des connexions éparses) sont ajoutés à chaque matrice bi-adjacence selon la valeur n_{nd}. Rappelons que la matrice bi-adjacence d'un graphe biparti dispose d'un bloc hors diagonale sous forme du bloc-matrice (cf. figure 2.6) et que les nœuds de même type ne sont jamais connectés. Puis, chaque matrice est divisée en (k_1+k_2) blocs de taille variable, de sorte que chaque bloc correspond à une communauté associée à un type de nœuds (soit V_1 ou V_2). Par la suite, les connexions internes (entre les nœuds de deux communautés correspondantes) sont attribuées aléatoirement selon la valeur de P_{in}. Notons ici que les nœuds non discriminants (avec des connexions éparses) sont affectés à leurs communau-

tés en se basant sur le nombre de connexions partagées avec leurs *co-clusters*.

2. Après que les nœuds non discriminants sont ajoutés et que les connexions internes (les liens intra communautés) sont attribuées, les connexions inter communautés (celles entre les noeuds de différentes communautés correspondantes ou *co-cluster*) sont ajoutées d'une manière aléatoire en se basant sur le paramètre de mixage mxp. Il convient de noter ici qu'une valeur élevée du paramètre de mixage mxp implique la présence des nœuds non discriminants (les nœuds massivement connectés).

À titre illustratif, la figure 4.1 montre 6 matrices bi-adjacences de 6 réseaux synthétiques construits par notre générateur décrit ci-dessus. Chaque matrice représente un graphe biparti synthétique qui est caractérisé par le paramètre de mixage mxp et le pourcentage des nœuds non discriminants n_{nd}. De ce fait, les lignes et les colonnes représentent les deux types de nœuds du graphe biparti en question, chaque point bleu dans la matrice représente un lien entre deux nœuds de différents types et les points blancs relatent l'absence de liens entre les nœuds dans le graphe biparti.

Chaque matrice dans la figure 4.1 illustre 5 blocs et chaque bloc représente une communauté (de différentes tailles) dans un type de nœuds. Prenons la figure 4.1(a) comme exemple, nous remarquons que les communautés dans cette matrice sont bien séparées et donc, faciles à identifier, car il n'y a pas de chevauchement entre elles ($mxp = 0$, $n_{nd} = 0$).

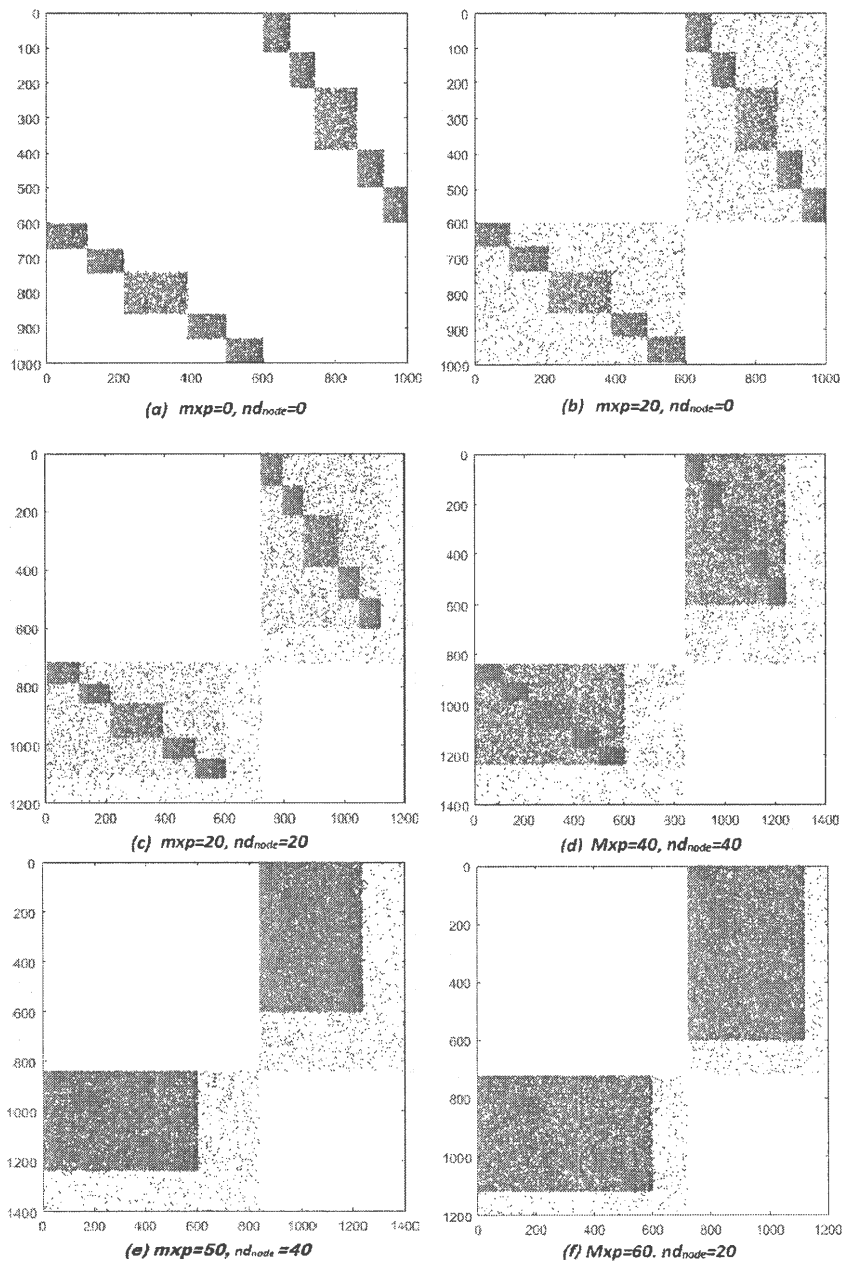


Figure 4.1: Les matrices bi-adjacences de 6 réseaux bipartis synthétiques avec différents paramètres d'entrées.

Un autre exemple plus complexe (même extrême) est celui de la figure 4.1(f), les

communautés ici ne sont pas visibles, car le paramètre de mixage est très élevé ($mxp = 0.60$). Ce dernier augmente le chevauchement entre les communautés, ce qui lui permet à son tour d'augmenter le nombre de nœuds non discriminants (dans ce cas nous faisons référence aux nœuds massivement connectés) et de rendre les communautés très difficilement identifiables. La région blanche, faiblement granulée dans cette matrice illustre la présence des nœuds non discriminants (ceux avec des connexions éparses), puisque $n_{nd} = 20$.

4.2.2 Résultats et discussions

Pour que les résultats de cette comparaison soient fiables et pertinents, nous avons généré aléatoirement pour chaque même ensemble de paramètres d'entrées 10 réseaux synthétiques, puis nous avons appliqué les algorithmes mentionnés ci-dessus sur chaque graphe généré. Notons ici que certaines approches, à savoir, *LPAb+, AdaBrim, LPBrim et BiSBM* donnent différents résultats dans chaque exécution. Par conséquent, nous exécutions chacune de ces approches trois fois sur le même graphe et nous prenions la moyenne des résultats. En outre, le nombre de clusters prédéfini (5 clusters dans chaque type de nœuds) est considéré comme un paramètre d'entrée pour l'algorithme *BiSBM*.

Les valeurs des paramètres d'entrées que nous avons considérées pour ces expérimentations sont : $p = 600$ nœuds de type V_1, $q = 400$ nœuds de type V_2, $P_{in} = 0.20$, P_{out} [0, 0.05], le paramètre de mixage [0, 0.4], le pourcentage de nœuds non discriminants n_{nd} est choisi à partir de [0, 0.4] et le nombre de clusters (k_1 et k_2) a été fixé à 5 clusters dans chaque type de nœuds ($k_1 = 5$, $k_2 = 5$), donnant un nombre total de 10 clusters dans chaque graphe généré. La taille des communautés varie entre 6 et 20 pour cent du nombre total de nœuds n du graphe. Les résultats de cette expérimentation sont présentés ci-dessous.

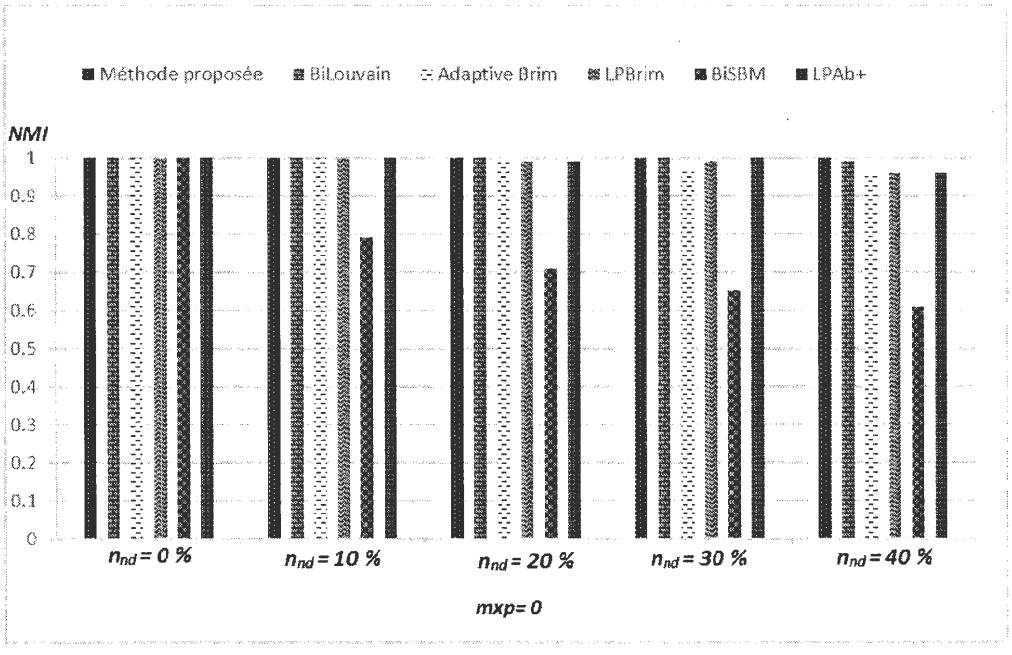


Figure 4.2: Résultats obtenus sur des graphes synthétiques avec le paramètre de mixage *mxp* égal à 0 et différents pourcentages de nœuds non discriminants (n_{nd}).

La figure 4.2 illustre les résultats des expérimentations sur des graphes synthétiques avec un paramètre de mixage ($mxp = 0$) et différents pourcentages de nœuds non discriminants n_{nd} (allant de 0 % jusqu'à 40 %). Comme montré dans cette figure, la plupart des algorithmes arrivent à identifier les structures de communautés réelles (prédéfinies par le générateur). Cette bonne performance est due au fait que les structures de communautés prédéfinies sont bien séparées et il n'y a pas de chevauchement entre elles, car le paramètre de mixage est égal à 0 ($mxp = 0$). Toutefois, nous remarquons que la performance de l'algorithme *BiSBM* se dégrade quand le pourcentage de nœuds non discriminants augmente ($n_{nd} >= 10$ %). Nous pouvons dire dans ce cas que l'algorithme *BiSBM* trouve des difficultés en présence des nœuds non discriminants, ce qui peut être expliqué par le fait que le déplacement des nœuds non discriminants vers leurs communautés ne maximise

pas la fonction de probabilité utilisée par cet algorithme. En outre, comme mentionné précédemment, *BiSBM* est un algorithme non déterministe et, à chaque exécution, il nécessite de spécifier le nombre de clusters dans les deux types de nœuds. À noter que cette dégradation de performance est aussi claire dans les deux autres figures (4.3 et 4.4).

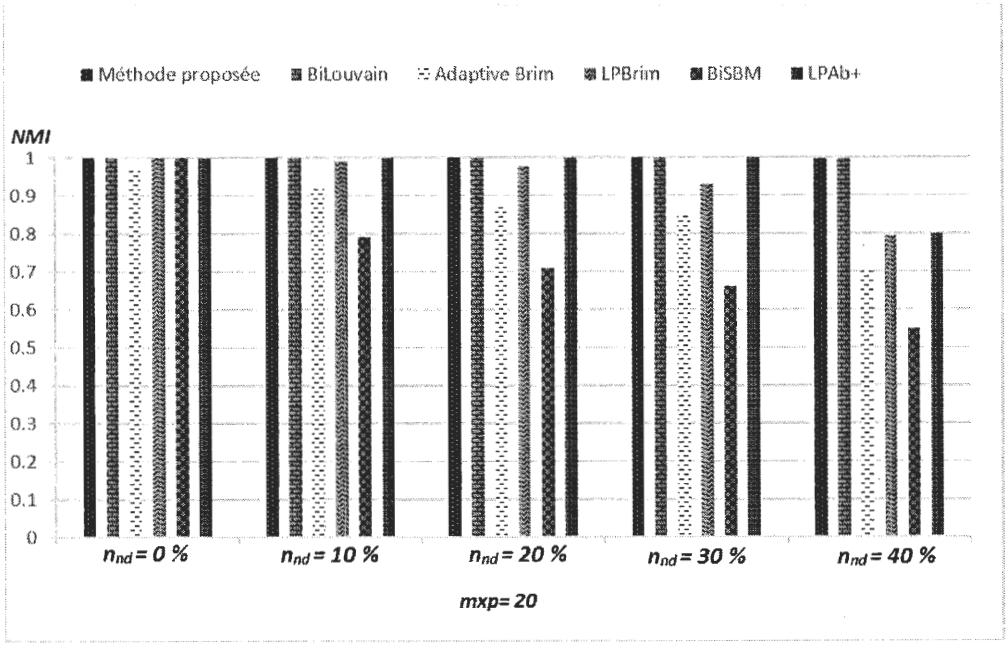


Figure 4.3: Résultats obtenus sur des graphes synthétiques avec le paramètre de mixage *mxp* égal à 20 et différents pourcentages de nœuds non discriminants (n_{nd}).

L'augmentation de la valeur du paramètre de mixage (mxp) à 20 et l'accroissement du pourcentage de nœuds non discriminants ($n_{nd} >= 10\ \%$) dans la figure 4.3 ont influencé d'une manière directe la précision de l'algorithme *Adaptive Brim*. Cela peut être causé par le mécanisme utilisé dans cet algorithme qui se base principalement sur la réaffectation aléatoire des nœuds aux différentes communautés afin de chercher le nombre de clusters approprié qui maximise la modularité bipartie (définie par l'équation (2.1)). Nous remarquons aussi, dans la même figure

(figure 4.3), que la qualité des résultats de l'algorithme *LPBrim* se dégrade au fur et à mesure que le pourcentage des nœuds non discriminants augmente (n_{nd} >= 30 %). Par exemple, avec un paramètre de mixage $mxp = 20$ et un pourcentage de nœuds non discriminants égal à 40, la valeur de *NMI* pour l'algorithme *LPBrim* est égale à 0.79. Toutefois, comme nous l'avons mentionné précédemment, l'algorithme *LPBrim* arrive à bien améliorer les résultats de l'algorithme *Adaptive Brim.*

En revanche, comme illustrée dans les deux figures (4.2 et 4.3), notre approche proposée dans ce mémoire donne toujours la plus grande valeur de *NMI*, peu importe le pourcentage de nœuds non discriminants (n_{nd}) et de paramètres de mixage (mxp). Cette bonne performance est due au fait que l'approche proposée se base sur un mécanisme de *scoring* qui est incorporé dans une fonction objective pour lui permettre de distinguer les nœuds discriminants des nœuds non discriminants dans le processus de clustering. D'une manière spécifique, ce mécanisme de *scoring* attribue un poids faible aux nœuds non discriminants (les nœuds massivement connectés et les nœuds avec des connexions éparses) et un poids élevé aux nœuds discriminants (ceux qui contribuent à la formation des communautés) au cours du processus de clustering. De ce fait, l'identification des communautés dans la première phase de notre approche se base principalement sur les nœuds les plus importants pour le processus de clustering (c.-à-d., les nœuds discriminants). Par la suite, dans la deuxième phase, un processus itératif est lancé afin de maximiser la modularité bipartie (présentée par l'équation (3.5)) et de raffiner les résultats de la première phase.

Nous remarquons également dans les deux figures (4.2 et 4.3) que l'algorithme *BiLouvain* donne des résultats compétitifs avec notre approche proposée. Ceci est dû au fait que cet algorithme (*BiLouvain*) utilise un processus itératif qui se base d'une part, sur le gain de la modularité bipartie résultant du processus de fusion des nœuds du même type et d'autre part, sur l'optimisation de la modularité globale du graphe biparti sous investigation. Ce mécanisme d'optimisation permet à *BiLouvain* de trouver une structure de communautés relativement optimale et qui maximise la modularité bipartie (l'équation (3.5)). Toutefois, cette bonne performance se dégrade avec l'augmentation du paramètre de mixage à 40 et l'augmentation du pourcentage de nœuds non discriminants ($n_{nd} >= 20$ %) dans la figure 4.4.

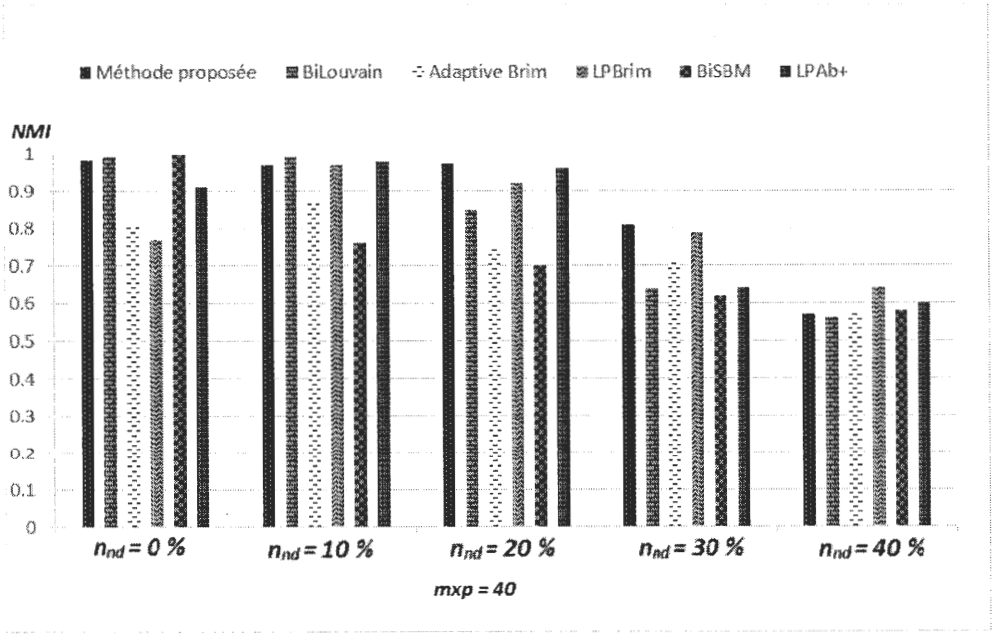


Figure 4.4: Résultats obtenus sur des graphes synthétiques avec le paramètre de mixage *mxp* égal à 40 et différents pourcentages de nœuds non discriminants (n_{nd}).

En augmentant davantage le paramètre de mixage (mxp) à 40 dans la figure 4.4, nous remarquons que la qualité de partitionnement de plusieurs algorithmes a été dégradée. À titre d'exemple, avec un $mxp = 40$ et $n_{nd} = 0\%$, le NMI de l'algorithme *Adaptive Brim* vaut 0,81 et quand $n_{nd} = 40\%$, le NMI du même algorithme vaut 0.58. Pour l'algorithme *LPBrim*, avec un $mxp = 40$ et $n_{nd} = 0\%$, le NMI vaut 0.77 et quand n_{nd} atteint 40%, le NMI de cet algorithme diminue à 0.64. L'algorithme *BiLouvain* a trouvé également une difficulté quand le pourcentage de nœuds non discriminants est supérieur à 10%, il retourne une valeur de NMI égale à 0.63 dans le cas où $n_{nd} = 30\%$.

Par contre, les valeurs de NMI données par notre approche sont souvent meilleures par rapport aux approches mentionnées ci-dessus (*Adaptive Brim, LPBrim et BiLouvain*). En outre, notre approche se compare avec l'algorithme *LPAb+* dans le cas où $mxp = 40$ et $n_{nd} <= 20\%$, et meilleure quand $mxp = 40$ et $n_{nd} = 30\%$. Dans le cas où $mxp = 40$ et $n_{nd} <= 40\%$, toutes les approches optées pour la comparaison, ainsi que notre approche, ont donné des valeurs semblables qui gravitent autour d'une valeur de NMI de 0.6. En terminant, la méthode proposée a fait preuve d'une grande stabilité face aux différentes valeurs de paramètre de mixage mxp et de pourcentage de nœuds non discriminants n_{nd}. Elle est souvent meilleure que les autres approches et se compare parfois avec elles.

4.3 Expérimentations sur les graphes réels

Dans cette section, nous évaluons la performance de notre approche proposée dans ce mémoire ainsi que les autres approches dédiées pour la comparaison (*Bi-Louvain, Adaptive Brim, LPBrim et LPAB+*) sur des graphes réels de différentes

tailles[6]. Contrairement aux graphes synthétiques où on connaît l'appartenance de chaque nœud à sa communauté, la structure de communautés prédéfinies est souvent manquante dans les graphes réels (nous n'avons aucune connaissance *à priori* sur le partitionnement des nœuds dans le graphe biparti sous investigation) à l'exception des graphes benchmarks où cette structure est connue. De ce fait, nous utilisons des critères externes (déjà mentionnés ci-dessus) pour évaluer la qualité des résultats obtenus.

Il convient de rappeler ici que l'algorithme *BiSBM* est considéré seulement pour les expérimentations sur le graphe réel benchmark, car il exige que le nombre de communautés soit fixé à l'avance. Dans ce qui suit, nous allons présenter les résultats obtenus de notre évaluation en deux parties : Dans la première partie, nous utilisons un graphe considéré comme benchmark dans le domaine. Dans la deuxième partie, nous effectuons des expérimentations sur cinq graphes réels pour lesquels aucune connaissance *à priori* sur les structures de communautés n'est disponible.

4.3.1 Partie 1 : Expérimentations sur un graphe benchmark

Dans cette partie d'expérimentation, nous allons établir des comparaisons sur un graphe benchmark réel, appelé *Southern Women*, où la partition prédéfinie est déjà connue. Pour comparer les résultats obtenus de chaque algorithme avec la partition prédéfinie, nous utilisons le NMI. Dans ce qui suit, nous allons présenter une description de ce graphe ainsi que les différents résultats obtenus par chaque algorithme.

---

6. Les données des graphes réels que nous avons utilisées pour la comparaison sont disponibles dans http ://konect.uni-koblenz.de/

Le réseau *Southern Women* :

C'est un réseau social biparti qui contient 18 nœuds représentant des femmes ainsi que 14 nœuds désignant des événements et 89 liens reliant ces deux types de nœuds. Il s'agit d'un réseau très populaire et largement utilisé comme un benchmark par les chercheurs pour valider leurs approches proposées dans le cadre des graphes bipartis ((Barber, 2007) ; (Murata, 2009) ; (Suzuki et Wakita, 2009) ; (Liu et Murata, 2010b) ; (Larremore *et al.*, 2014) et (Pesantez et Kalyanaraman, 2017)), car sa structure de communautés est connue (cf. tableau 4.1).

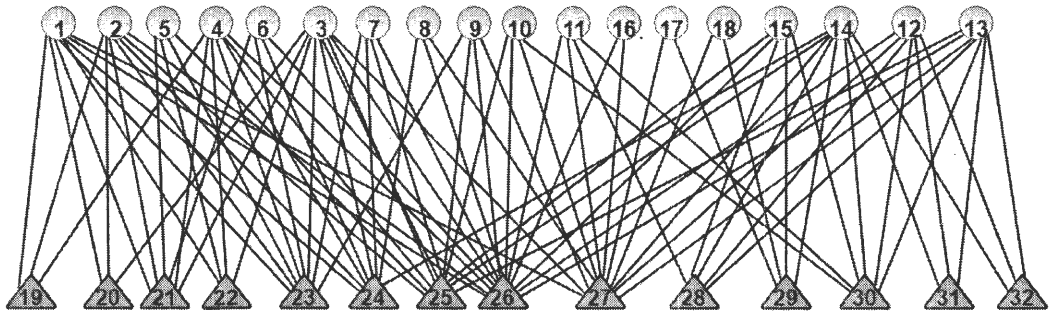


Figure 4.5: Le graphe biparti *Southern Women.*

La figure 4.5 illustre les interactions entre les différents types de nœuds du graphe *Southern Women.* Chaque lien dans cette figure relate la participation d'une femme à un événement. Les données de ce graphe font partie des résultats d'une étude comparative de la catégorie sociale de personnes noires et blanches dans la ville de *Natchez, Mississippi*, USA. La structure de communautés prédéfinie du graphe *Southern Women* est présentée dans le tableau 4.1 ci-dessus.

Tableau 4.1: Le partitionnement prédéfini du graphe *Southern Women*.

#Cluster V_1	#Nœuds	Nom de femmes	#Cluster V_2	#Nœuds	ID événements
1	1	Evelyn	**3**	19	06/27
	2	Laura		20	03/02
	3	Theresa		21	04/12
	4	Brenda		22	09/26
	5	Charlotte		23	02/25
	6	Frances		24	05/19
	7	Eleanor		25	03/15
	8	Pearl		26	09/16
	9	Ruth			
2	10	Verne	**4**	27	04/08
	11	Myra		28	06/10
	12	Katherine		29	02/23
	13	Sylvia		30	04/07
	14	Nora			
	13	Helen		31	11/21
	16	Dorothy			
	17	Olivia		32	08/03
	18	Flora			

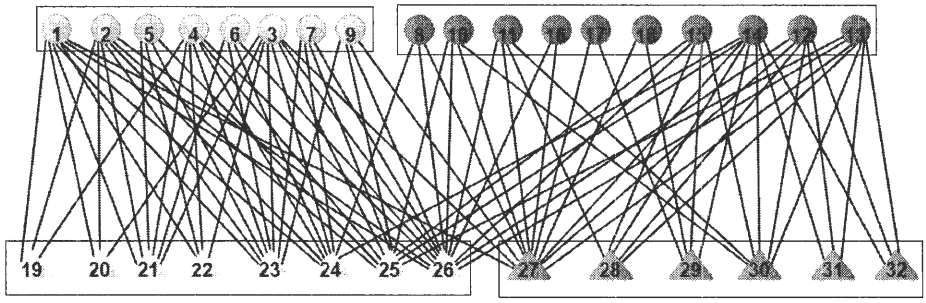


Figure 4.6: Structure de communautés identifiée par notre approche sur le réseau *Southern Women*.

Comme mentionné dans le tableau 4.1 et tel que indiqué dans (Gardner *et al.*, 1941), *Southern Women* est divisé en 4 communautés, à savoir, 2 communautés qui contiennent les nœuds femmes et 2 communautés qui contiennent les nœuds événements. D'après la figure 4.6 ci-dessus, la structure de communautés identifiée par notre approche est très proche à la structure réelle (présentée par le tableau 4.1) avec 2 communautés dans chaque type de nœuds. Notre approche a également identifié le même partitionnement réel pour les nœuds événements (cf. tableau 4.1 et figure 4.6). Cependant, pour les nœuds qui représentent les femmes, précisément, le nœud qui désigne une femme nommée *Pearl* (étiqueté par le nœud 8 dans le tableau 4.1) est placé dans le cluster 2 (cf. figure 4.6), tandis que, son cluster réel est le cluster 1. Comme Madame *Pearl* (nœud 8) avait participé aux trois événements suivants : "05/19", "09/16" et "04/08", étiquetés par les nœuds 24, 26 et 27, respectivement (cf. figure 4.5), notre algorithme a affecté le nœud 8 au cluster 2 en se basant sur l'événement "04/08" (étiqueté par le nœud 27) qui appartient à la communauté 4. Nous pouvons expliquer cela par le fait que le nœud 27 est considéré comme plus discriminant par rapport aux autres voisins du nœud 8 (les nœuds 24 et 26) et le *co-cluster* de la communauté 4 est le cluster 2. Un point notable de notre approche dans ce graphe benchmark, est qu'elle est

classée en second après l'algorithme *BiLouvain* avec une valeur de NMI égale à 0.87 (cf. tableau 4.2).

Tableau 4.2: Les valeurs de NMI pour le graphe *Southern Women.*

Algorithmes	Southern Women
Approche proposée	0.87
Bilouvain	**1**
Adaptive Brim	0.6
LPBrim	0.58
LPAb+	0.60
BiSBM	0.7

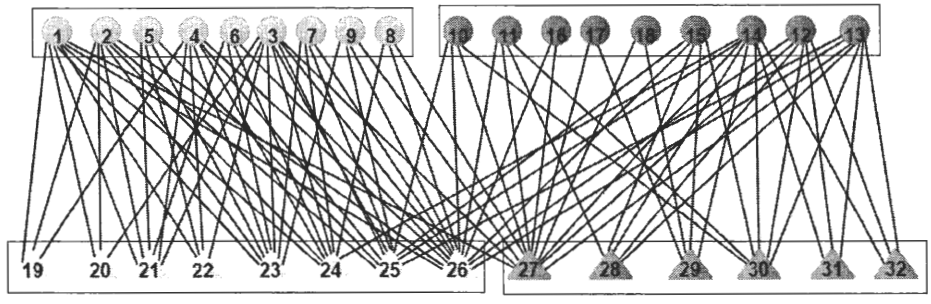


Figure 4.7: Structure de communautés identifiée par l'algorithme *BiLouvain* sur le réseau *Southern Women.*

Le tableau 4.2 et la figure 4.7 illustrent que la structure de communautés identifiée par l'algorithme *BiLouvain* est semblable à la structure réelle présentée dans le tableau 4.1 et qu'il retourne la valeur de NMI la plus élevée. L'algorithme *BiSBM* est en troisième position et il donne le même nombre de communautés réel. Cependant, sa qualité de résultat est plus faible que notre approche et

l'algorithme *BiLouvain* (NMI = 0.7). Les autres algorithmes, à savoir, *Adaptive Brim, LPBrim et LPAb+* ont donné des valeurs presque semblables qui gravitent autour d'une valeur de NMI de 0.6. Il convient de noter, toutefois, que ces valeurs sont plus faibles par rapport à notre approche et l'algorithme *BiLouvain.* Les figures 4.8, 4.9, 4.10, et 4.11 ci-dessous, illustrent, respectivement, les résultats de *Adaptive Brim, LPBrim, LPAb+ et BiSBM.*

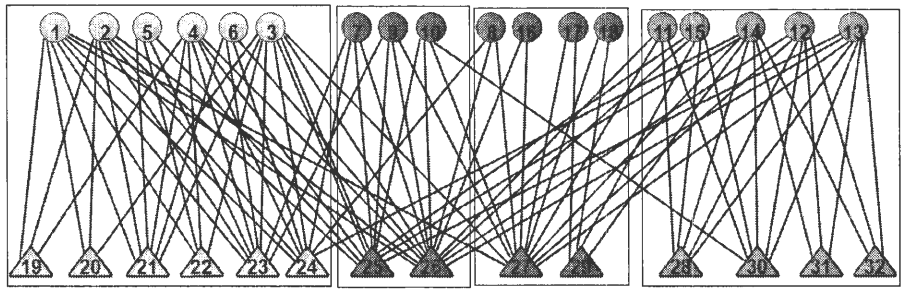


Figure 4.8: Structure de communautés identifiée par l'algorithme *Adaptive Brim* sur le réseau *Southern Women.*

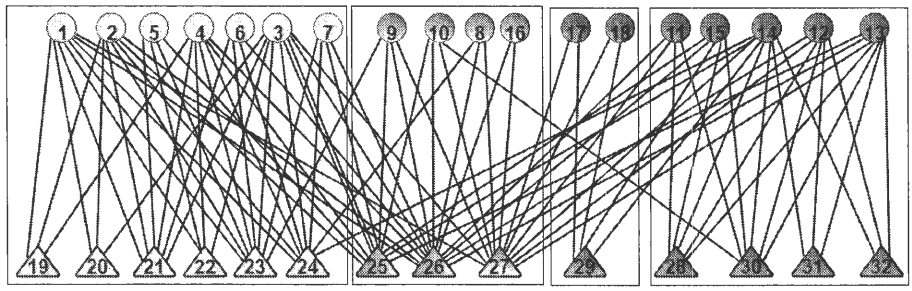


Figure 4.9: Structure de communautés identifiée par l'algorithme *LPBrim* sur le réseau *Southern Women.*

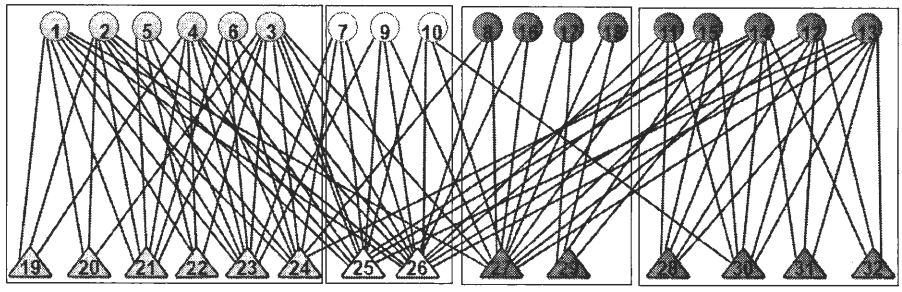


Figure 4.10: Structure de communautés identifiée par l'algorithme *LPAb+* sur le réseau *Southern Women.*

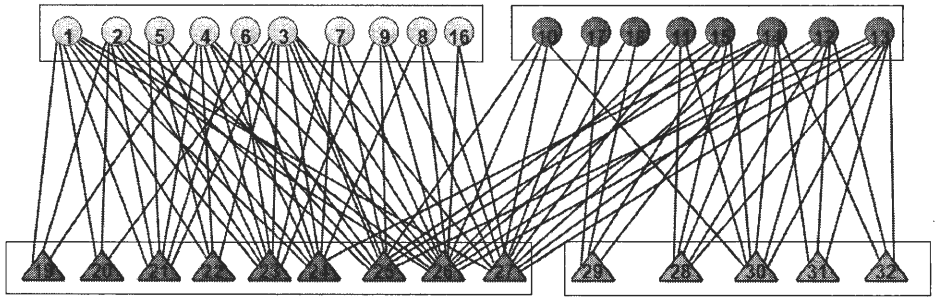


Figure 4.11: Structure de communautés identifiée par l'algorithme *LBiSBM* sur le réseau *Southern Women.*

4.3.2 Partie 2 : Expérimentations sur des graphes réels non étiquetés

Dans cette partie, nous évaluons notre approche ainsi que les autres algorithmes considérés pour la comparaison (*Adaptive Brim, LPBrim, LPAb+ et BiLouvain*) sur des graphes réels non étiquetés, c.-à-d., des graphes qui n'ont aucune structure de communautés connue *à priori.* Le tableau 4.3 relate des informations au sujet de la taille des graphes sélectionnés. Une description de chaque graphe est donnée dans ce qui suit.

Tableau 4.3: Statistiques sur les graphes réels non étiquetés.

Graphes bipartis	$\mid V_1 \mid$	$\mid V_2 \mid$	$\mid E \mid$
Corporate L	20	24	99
Americain R	136	5	160
Crime	829	551	1476
Malaria	297	806	2965
arXiv	16726	22015	58595

Le réseau Corporate leadership (Corporate L) :

Ce graphe biparti contient des informations sur les personnes et les entreprises dans lesquelles elles ont eu une position de leader. Les nœuds de type V_1 représentent les personnes et les nœuds de type V_2 dénotent les entreprises. Chaque lien entre ces deux types de nœuds indique que la personne avait une position de leader dans cette entreprise. Le graphe contient 20 nœuds de type V_1, 24 nœuds de type V_2 et 99 liens.

Le réseau American Revolution (Americain R) :

Ce graphe biparti est formé de 136 personnes et de 5 organisations connues durant l'époque qui précède la révolution américaine. Les nœuds de type V_1 représentent les personnes et les nœuds de type V_2 dénotent les organisations. Chaque lien entre une personne et une organisation désigne que la personne en question était un membre dans cette organisation. Ce graphe contient 160 liens au total.

Le réseau Crime :

Crime est un graphe biparti qui regroupe des personnes ayant apparu dans au moins une affaire criminelle en tant que suspect, victime, témoin ou à la fois un

suspect et une victime en même temps. Les nœuds de type V_1 représentent des personnes, tandis que les nœuds de type V_2 dénotent les crimes. Un lien entre une personne et un crime désigne que cette personne a été impliquée dans ce crime. Le graphe contient 829 nœuds de type V_1, 551 nœuds de type V_2 et 1 476 liens.

Le réseau Malaria :

Ce graphe provient du *Plasmodium falciparum* qui est un parasite du paludisme. Le parasite évite le système immunitaire humain par l'intermédiaire d'un camouflage de protéines, qui est encodé par les gènes *var* (Rask *et al.*, 2010). Afin de créer de nouveaux camouflages, les gènes *var* se recombinent fréquemment, ce qui revient à un épissage et un remaniement en sous-chaînes génétiques, donnant lieu à des structures de communautés (Bull *et al.*, 2008). Les deux types de nœuds correspondent aux gènes et à leurs sous-chaînes, et chaque sous-chaîne génétique se connecte avec chaque gène dans lequel elle est présente. Le graphe est composé de 297 gènes, 806 sous-chaînes et 2 965 liens.

Le réseau arXiv :

arXiv est une archive de prépublications électroniques d'articles scientifiques dans plusieurs domaines, à savoir, la physique, l'astrophysique, la mathématique, l'informatique et la biologie. Ce site est accessible gratuitement par Internet et il a été originellement développé par *Paul Ginsparg* en 1991 comme une archive pour les prépublications dans le domaine physique. Ce graphe biparti relate les liens de publication entre les auteurs et les articles dans la section de la matière condensée de arXiv entre 1995 et 1999. Il est formé de 16726 nœuds qui correspondent aux auteurs et 22015 nœuds qui représentent les articles ainsi que 58595 liens désignant les relations de publications.

Tableau 4.4: Valeurs de la *couverture*.

Algorithmes	Corporate L	Americain R	Crime	Malaria	arXiv
Approche proposée	**0.77**	**0.85**	0.90	0.68	**0.84**
Bilouvain	0.64	**0.85**	0.80	0.64	0.81
Adaptive Brim	0.62	**0.85**	0.80	0.66	-
LPBrim	0.63	**0.85**	0.82	0.66	-
LPAb+	0.62	**0.85**	**0.96**	**0.75**	-

Tableau 4.5: Valeurs de la *densité de modularité* M_D.

Algorithmes	Corporate L	Americain R	Crime	Malaria	arXiv
Approche proposée	**1.5**	**0.62**	0.75	0.074	**0.78**
Bilouvain	-0.13	**0.62**	0.44	-0.20	0.66
Adaptive Brim	-0.27	**0.62**	0.43	-0.01	-
LPBrim	-0.20	**0.62**	0.49	0.022	-
LPAb+	-0.27	**0.62**	**0.95**	**0.69**	-

Les tableaux 4.4 et 4.5 montrent les résultats de la *couverture* et de la *densité de modularité* pour les différents algorithmes considérés dans nos expérimentations. À noter que la présence de quelques cases vides dans ces deux tableaux (la colonne du graphe arXiv), est due au fait que les algorithmes *Adaptive Brim, LPBrim et LPAb+* n'ont pas été en mesure de retourneré un résultat, et ce, même après plus de 48 heures de temps d'exécution.

Nous commençons l'analyse par le graphe *Corporate leadership (Corporate L)*, où notre approche réussite à retourner la meilleure qualité de partitionnement avec

une valeur de *couverture* égale à 0.77 et une valeur de *densité de modularité* égale à 1.5 (à titre de rappel, la plus grande valeur de *couverture* et *densité de modularité*, suggère un bon résultat). Le deuxième meilleur résultat est retourné par l'algorithme *BiLouvain* avec une valeur de *couverture* = 0.64 et une *densité de modularité* M_D = -0.13. La performance des autres approches, à savoir, *Adaptive Brim, LPBrim et LPAb+*, est bien inférieure à notre approche ainsi qu'à l'algorithme *BiLouvain.*

En ce qui concerne le graphe *Americain Revolution (americain R)*, comme nous pouvons le constateré à partir des deux tableaux (4.4 et 4.5), toutes les approches comparées ont retourné le même partitionnement (même résultat). Cela peut être expliqué par le fait que le graphe (*Americain Revolution*) contient une structure de communautés facilement identifiable.

Au sujet du graphe *Crime* et selon les valeurs de métriques présentées dans les tableaux 4.4 et 4.5, l'algorithme *LPAb+* montre une meilleure performance (*couverture* = 0.95 et la *densité de modularité* M_D = 0.95). Notre approche a, quant à elle, retournée le deuxième meilleur résultat avec une valeur de *couverture* égale à 0.90 et une *densité de modularité* M_D de 0.75. L'algorithme *LPBrim* vient en troisième position avec une valeur de *couverture* égale à 0.82 et une valeur de *densité de modularité* égale à 0.49. Les algorithmes *BiLouvian* et *Adaptive Brim*, quant à eux, sont moins compétitifs, comparativement aux autres approches.

Pour le graphe *Malaria*, les deux tableaux 4.4 et 4.5 suggèrent que l'algorithme *LPAb+* donne le meilleur résultat (*couverture* = 0.75 et *densité de modularité* M_D = 0.69). Le deuxième meilleur résultat est retourné par notre approche avec une valeur de *couverture* égale à 0.68 et une valeur de *densité de modularité*

égale à 0.074. Quant aux autres approches, à savoir, *Adaptive Brim, LPBrim et BiLouvain* et d'après les valeurs des métriques externes, nous remarquons que ces trois approches viennent en dernier et elles ont presque la même performance pour ce graphe.

Passons au dernier graphe (arXinv), comme ce graphe est de grande taille (avec 38 741 nœuds et 58 598 liens) et d'après les cases vides dans les deux tableaux (4.4 et 4.5), la plupart des algorithmes, tels que *LPBrim, Adaptive Brim* et *LPAb+* ne retournent rien (ne convergent pas), et même après avoir laissé ces algorithmes rouler plus de 48 heures. Cela suggère que ces algorithmes (*Adaptive Brim, LPBrim et LPAb+*) trouvent des difficultés avec les graphes de grande taille. Cependant, l'approche proposée a démontré une meilleure performance et a donné le meilleur résultat dans ce graphe avec une valeur de *couverture* égale à 0.84 et une valeur de *densité de modularité* $M_D = 0.78$. Quant à l'algorithme *BiLouvain*, il arrive en seconde place avec des résultats relativement bons (*couverture* = 0.81 et *densité de modularité* $M_D = 0.78$).

4.4 Conclusion

Les résultats présentés dans ce chapitre sur des graphes synthétiques et réels suggèrent que l'approche proposée a démontré une performance compétitive et, dans plusieurs cas, meilleure par rapport aux autres approches existantes. En effet, la stratégie de détection de communautés que nous avons élaborée dans ce mémoire a pu donner de bons partitionnements, peu importe la valeur de paramètre de mixage (qui relate la présence des nœuds massivement connectés avec les autres nœuds du graphe), le pourcentage des nœuds non discriminants (c'est le cas des nœuds avec des connexions éparses) et la taille du graphe biparti.

CHAPITRE V

CONCLUSION ET PERSPECTIVES

Dans le cadre de ce mémoire, nous avons exposé la problématique de détection de communautés dans les graphes bipartis. Nous avons soulevé le fait que les approches existantes souffrent de plusieurs problèmes, à savoir : (1) la perte d'informations à la suite de la transformation du graphe biparti vers des graphes homogènes (c'est le cas des approches basées sur la projection), (2) la sensibilité à la présence des nœuds non discriminants, c.-à-d., les nœuds qui sont distribués de façon éparse dans le graphe et les nœuds qui se connectent massivement aux autres nœuds du graphe, (3) la nécessité de spécifier manuellement le nombre de communautés dans le graphe biparti (comme l'algorithme *BiSBM*), (4) ainsi que la restriction liée au fait que les deux types de nœuds du graphe biparti doivent avoir le même nombre de communautés (c'est le cas des approches basées sur les graphes homogènes). Afin de pallier ces limites, nous avons proposé une nouvelle approche de détection de communautés dans les graphes bipartis. Notre approche opère en deux principales phases. La première se base sur un mécanisme de *scoring*, incorporé dans une fonction objective, qui lui permet de distinguer les nœuds discriminants des nœuds non discriminants lors du processus de clustering. La deuxième phase consiste à raffiner les résultats de la première phase par le biais d'un processus itératif qui permet à son tour d'identifier une structure de communautés relativement optimale.

En conclusion, les tests empiriques que nous avons réalisés sur des graphes réels et synthétiques de diverses tailles et caractéristiques ont confirmé la stabilité et la performance de notre approche par rapport aux autres algorithmes existants. Un point notable de notre approche est qu'elle donne de meilleurs résultats dans les graphes de grande taille et de densité élevée. Par contre, d'autres approches, à savoir, *Adaptive Brim, LPBrim et LPAb+* trouvent de difficultés avec ce genre de graphes et ne convergent vers aucun résultat. En outre, notre approche a démontré également une meilleure performance avec la présence des nœuds non discriminants.

Pour rendre notre approche encore plus compétitive, certaines améliorations peuvent être réalisées dans le futur. Par exemple, rendre la correspondance entre les communautés de plusieurs à plusieurs, c.-à-d., une communauté dans V_1 peut être connectée avec une ou plusieurs communautés dans V_2 et vice versa. Une autre idée consisterait à étendre notre approche afin de supporter les graphes bipartis pondérés et également étendre le mécanisme de pondération pour qu'il puisse supporter ce genre de graphe. Une autre possibilité à envisager serait de rendre notre approche applicable dans des graphes hétérogènes de différents types de nœuds et de liens. Finalement, il est souhaitable d'introduire la notion de chevauchement de communautés. Cela permet à notre approche de supporter le partitionnement flou où un nœud peut appartenir à plusieurs communautés.

RÉFÉRENCES

Alzahrani, T. et Horadam, K. (2016). Community detection in bipartite networks : Algorithms and case studies. In *Complex Systems and Networks* 25–50. Springer.

Barber, M. J. (2007). Modularity and community detection in bipartite networks. *Physical Review E*, *76*(6), 066102.

Barber, M. J. et Clark, J. W. (2009). Detecting network communities by propagating labels under constraints. *Physical Review E*, *80*(2), 026129.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R. et Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics : theory and experiment*, *2008*(10), P10008.

Bouguessa, M. (2011). A practical approach for clustering transaction data. Dans *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, 265–279. Springer.

Brandes, U. et Erlebach, T. (2005). Network analysis : methodological foundation.

Brun, C., Chevenet, F., Martin, D., Wojcik, J., Guénoche, A. et Jacq, B. (2003). Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome biology*, *5*(1), R6.

Brun, C., Herrmann, C. et Guénoche, A. (2004). Clustering proteins from interaction networks for the prediction of cellular functions. *BMC bioinformatics*, *5*(1), 95.

Bull, P. C., Buckee, C. O., Kyes, S., Kortok, M. M., Thathy, V., Guyah, B., Stoute, J. A., Newbold, C. I. et Marsh, K. (2008). Plasmodium falciparum antigenic variation. mapping mosaic var gene sequences onto a network of shared, highly polymorphic sequence blocks. *Molecular microbiology*, *68*(6), 1519–1534.

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

Gardner, B. B. B., Gardner, M. R., Lloyd, W. et Lloyd, W. (1941). *Deep South : a social anthropological study of caste and class.* Chicago, Ill. : The University of Chicago Press.

Karrer, B. et Newman, M. E. (2011). Stochastic blockmodels and community structure in networks. *Physical review E*, *83*(1), 016107.

Kleinberg, J. et Lawrence, S. (2001). The structure of the web. *Science*, *294*(5548), 1849–1850.

Lancichinetti, A. et Fortunato, S. (2009). Community detection algorithms : a comparative analysis. *Physical review E*, *80*(5), 056117.

Larremore, D. B., Clauset, A. et Jacobs, A. Z. (2014). Efficiently inferring community structure in bipartite networks. *Physical Review E*, *90*(1), 012805.

Liu, X. et Murata, T. (2010a). Community detection in large-scale bipartite networks. *Information and Media Technologies*, *5*(1), 184–192.

Liu, X. et Murata, T. (2010b). An efficient algorithm for optimizing bipartite modularity in bipartite networks. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, *14*(4), 408–415.

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

Moody, J. et White, D. R. (2003). Structural cohesion and embeddedness : A hierarchical concept of social groups. *American Sociological Review*, 103–127.

Murata, T. (2009). Detecting communities from bipartite networks based on bipartite modularities. Dans *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 4, 50–57. IEEE.

Newman, M. E. (2004a). Detecting community structure in networks. *The European Physical Journal B*, *38*(2), 321–330.

Newman, M. E. (2004b). Fast algorithm for detecting community structure in networks. *Physical review E*, *69*(6), 066133.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, *103*(23), 8577–8582.

Newman, M. E. et Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, *69*(2), 026113.

Pesantez, P. G. et Kalyanaraman, A. (2017). Efficient detection of communities in biological bipartite networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics.*

Pons, P. et Latapy, M. (2005). Computing communities in large networks using random walks. Dans *International symposium on computer and information sciences*, 284–293. Springer.

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

Rask, T. S., Hansen, D. A., Theander, T. G., Pedersen, A. G. et Lavstsen, T.

(2010). Plasmodium falciparum erythrocyte membrane protein 1 diversity in seven genomes–divide and conquer. *PLoS computational biology*, *6*(9), e1000933.

Schuetz, P. et Caflisch, A. (2008a). Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Review E*, *77*(4), 046112.

Schuetz, P. et Caflisch, A. (2008b). Multistep greedy algorithm identifies community structure in real-world and computer-generated networks. *Physical Review E*, *78*(2), 026112.

Suzuki, K. et Wakita, K. (2009). Extracting multi-facet community structure from bipartite networks. Dans *Computational Science and Engineering, 2009. CSE'09. International Conference on*, volume 4, 312–319. IEEE.

Xu, Y. et Chen, L. (2014). Modularity density for evaluating community structure in bipartite networks. *Open Automation and Control Systems Journal*, *6*, 684–691.