Les communautés dans les multigraphes de co-appartenance : Application aux listes Twitter

Mohamed Benabdelkrim

▶ **To cite this version:**

Mohamed Benabdelkrim. Les communautés dans les multigraphes de co-appartenance : Application aux listes Twitter. Réseaux sociaux et d'information [cs.SI]. Université de Lyon, 2021. Français. NNT : 2021LYSEI100 . tel-03670875

HAL Id: tel-03670875
https://theses.hal.science/tel-03670875

Submitted on 17 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre NNT : 2021LYSEI100

Thèse de doctorat de l'Université de Lyon
opérée au sein de
L'INSA de Lyon

Ecole Doctorale N° 512
Mathématiques et Informatique (InfoMaths)

Spécialité / discipline de doctorat : Informatique

À soutenir publiquement le 09 décembre 2021 par
Mohamed Benabdelkrim

Les communautés dans les multigraphes de co-appartenance : application aux listes Twitter

Devant le jury composé de :

Pr. Céline Robardet	INSA Lyon	Directrice de thèse
Pr. Jean Savinien	EM Lyon	Co-directeur de thèse
Pr. Anne Boyer	Université de Lorraine	Rapporteur
Pr. Jean-Loup Guillaume	Université de La Rochelle	Rapporteur
Pr. Sandra Bringay	Université de Montpellier 3	Examinatrice
Pr. Nicolas Labroche	Université François Rabelais de Tours	Examinateur

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Table des matières

Table des matières **i**

1 Introduction **3**

2 État de l'art **9**
2.1 Notions de graphe, multigraphe et graphe de co-appartenance 11
2.2 Sélection de sous-graphes d'intérêt dans des multigraphes 13
2.2.1 Sous-graphe d'intérêt à partir du graphe agrégé 13
2.2.2 Sous-graphes d'intérêt dans des multigraphes 15
2.3 Détection de communautés dans des multigraphes 17
2.3.1 Méthodes basées sur la découverte de communautés dans des graphes simples 18
2.3.1.1 Méthodes hiérarchiques 18
2.3.1.2 Méthodes par optimisation de la modularité 20
2.3.1.3 Représentation dans un espace vectoriel et clustering 21
2.3.1.4 Propagation de labels 23
2.3.1.5 Modèles de graphes 24
2.3.1.6 Discussion : avantages et inconvénients des différentes approches 25
2.3.2 Agrégation des communautés de plusieurs couches 25
2.3.3 Détection directe de communautés dans un multigraphe 26
2.3.3.1 Approche par optimisation de grandeurs 26
2.3.3.2 Représentation dans un espace vectoriel et clustering 28
2.4 Évaluation de la qualité d'un partitionnement 29
2.4.1 Graphes synthétiques 29
2.4.2 Comparaison avec les communautés réelles 29
2.4.3 Métriques d'évaluation 31
2.5 Applications aux réseaux sociaux 32
2.6 Bilan 34

3 Sélection de sous-graphes d'intérêt **37**
3.1 Introduction 37
3.2 Les multigraphes de co-appartenance 37
3.3 Présentation des jeux de données 39
3.4 Méthodologie 39

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

3.4.1 Pré-traitement des attributs textuels . . . 40
3.4.2 Première stratégie : *Double couronnes* . . . 41
3.4.3 Deuxième stratégie : *Marche aléatoire* . . . 42
3.5 Résultats et comparaisons . . . 43
3.5.1 *Double couronnes* . . . 44
3.5.2 *Marche aléatoire* . . . 46
3.5.3 Comparaison des deux approches . . . 50
3.5.4 Conclusion . . . 51

4 Détection de communautés par extraction de motifs symboliques 53
4.1 Introduction . . . 53
4.2 Extraction de motifs symboliques sous contraintes . . . 54
4.2.1 Le langage de motifs symboliques . . . 55
4.2.2 Les contraintes . . . 56
4.2.3 Les algorithmes de calcul . . . 59
4.2.3.1 Calcul exhaustif de motifs . . . 59
4.2.3.2 Calcul heuristique . . . 60
4.2.3.3 Calcul par échantillonnage . . . 61
4.2.4 Filtrage des motifs . . . 62
4.3 Communautés dans un multigraphe de co-appartenance . . . 63
4.3.1 Le langage des sous-graphes . . . 63
4.3.2 Des communautés partageant des nœuds et des attributs textuels . . . 64
4.3.2.1 Propriétés des contraintes . . . 64
4.3.2.2 Fermeture . . . 65
4.3.3 Différents algorithmes de calcul de communautés . . . 67
4.3.3.1 Par énumération exhaustive . . . 67
4.3.3.2 Par échantillonnage . . . 69
4.4 Résultats et comparaisons . . . 73
4.4.1 Étude du réseau social Twitter . . . 75
4.4.2 Étude de l'encyclopédie collaborative Wikipedia . . . 86
4.5 Conclusion . . . 91

5 Détection de communautés par extraction de motifs numériques 93
5.1 Extraction de motifs dans des données numériques . . . 94
5.1.1 Structures de motifs . . . 95
5.1.2 Motifs d'intervalles dans un jeu de données numériques . . . 96
5.1.3 Langage des motifs d'intervalles . . . 97
5.2 Traitement du langage naturel . . . 98
5.3 Motifs numériques pour les multigraphes de co-appartenance . . . 99
5.3.1 Construction du jeu de données numériques . . . 99
5.3.2 Calcul de similarités entre attributs textuels avec le modèle BERT . . . 101
5.3.3 Extraction des top-k motifs d'intervalles . . . 102
5.4 Résultats et comparaisons . . . 104
5.4.1 Description des jeux de données et des méthodes compétitives . . . 105
5.4.2 Évaluation de l'algorithme . . . 106
5.4.3 La sémantique améliore-t-elle la qualité des communautés ? . . . 109

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

5.5 Conclusion 109

6 Conclusion et perspectives **111**

A Graphes : définitions **115**

B Analyse formelle de concepts : définitions **119**

C Diagramme de synthèse **123**

D Organization studies application **125**

Bibliographie **127**

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Résumé

Les graphes sont considérés comme l'outil mathématique de prédilection pour la modélisation d'une relation entre des paires d'entités. Ils peuvent être utilisés pour modéliser une relation de connaissance entre des personnes, ou encore une relation de similarité entre des mots. Parmi ces relations, celle de l'appartenance commune d'un ensemble d'entités à un ensemble de groupes permet de définir un multigraphe de co-appartenance. Dans ce multigraphe, les entités correspondent aux nœuds, les relations aux liens et les groupes aux couches. Par ailleurs, la relation d'appartenance commune étant transitive, les nœuds d'une même couche forment une clique. Et dans plusieurs cas, des descriptions textuelles sont associées aux groupes et sont représentées par des attributs associés aux couches dans le multigraphe de co-appartenance. L'objectif de cette thèse est de développer des outils de détection de communautés de nœuds homogènes dans les multigraphes de co-appartenance en utilisant les informations structurelle et sémantique qui y sont contenues. La relation d'appartenance commune à des groupes, que nous étudions, se retrouve dans plusieurs situations réelles comme, par exemple, la participation d'un ensemble de chercheurs à des conférences ou la rédaction commune par des auteurs d'un ensemble d'articles. Dans les réseaux sociaux, cette relation est encore plus courante et nous la retrouvons, entre autres, dans le réseau social Twitter avec l'appartenance commune des utilisateurs aux listes.

Dans ce travail, nous définissons formellement les multigraphes de co-appartenance et soulignons les propriétés qui les distinguent des multigraphes classiques. Nous présentons deux algorithmes pour y délimiter des sous-graphes qui correspondent à des domaines d'intérêt pour l'analyste utilisateur de l'outil de détection de communautés. Nous expliquons, aussi, comment adapter certaines méthodes de fouille de données pour l'extraction de motifs symboliques et numériques à la détection de communautés dans les multigraphes de co-appartenance. Les résultats obtenus sont comparés à ceux des méthodes de l'état de l'art en utilisant des exemples de multigraphes de co-appartenance issus du réseau social Twitter et de la plateforme collaborative Wikipedia.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Chapitre 1

Introduction

Les graphes sont des objets mathématiques utilisés pour modéliser des relations entre des paires d'entités. Ils sont considérés comme un outil incontournable pour la modélisation de systèmes complexes dans plusieurs domaines d'application comme la biologie pour la structure des protéines, les transports et la logistique pour l'optimisation des trajets, les télécommunications pour le routage des réseaux, ou encore la sociologie et les sciences humaines pour l'analyse des processus de diffusion de nouvelles dans les réseaux sociaux. Avec le développement des systèmes d'information, le nombre et la taille de ces graphes ont augmenté considérablement ce qui a suscité de nombreux travaux de recherche. En plus des méthodes qui ont été développées pour des tâches spécifiques comme la recherche de plus court chemin, ou l'identification de processus de diffusion dans les graphes, une direction de recherche importante a porté sur la détection de sous-graphes exceptionnels en général, et de communautés en particulier. L'intérêt de telles approches est qu'elle permettent de construire une vue synthétique des graphes et ainsi de mieux les appréhender. En effet, les communautés, ou les groupes de nœuds similaires et fortement connectés, permettent de restructurer un graphe en un petit nombre de composantes indépendantes facilitant son analyse.

Les graphes sont composés, dans leur version la plus simple, d'un ensemble de nœuds et d'un ensemble de liens. Les nœuds représentent des entités, et les liens modélisent la présence d'une relation entre deux entités. Par exemple, la relation "a est une connaissance de b" peut être modélisée par un graphe simple dans lequel les nœuds sont les personnes (a,b,...) et l'existence d'un lien entre deux nœuds signifie que les deux entités correspondantes vérifient la relation c'est-à-dire que les deux personnes se connaissent. Cette structure de graphe simple a été enrichie de différentes façons afin d'intégrer des données additionnelles, Les relations asymétriques comme "a envoie un message à b", sont modélisées par des graphes dirigés dans lesquels un lien seul ne détermine pas la réciprocité de la relation. Pour les cas où la relation a une intensité mesurable, comme "le nombre de messages échangés" pour les exemples de relation précédents, il est possible d'associer cette mesure aux liens et obtenir de cette façon un graphe pondéré. Les graphes attribués sont, eux, utilisés quand des attributs qui caractérisent la relation ou les entités sont disponibles. En reprenant l'exemple de connaissance entre personnes, certaines informations associées aux personnes comme leur âge ou leur profession peuvent être associées aux nœuds, et les contenus des messages échangés peuvent constituer des attributs de liens. Un autre cas de figure concerne les relations

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

à types multiples qui sont modélisées par des graphes multicouches (ou multigraphes). En effet, les types de relation de connaissance sont multiples : ça peut être une relation familiale, amicale, de travail ou autres. Enfin, il est aussi possible de prendre en compte l'évolution temporelle de la relation à l'aide des graphes dynamiques. En pratique, ce sont des combinaisons de ces types de graphes qui servent à modéliser les relations issues de données réelles. Par exemple, un réseau de transport multimodal peut être modélisé à l'aide d'un graphe multicouche pondéré et attribué sur les nœuds.

Toutefois, un aspect répandu dans les graphes rencontrés dans des situations réelles a été peu considéré jusqu'à présent : le cas des graphes multicouches pour lesquels les relations entre nœuds sont transitives. La relation de fraternité est un exemple de relations transitives : si a est frère de b et c est frère de b, alors a et c sont frères aussi. C'est le cas pour plusieurs autres exemples de réseaux comme les réseaux de co-citation, les réseaux de co-achat et les réseaux de co-appartenance à des listes. Ces réseaux représentent bien une relation de co-appartenance à un ensemble de groupes, et ces groupes sont respectivement les articles cités, les marchandises achetées et les listes. Dans le multigraphe de co-appartenance, ces groupes correspondent aux couches dont les nœuds sont complètement connectés par une relation transitive, les communautés dans de tels graphes correspondent à de nouveaux ensembles de nœuds qui apparaissent simultanément dans différentes couches.

Pendant l'analyse de ces graphes multicouches issus de contextes réels, nous avons remarqué qu'à chaque couche est généralement associée une donnée textuelle non structurée qui caractérise et contextualise l'interaction. Par exemple, les graphes de co-citation et de co-rédaction de publications scientifiques peuvent être enrichis avec des informations supplémentaires telles que la conférence de publication, les mots-clés ou encore le résumé de la publication. Un autre exemple concerne un réseau social dans lequel chaque couche correspond à un type de relations sociales comme 'amis' ou 'collègues'. L'information supplémentaire associée aux couches peut alors être les messages échangés entre membres de la couche. Cette information textuelle est précieuse car elle peut être utilisée pour détecter les communautés de nœuds reliés entre eux dans plusieurs couches et tel que les caractéristiques de ces couches soient similaires. La prise en compte de la similarité sémantique entre les couches rend possible la construction de communautés de nœuds reliés par des liens dont les attributs ont une certaine homogénéité sémantique, comme par exemple les communautés d'auteurs qui ont co-publié plusieurs articles avec des sujets analogues. Les exemples de graphes multicouches avec des attributs textuels associés aux couches sont courants et les situations réelles pour lesquelles cette modélisation est convenable sont multiples comme nous le verrons dans les parties expérimentales des prochains chapitres. Par ailleurs, les couches de ces multigraphes ont souvent une information temporelle concernant leurs temps de début et/ou de fin comme les dates de publication des articles ou les dates de création et de suppression des groupes dans les réseaux sociaux. Ces multigraphes seront définis formellement dans le prochain chapitre sous le nom de **multigraphes de co-appartenance**. Ce chapitre sera aussi l'occasion de présenter un état de l'art sur les méthodes locales et globales de détection de communautés dans les multigraphes.

Nous utiliserons principalement des données issues du réseau social Twitter pour illustrer les multigraphes de co-appartenance et évaluer les performances des méthodes proposées. Plus spécifiquement, nous nous baserons sur la relation d'apparition commune des utilisateurs Twitter dans les listes Twitter pour construire un exemple de multigraphe de co-appartenance. Le sujet de cette thèse porte sur la détection de communautés dans les mul-

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

tigraphes de co-appartenance. Nous cherchons, en particulier, à répondre à la problématique suivante : **Comment détecter des communautés dans les multigraphes de co-appartenance en utilisant l'information topologique du graphe et l'information sémantique de ses attributs textuels ?**

La plupart des algorithmes qui ont été proposés dans la littérature scientifique pour la détection de communautés utilisent la structure du graphe en identifiant et en retirant directement certains liens de sorte à le séparer en parties disjointes, ou encore en optimisant une métrique de connectivité dont la plus populaire est la modularité. Dans ce travail, nous avons choisi de développer une approche basée sur la fouille de données (ou data mining). Cette discipline informatique a pour but d'extraire les informations pertinentes depuis de grands ensembles de données. Initialement, les travaux de recherche de cette discipline s'étaient tournés vers les jeux de données à attributs binaires en se basant sur le formalisme de l'Analyse Formelle de Concepts (AFC). Un exemple courant qui illustre ce type de jeux de données est celui des achats effectués dans un magasin communément appelé exemple du panier de la ménagère. Dans cet exemple, chaque observation correspond à un ensemble d'articles achetés par un client lors de sa visite d'un magasin. La recherche des motifs les plus fréquents dans ces jeux de données permet de déduire des règles d'association entre leurs attributs. Pour l'exemple du panier de la ménagère, les motifs fréquents correspondent aux sous-ensembles d'articles achetés ensemble à plusieurs occasions. Ils permettent de déduire des règles d'association entre les articles du magasin de la forme $A \rightarrow B$ qui veut dire que si un client achète les articles A alors son panier contiendra aussi les articles B. Ces règles peuvent être utilisées comme base pour la prise de décision des responsables marketing du magasin pour établir des promotions ou choisir les meilleurs emplacements pour les produits associés. Dans des travaux plus récents, un nouveau formalisme appelé structures de motifs a été introduit pour permettre d'extraire des motifs depuis des jeux de données dont les attributs ne sont pas forcément binaires. Il permet, entre autres, de traiter les attributs numériques à l'aide des motifs d'intervalles. Dans cette thèse, nous utilisons ces formalismes pour la détection de communautés dans les multigraphes de co-appartenance. Dans un premier temps, nous définissons les communautés comme des ensembles de couches partageant des nœuds et des attributs textuels. Puis, nous proposons une méthode qui prend en compte la richesse des attributs textuels en définissant une relation numérique d'appartenance aux couches qui intègre les similarités sémantiques entres les descriptions des couches.

Contributions

Nous répondons à la problématique en apportant les contributions suivantes :

(1) Nous proposons de modéliser des relations transitives de co-appartenance à l'aide de multigraphes qui peuvent être statiques ou dynamiques :

- Le multigraphe de co-appartenance est un multigraphe attribué sur les couches. Il est construit en disposant chaque groupe dans une couche dont l'attribut textuel est la description du groupe. Les entités sont représentées par les nœuds et les entités d'un même groupe forment une clique dans la couche correspondante.
- Le multigraphe de co-appartenance dynamique est un multigraphe dynamique attribué sur les couches. A la différence du multigraphe de co-appartenance, ses

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

couches sont ordonnées temporellement. Cet ordre temporel associé aux couches du multigraphe découle des dates de création des groupes qui leur correspondent.

Dans la suite de ce rapport, nous nous focalisons principalement sur le multigraphe de co-appartenance statique. Le cas dynamique constitue une perspective de recherche future que nous présenterons en conclusion générale. Nous verrons, qu'en pratique, le multigraphe de co-appartenance a des propriétés spécifiques qui le distinguent d'un multigraphe classique. Ces propriétés, que nous présentons au chapitre 3, concernent entre autres la taille du multigraphe (nombre de nœuds et de couches), la densité moyenne de ses nœuds et la présence de cliques et d'attributs textuels.

(2) En pratique, les multigraphes de co-appartenance sont de très grande taille et peuvent comprendre une variété de domaines. Certains graphes de co-citation d'articles, par exemple, représentent les relations entre des millions d'auteurs qui publient dans divers sujets. Les algorithmes classiques dédiés aux multigraphes nécessitent alors beaucoup de ressources de calcul et peuvent même être inutilisables dans certains cas. Nous proposons d'extraire des sous-graphes qui correspondent aux sujets et domaines qui intéressent l'utilisateur. Une application de cette méthode de sélection de sous-graphes d'intérêt a fait l'objet d'une publication dans le journal M@n@gement (Benabdelkrim et al., 2020a). Dans l'état de l'art associé aux multigraphes, les méthodes locales de détection de communautés permettent de réaliser cette tâche de sélection de sous-graphes d'intérêt. Nous verrons néanmoins que ces méthodes ont des limites inhérentes : les communautés qu'elles retournent sont de très petite taille et elles ne prennent pas en compte les attributs textuels présents sur les couches du multigraphe de co-appartenance. Nous présentons, au chapitre 3, deux procédures d'expansion qui, à partir d'un ensemble de nœuds graines, délimitent les frontières du sous-graphe qui représente le domaine d'intérêt de l'utilisateur de l'outil. Les nœuds graines sont choisis de sorte à représenter les acteurs centraux du domaine d'intérêt :

— La première procédure détermine deux couronnes autour des nœuds graines dans le multigraphe de co-appartenance. Ces deux couronnes sont par la suite filtrées pour ne garder que les nœuds similaires aux nœuds graines en utilisant une fonction score qui calcule les similarités structurelle et sémantique entre couches du multigraphe

— La deuxième procédure est basée sur la simulation de marches aléatoires dans le graphe de co-appartenance. Ces marches ont pour points de départ les nœuds graines et les probabilités de transition sont pondérées avec les poids des liens.

(3) Nous proposons des méthodes de fouille de données binaires et numériques à la détection de communautés dans les multigraphes de co-appartenance. Nous nous intéressons plus spécifiquement aux méthodes d'extraction de motifs symboliques et numériques sous contraintes :

— Pour nous placer dans le formalisme de l'Analyse Formelle de Concepts (AFC (Wille, 1982)), nous montrons dans le chapitre 4 comment construire un contexte formel à partir d'un multigraphe de co-appartenance. Nous utilisons ensuite des algorithmes d'énumération exhaustive et des algorithmes d'échantillonnage pour extraire les motifs symboliques du contexte formel. Ces motifs correspondent à des communautés de nœuds dans le multigraphe de co-appartenance. Ce travail

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

a fait l'objet d'une publication dans la conférence SAC 2020 (Benabdelkrim et al., 2020b).

— Nous montrons, dans le chapitre 5, comment dériver un jeu de données numériques à partir des similarités structurelle et sémantique entre les couches du multigraphe de co-appartenance. Nous utilisons notamment le modèle de traitement du langage naturel BERT (Devlin et al., 2019) qui permet de représenter des séquences de mots dans un espace vectoriel pour calculer les similarités sémantiques entre les attributs du multigraphe. Les motifs d'intervalles (Kaytoue et al., 2011) d'un jeu de données numériques permettent d'obtenir les ensembles d'objets qui ont des valeurs d'attributs proches appartenant à un même intervalle. Pour un multigraphe de co-appartenance, les motifs d'intervalles du jeu de données numériques qui lui est associé correspondent à des communautés de nœuds. Cette méthode a été présentée à la conférence CanadianAI 2021 (Benabdelkrim et al., 2021).

(4) Nous menons des expériences sur plusieurs exemples de multigraphes de co-appartenance. Nous nous intéressons particulièrement au multigraphe de co-appartenance des utilisateurs Twitter aux listes Twitter dans lequel nous sélectionnons trois sous-graphes correspondant aux domaines d'intérêt : la "Scène politique française", l'"Intelligence artificielle" et l'"Impact social des organisations à but non lucratif" (Section 3.5). Nous utilisons des algorithmes de fouille de données pour extraire les communautés de ces sous-graphes. Nous comparons par la suite nos résultats à ceux des méthodes de l'état de l'art à l'aide de métriques d'exceptionnalité sémantique : le TF-IDF et la divergence de Kullback-Leibler, mais aussi en se comparant à la vérité-terrain des communautés quand celle-ci est disponible (Sections 4.4 et 5.4). Nous remarquons que la prise en compte de la sémantique améliore considérablement les résultats des algorithmes de détection de communautés dans les multigraphes de co-appartenance.

Dans la suite du rapport, nous développons au chapitre 2 un état de l'art des méthodes de détection de communautés dans les multigraphes. Dans le chapitre 3, nous définissons formellement les multigraphes de co-appartenance utilisés dans la suite, puis nous expliquons le fonctionnement des deux procédures permettant de délimiter des sous-graphes d'intérêt. Au chapitre 4, nous rappellons les concepts de base de l'AFC et nous montrons comment utiliser l'extraction de motifs symboliques pour la détection de communautés dans les multigraphes de co-appartenance. Enfin dans le dernier chapitre (chap. 5), nous établissons le lien entre l'extraction des motifs d'intervalles d'un jeu de données numériques et les communautés d'un multigraphe de co-appartenance.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Chapitre 2

État de l'art

Durant la deuxième moitié du siècle dernier, plusieurs travaux de recherche se sont intéressés à l'analyse de la structure des graphes en proposant des modèles de graphes, des méthodes de détection de communautés ou encore plus récemment des méthodes de plongement de graphes dans des espaces vectoriels. Ces différentes approches permettent de simplifier la représentation et la compréhension des graphes complexes qui contiennent un très grand nombre de nœuds et un nombre encore plus grand de liens. Ainsi, plusieurs représentations condensées peuvent être réalisées pour un graphe : il peut être représenté par un modèle et les valeurs des paramètres qui donnent le meilleur ajustement ; ou en utilisant les communautés, par un hypergraphe dans lequel les nœuds correspondent aux communautés ; ou encore en utilisant les méthodes de plongement, par les coordonnées de ses nœuds dans un espace euclidien calculées de sorte à préserver la structure de connectivité du graphe.

La littérature scientifique est riche en méthodes d'analyse de la structure des graphes. Mais, les graphes sont des objets mathématiques divers qui peuvent être de différents types. Ils sont potentiellement pondérés : avec des poids numériques associés à leurs liens. Ils peuvent aussi être attribués quand ils possèdent des informations numériques, catégoriques ou textuelles décrivant les éléments du graphe. Les graphes sont dits multicouches quand plusieurs relations différentes entre paires de nœuds sont modélisées dans le graphe. Et, les graphes peuvent être dynamiques si des pas temporels sont associés aux liens du graphe de sorte à donner les temps d'apparition et de disparition des liens.

Parmi ces types de graphes spécifiques, les graphes de co-appartenance, qui nous intéressent, modélisent les relations d'appartenance commune de paires d'entités à des groupes. Ces graphes sont dérivés de données de co-appartenance $(\mathcal{V}, \mathcal{K}, \mathcal{L}, \mathbf{v}, \mathbf{k}, \mathbf{t})$ définies par trois ensembles et trois applications :

- un ensemble d'entités $\mathcal{V}$
- un vocabulaire de mots $\mathcal{K}$
- un ensemble de groupes $\mathcal{L}$ dont chacun regroupe un sous-ensemble d'entités $\mathcal{V}$ et possède une description sémantique avec des mots du vocabulaire $\mathcal{K}$
- $\mathbf{v} : \mathcal{L} \to \mathcal{P}(\mathcal{V})$ associe à un groupe de $\mathcal{L}$ un sous-ensemble d'entités de $\mathcal{P}(\mathcal{V})$
- $\mathbf{k} : \mathcal{L} \to \mathcal{P}(\mathcal{K})$ associe à un groupe de $\mathcal{L}$ une description de $\mathcal{P}(\mathcal{K})$
- $\mathbf{t} : \mathcal{L} \to \mathbb{N}$ associe à un groupe de $\mathcal{L}$ une date de création

Les relations d'appartenance commune sont transitives : s'il y en a une entre les entités

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

a et b et les entités b et c, alors il y a nécessairement une relation d'appartenance commune entre les entités a et c. Les entités appartenant à un même groupe forment une clique dans le graphe de co-appartenance, et les liens d'une couche L sont définis implicitement par l'ensemble des nœuds $\mathbf{v}(L)$ de celle-ci. Par ailleurs, à ces groupes sont souvent associées des descriptions textuelles (l'application **k** dans les données de co-appartenance) qui permettent de caractériser les relations d'appartenance commune représentées par les liens dans le graphe de co-appartenance. Dans certains cas, une information temporelle sur les dates de créations des groupes est aussi présente dans les données de co-appartenance (l'application **t** dans les données de co-appartenance), il est alors possible de définir un graphe de co-appartenance dynamique.

Ces graphes peuvent être vus comme des multigraphes dans lesquels chaque groupe est représenté dans une couche à laquelle sont associés des attributs textuels issus de la description du groupe. Ce type de graphe est souvent rencontré dans les réseaux sociaux avec par exemple la co-appartenance des utilisateurs d'un réseau social à des listes ou des groupes créés par d'autres utilisateurs comme les listes Twitter ou les groupes Facebook. Les autres cas d'usage de ce type de graphes sont la co-participation à des événements (conférences, séminaires ...), ou encore la co-publication ou la co-citation d'articles scientifiques.

Dans cette thèse, nous nous intéressons à l'analyse de ces multigraphes de co-appartenance en identifiant des sous-graphes d'intérêt dans lesquels nous détectons des communautés de nœuds homogènes structurellement et sémantiquement. Les sous-graphes d'intérêt permettent de sélectionner une région du multigraphe de co-appartenance qui correspond aux préférences de l'utilisateur de l'outil d'analyse de ces graphes. Cette étape permet de réduire la taille, souvent très grande, des multigraphes de co-appartenance et de pouvoir en extraire les communautés en un temps raisonnable. La structure en communautés correspond au partitionnement des nœuds du réseau en groupes de sorte que les nœuds d'un même groupe soient densément reliés entre eux et que les liens entre groupes soient clairsemés (Fortunato, 2010). Cette structure en modules se rencontre dans plusieurs organisations sociales comme dans les groupes d'amis, de collègues, mais aussi dans les groupes virtuels d'internautes qui partagent les mêmes centres d'intérêt, et nous la retrouvons naturellement dans les graphes de co-appartenance. Le partitionnement de nœuds d'un graphe en communautés a des applications dans plusieurs autres domaines comme la biologie (interactions protéine-protéine (Chen and Yuan, 2006)), l'ingénierie (réseaux de transport aérien (Guimerà et al., 2005)) et l'économie (modèles de réseaux pour les marchés (Jackson, 2011)). Il est important de noter qu'il n'y a pas de définition unique dans la littérature du concept de communauté dans un graphe. Les chercheurs qui se sont intéressés à cette problématique ont proposé diverses définitions en fonction du type de graphes auquel ils s'intéressent et de leur formalisation de la notion de communauté. En plus de la structure en communautés, les réseaux ont souvent une structure hiérarchique. En d'autres termes, chaque communauté d'un réseau peut elle-même être partitionnée en d'autres sous-communautés (Clauset et al., 2008). La recherche de communautés peut alors être utilisée pour construire des vues à différentes échelles d'un graphe. Dans la vue macroscopique, chaque communauté est une partie indépendante dans la structure du graphe et est en relation plus ou moins robuste avec les autres communautés.

Dans ce chapitre, nous présentons une synthèse de l'état de l'art des méthodes de détection de communautés dans les multigraphes et discutons de leur possible application au cas des multigraphes de co-appartenance. Nous commençons par présenter les différentes

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

approches pour la sélection de sous-graphes d'intérêt. Puis, nous présentons les méthodes globales de détection de communautés dans les multigraphes. Ces méthodes peuvent être classées en trois familles : les méthodes qui projettent les multigraphes sur une seule couche et utilisent des algorithmes de détection de communautés dans les graphes simples, les méthodes qui obtiennent les communautés du multigraphe en agrégeant les partitionnements de nœuds calculés couche par couche, et les méthodes qui s'appliquent directement au graphe multicouche. Puis, nous expliquons comment évaluer la qualité d'un partitionnement de nœuds et comparer les performances de différents algorithmes de détection de communautés.

2.1 Notions de graphe, multigraphe et graphe de co-appartenance

A partir des données de co-appartenance définies précédemment, il est possible de dériver trois types de graphes : le graphe de co-appartenance, le multigraphe de co-appartenance et le multigraphe de co-appartenance dynamique.

Nous commençons par donner une définition d'un graphe simple et de sa matrice d'adjacence qui permet de le représenter. Puis, nous définissons les graphes et multigraphes de co-appartenance. Par ailleurs, certaines définitions associées aux graphes sont données dans l'annexe A.

Définition 1 (Graphe). ***Un graphe*** $G = (\mathcal{V}, E)$ *défini par un ensemble de nœuds* $\mathcal{V}$ *et un ensemble de liens* $E \subseteq \mathcal{V} \times \mathcal{V}$ *entre ces nœuds. Il peut être représenté par une* ***matrice d'adjacence*** A *de dimension* $|\mathcal{V}| \times |\mathcal{V}|$ *telle que, pour toute paire de nœuds* $V_i, V_j \in \mathcal{V}$,

$$A_{ij} = \begin{cases} 1 & si\ (V_i, V_j) \in E \\ 0 & sinon \end{cases}$$

Souvent, les liens d'un graphe possèdent un poids $p_{ij} \in \mathbb{R}$ qui représente leur importance relative. Nous parlons alors de graphe pondéré :

Définition 2 (Graphe pondéré). ***Un graphe pondéré*** $G = (\mathcal{V}, E, \mathbf{p_G})$ *est un graphe simple possédant des poids* $p_{ij} \in \mathbb{R}$ *associés à ses liens* $e_{ij} = (V_i, V_j) \in E$. *L'application* $\mathbf{p_G} : E \to \mathbb{R}$ *retourne le poids associé à chaque lien du graphe. La matrice d'adjacence d'un graphe pondéré est telle que* $A_{ij} = p_{ij}$.

Les données de co-appartenance peuvent être modélisées dans un graphe pondéré. Pour cela, les entités sont représentés par des nœuds et le poids d'un lien entre deux nœuds est le nombre de groupes dans lesquels les deux entités correspondantes apparaissent ensemble. Mais, ce type de graphes a de propriétés spécifiques par rapport à un graphe pondéré quelconque. En effet, la relation de co-appartenance est transitive entre toutes les paires d'entités d'un groupe. Il en résulte que le graphe pondéré modélisant cette relation est composé de cliques : c'est-à-dire de sous-graphes complètement connectés.

Définition 3 (Graphe de co-appartenance pondéré). ***Un graphe de co-appartenance pondéré*** $\mathcal{G}$ *est un graphe pondéré dérivé des données de co-appartenance. Il est défini par* $\mathcal{G} = (\mathcal{V}, E, \mathbf{p}_e)$ *où :*

- *l'ensemble de nœuds* $\mathcal{V}$ *est l'ensemble des entités*
- *l'ensemble des liens* $E \subseteq \mathcal{V} \times \mathcal{V}$ *représente la relation d'appartenance commune*

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

— *l'application* $\mathbf{p}_e : E \rightarrow \mathbb{N}^\star$ *qui associe aux liens un poids égal au nombre de groupes dans lequel ils apparaissent ensemble*

Par ailleurs, les données de co-appartenance contiennent aussi une information sémantique liée aux groupes $\mathcal{L}$. La prise en compte de cette information dans la modélisation permet d'associer des attributs textuels aux liens du graphe. Il en résulte un graphe de co-appartenance attribué $\mathcal{G}$.

Définition 4 (Graphe de co-appartenance attribué)**.** ***Un graphe de co-appartenance attribué*** $\mathcal{G}_a$ *est défini par* $\mathcal{G}_a = (\mathcal{V}, E, \mathbf{p}_e, \mathbf{k}_e)$*. C'est un graphe de co-appartenance pondéré avec des attributs textuels associés aux liens avec l'application* $\mathbf{k}_e) : E \rightarrow \mathcal{P}(\mathcal{P}(\mathcal{K}))$*.* $\mathbf{k}_e)$ *associe à chaque lien des documents textuels : chacun contient la description d'un groupe commun aux entités représentées par les nœuds extrémités du lien.*

Le cadre des graphes simples peut s'avérer insuffisant quand les relations à modéliser sont complexes et/ou de différentes natures. Dans ce cas, les graphes multicouches (ou multigraphes) peuvent apporter une solution pour modéliser plus fidèlement ces systèmes : ils permettent de séparer différents types de liens dans des couches indépendantes. Kivela et al. (2014) ont proposé une revue de la littérature qui synthétise les différents cadres théoriques des graphes multicouches et les outils développés pour leur analyse.

Définition 5 (Multigraphe)**.** *Un multigraphe* $G = (\mathcal{V}, E, \mathcal{L})$ *est défini par un ensemble de nœuds* $\mathcal{V}$ *et un ensemble de couches* $\mathcal{L}$*, les liens* $e \in E$ *sont des triplets de* $\mathcal{V} \times \mathcal{V} \times \mathcal{L}$ *qui représentent les nœuds extrémités du lien et la couche dans laquelle il apparaît.*

Pour les deux modélisation des données de co-appartenance précédentes $\mathcal{G}$ et $\mathcal{G}_a$ des définitions 3 et 4, toutes les relations sont considérées être les mêmes et, par conséquent, les graphes de co-appartenance ne contiennent qu'un seul type de lien. Ces approches ont l'inconvénient de produire des graphes très denses dans lesquels la majorité des nœuds est connectée. Aussi, dans le cas du graphe de co-appartenance attribué, les attributs textuels présents sur les liens sont très lourds : chacun contient plusieurs documents qui correspondent aux descriptions des groupes communs aux entités représentées par les extrémités du lien. Ces deux inconvénients cumulés font que la détection de communautés dans un graphe de co-appartenance attribué est une tâche difficile à cause de la confusion créée par la grande densité du graphe ; c'est aussi une tâche coûteuse en temps de calcul à cause de la grande taille des attributs textuels associés aux liens.

Une alternative est alors de modéliser les données de co-appartenance avec un multigraphe dans lequel les liens sont de différents types : chaque type correspond à un groupe en particulier et est présent dans une couche dédiée du multigraphe. Comme les relations de co-appartenance sont transitives, les couches du multigraphe forment des cliques dans lesquelles toutes les paires de nœuds sont connectées. Les attributs textuels sont ici associés aux couches du multigraphe et ne sont composés que d'un seul document qui est l'information sémantique associée au groupe correspondant. Le multigraphe de co-appartenance ainsi construit a les propriétés de couplage diagonal, indépendant des nœuds et catégorique.

Définition 6 (Multigraphe de co-appartenance)**.** ***Un multigraphe de co-appartenance*** $\mathcal{G}_{ma}$ *est un multigraphe dérivé des données de co-appartenance. Il est défini par* $\mathcal{G}_{ma} = (\mathcal{V}, E, \mathcal{L}, \mathbf{k_l})$ *où :*

— *l'ensemble de couches* $\mathcal{L}$ *est l'ensemble des groupes*

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

— $\mathbf{k_l} : \mathcal{L} \rightarrow \mathcal{P}(\mathcal{K})$ *associe la description du groupe à la couche correspondante*
— *les liens* $\mathcal{V} \times \mathcal{V} \times \mathcal{L}$ *représentent l'appartenance commune d'une paire d'entités dans un groupe spécifique*

Souvent, une information temporelle correspondant à la date de création des groupes est aussi disponible dans les données de co-appartenance. Elle peut être utilisée pour définir un multigraphe de co-appartenance dynamique dans lequel les couches sont rajoutées au fur et à mesure que les groupes correspondants sont créés.

Définition 7 (Multigraphe de co-appartenance dynamique)**.** ***Un multigraphe de co-appartenance dynamique*** $\mathcal{G}_{mad} = (\mathcal{V}, E, \mathcal{L}, \mathbf{k_l}, \mathbf{t_l})$ *est un multigraphe de co-appartenance dans lequel les couches ont un ordre temporel représenté par l'application* $\mathbf{t_l} : \mathcal{L} \rightarrow \mathbb{N}$ *qui associe à chaque couche un pas temporel fonction de la date de création du groupe correspondant.*

2.2 Sélection de sous-graphes d'intérêt dans des multigraphes

Parmi les caractéristiques des multigraphes de co-appartenance, et particulièrement ceux qui sont issus des réseaux sociaux, nous pouvons citer leur très grande taille et le fait qu'ils ne sont pas nécessairement homogènes et ciblés sur le domaine qui intéresse l'analyste utilisateur de l'outil. Une première étape est donc de délimiter les frontières d'un sous-graphe qui correspond au domaine d'intérêt de l'utilisateur. Les méthodes locales de détection de communautés sont appropriées pour cette tâche. Ces méthodes utilisent un ou plusieurs nœuds initiaux appelés graines et construisent la communauté grâce à une procédure d'expansion. L'identification de ces sous-graphes d'intérêt dans un multigraphe peut alors se faire en utilisant des méthodes de détection de communautés locales dans les graphes simples sur le graphe agrégé obtenu par projection du multigraphe sur une seule couche (Section 2.2.1) ou directement sur le multigraphe (Section 2.2.2).

2.2.1 Sous-graphe d'intérêt à partir du graphe agrégé

Pour analyser de grands graphes, une stratégie consiste à construire un sous-graphe à partir de nœuds "graines" initialement identifiés par l'utilisateur. C'est le cas de la méthode présentée dans (Chen et al., 2009) qui identifie un sous-graphe d'intérêt D à l'aide de son cœur C composé des nœuds "graines" et d'une bordure B contenant les nœuds voisins des nœuds "graines". Un test de connectivité est appliqué sur les nœuds adjacents à la bordure pour éventuellement être rajoutés à la communauté s'ils y sont fortement connectés. Pour cela, les auteurs estiment si l'ajout du nœud entraînera un gain pour le score de la communauté. Celui-ci est calculé comme le quotient du degré moyen interne de la communauté sur le degré moyen externe de la bordure. Plus exactement, les auteurs proposent le score L tel que :

$$L = \frac{L_{in}}{L_{ex}} \quad \text{où } L_{in} = \frac{\sum_{i \in D} IK_i}{|D|} \quad \text{et } L_{ex} = \frac{\sum_{j \in B} EK_j}{|B|}$$

IK_i dénote le nombre de liens entre le nœud i et les nœuds de la communauté D et EK_j représente le nombre de liens entre le nœud j et les nœuds externes à la communauté. Par la suite, la bordure B de la communauté est filtrée en mesurant l'effet de la suppression de ses nœuds sur le score L.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Une autre méthode (Huang et al., 2011) utilise une approche semblable pour détecter des communautés locales avec différents niveaux de résolution à partir d'un nœud source. Ils définissent une mesure de similarité structurelle $s(u,v)$ entre les paires (u,v) de nœuds adjacents telle que :

$$s(u,v) = \frac{\sum_{x \in (\mathcal{T}(u) \bigcap \mathcal{T}(v))} w(u,x).w(v,x)}{\sqrt{\sum_{x \in \mathcal{T}(u)} w^2(u,x)} \sqrt{\sum_{x \in \mathcal{T}(v)} w^2(v,x)}}$$

il en déduisent une mesure de qualité qu'ils appellent raideur (tightness) et notée $T(C)$ pour un sous-graphe C qu'ils cherchent à maximiser pour délimiter la communauté locale. La mesure T est définie de sorte à maximiser la similarité interne S_{in}^C somme des similarités des paires de nœuds adjacents interne au sous-graphe et à minimiser la similarité externe S_{out}^C somme des similarités des paires de nœuds adjacents telles que un nœud est à l'intérieur du sous-graphe et l'autre est à l'extérieur. La raideur T s'exprime comme suit :

$$T = \frac{S_{in}^C}{S_{in}^C + S_{out}^C}$$

Elle est utilisée pour définir le gain paramétré en raideur $\tau_C^\alpha(a)$ obtenu en ajoutant un nœud a à un sous-graphe C :

$$\tau_C^\alpha(a) = \frac{S_{out}^C}{S_{in}^C} - \frac{\alpha S_{out}^a - S_{in}^a}{2S_{in}^a}$$

où S_{in}^a est la somme des similarités entre le nœud a et ses nœuds adjacents dans C et S_{out}^a est la somme des similarités entre le nœud a et ses enœuds adjacents à l'extérieur de C. Le paramètre α contrôle la taille de la communauté retournée et permet d'explorer des communautés locales à différentes échelles. Une procédure itérative permet de détecter la communauté locale à partir des nœuds graines en rajoutant à chaque étape le nœud qui permet d'avoir le plus grand gain en raideur $\tau_C^\alpha(a)$.

De même, Li et al. (2015) utilisent une stratégie d'expansion à partir d'un ensemble de nœuds graines. Leur méthode se base sur l'analyse du spectre local du graphe obtenu avec des simulations de marches aléatoires qui partent des nœuds "graines". La procédure proposée est une procédure itérative qui, à chaque étape, cherche le sous-ensemble de nœuds qui pointent dans la direction la plus proche de celle des nœuds graines dans l'espace engendré par les vecteurs de probabilité des marches aléatoires. Après chaque étape, les nœuds choisis sont rajoutés aux nœuds graines et de nouvelles marches aléatoires sont simulées. Une condition d'arrêt permet de stopper la procédure et de retourner la communauté locale finale. Cette condition est basée sur une mesure de qualité d'une communauté appelée conductance et qui est une mesure à minimiser : la condition est vérifiée quand la conductance commence à croître pour la première fois.

D'autres méthodes pour la sélection de sous-graphes d'intérêt détectent automatiquement les nœuds "graines" des communautés. Fagnan et al. (2014) proposent de choisir les nœuds de degré le plus grand comme nœuds "graines". Ils utilisent une nouvelle métrique

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

pour évaluer la qualité d'un sous-graphe et pour délimiter les frontières de la communauté locale. Cette métrique est basée sur le nombre de triades (cliques de 3 nœuds) à l'intérieur et en bordure de la communauté. Par la suite, cette même métrique est utilisée pour identifier les rôles occupés par certains nœuds dont l'appartenance à la communauté locale est indécise : ce sont soit des nœuds aberrants qui n'appartiennent à aucune communauté du graphe, ou bien des nœuds passerelles qui font le lien entre la communauté locale et d'autres communautés. Plus récemment, (Whang et al., 2015) ont présenté deux stratégies pour déterminer un ensemble de nœuds "graines" dispersés dans un graphe. La première stratégie utilise un algorithme de partitionnement de graphe de faible complexité temporelle et choisit les nœuds centraux dans les partitions comme nœuds "graines". La deuxième approche pour le choix des nœuds "graines" est similaire à celle utilisée dans (Fagnan et al., 2014) avec une condition de grande dispersion dans le graphe supplémentaire. Ensuite, les auteurs proposent un algorithme d'expansion utilisant la distribution stationnaire d'une marche aléatoire spécifique appelée PageRank personnalisée. Ma and Fan (2020) ont proposé un algorithme qui détecte les cliques dans le graphe et les considère comme cœurs des communautés ; puis, l'optimisation d'une mesure de qualité permet de délimiter la bordure de la communauté autour de la clique centrale. Dans une première étape, les cliques du graphe sont détectées. Ensuite, une mesure de densité locale est calculée pour tous les nœuds du graphe. Le nœud de densité maximal est pris comme sous-graphe de départ de la procédure d'expansion. La mesure de qualité utilisée est définie pour les sous-graphes de sorte à maximiser la connectivité à l'intérieur du sous-graphe et à la minimiser à l'extérieur. Parmi les voisins du nœud de départ, la procédure choisit celui qui maximise l'augmentation de la mesure de qualité pour le sous-graphe. Toutes les cliques qui contiennent le voisin choisi sont alors ajoutées au sous-graphe.

2.2.2 Sous-graphes d'intérêt dans des multigraphes

L'algorithme MLLCD (Interdonato et al., 2017) est une méthode locale pour la détection locale de communautés dans des multigraphes à partir d'un ensemble de nœuds graines choisis par l'analyste. Les auteurs définissent l'enveloppe S d'une communauté C comme l'ensemble de nœuds externes à la communauté mais ayant un lien avec au moins un de ses membres. Ils définissent aussi la bordure B de la communauté comme le sous-ensemble de ses nœuds qui sont reliés à l'enveloppe S. L'algorithme proposé est basé sur l'optimisation d'une des trois fonctions fitness définies par les auteurs. La première fonction permet de maximiser le nombre de liens à l'intérieur de la communauté et de minimiser le nombre de liens périphériques à travers toutes les couches du graphe. Elle utilise une pondération qui permet de donner plus ou moins d'importance dans le calcul de la fonction aux différentes couches du multigraphe. Elle est notée LC_1 et est définie pour une communauté C par $LC_1 = LC_1^{int}/LC_1^{ext}$ où :

$$LC_1^{int} = \frac{1}{|C|} \sum_{v \in C} \sum_{L_i \in \mathcal{L}} w_i |E_i^C(v)| \quad \text{et} \quad LC_1^{ext} = \frac{1}{|B|} \sum_{v \in B} \sum_{L_i \in \mathcal{L}} w_i |E_i^B(v)| \tag{2.1}$$

telle que $\sum_{L_i \in \mathcal{L}} w_i = 1$ et avec $E_i^C(v)$ l'ensemble de liens reliant le nœud v à des nœuds de la communauté C dans la couche L_i ; et $E_i^B(v)$ l'ensemble de liens reliant le nœuds v à des nœuds de l'enveloppe S de la communauté.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

La deuxième fonction fitness à maximiser utilise la similarité entre les voisinages dans une même couche des nœuds pour trouver la communauté locale. Elle est définie par $LC_2 = LC_2^{int}/LC_2^{ext}$ avec :

$$LC_2^{int} = \frac{1}{|C|}\sum_{v\in C}\sum_{L_i\in\mathcal{L}}\sum_{\substack{(u,v)\in E_i^C \\ \wedge u\in C}} sim_i(u,v) \quad \text{et} \quad LC_2^{ext} = \frac{1}{|B|}\sum_{v\in B}\sum_{L_i\in\mathcal{L}}\sum_{\substack{(u,v)\in E_i^B \\ \wedge u\in S}} sim_i(u,v) \tag{2.2}$$

où $sim_i(u,v)$ mesure la similarité entre le voisinage des nœuds u et v dans la couche L_i : par exemple, l'indice de Jaccard entre l'ensemble des voisins de u et celui de v.

La troisième fonction fitness proposée dans (Interdonato et al., 2017) est similaire à LC_2 mais elle calcule les similarités entre nœuds à travers toutes les couches du multigraphe au lieu de les calculer couche par couche. Elle est notée LC_3 et c'est le quotient de LC_3^{int} sur LC_3^{ext} avec :

$$LC_3^{int} = \frac{1}{|C|}\sum_{u,v\in C}\sum_{\substack{L_i,L_j\in\mathcal{L} \\ \wedge u\in V_i, v\in V_j}} sim_{i,j}(u,v) \quad \text{et} \quad LC_3^{ext} = \frac{1}{|B|}\sum_{v\in B,u\in S}\sum_{\substack{L_i,L_j\in\mathcal{L} \\ \wedge u\in V_i, v\in V_j}} sim_{i,j}(u,v) \tag{2.3}$$

V_i dénote les nœuds de la couche $L_i \in \mathcal{L}$. Comme pour LC_2, $sim_{i,j}(u,v)$ correspond à la similarité topologique entre les voisins de u dans L_i et les voisins de v dans L_j.

Après le choix d'une des fonctions fitness ci-dessus, une formule de mise à jour est appliquée pour calculer le gain de l'ajout d'un nœud de la couverture S à la communauté C. L'algorithme proposé commence avec les nœuds "graines" puis, à chaque étape, le nœud permettant le plus grand gain pour la fonction fitness est rajouté à la communauté et ses voisins sont rajoutés à la couverture. L'algorithme s'arrête quand aucune amélioration de la fonction fitness n'est possible.

L'algorithme ACLcut (Jeub et al., 2017) utilise des marches aléatoires et considère les communautés comme des ensembles de nœuds dans lesquels les marches aléatoires se retrouvent piégées longuement. Les auteurs introduisent deux stratégies de marches aléatoires pour les graphes multicouches et montrent qu'elles engendrent des communautés différentes. Pour définir ces marches aléatoires, les auteurs utilisent le tenseur d'adjacence A dans lequel $A_{j\beta}^{i\alpha}$ correspond au poids du lien entre le nœud i de la couche α et le nœuds j de la couche β. La première stratégie de marche aléatoire est une stratégie classique semblable à celle utilisée dans les graphes simples. Sa probabilité de transition entre le nœuds i de la couche α et le nœud j de la couche β est donnée par :

$$P_{j\beta}^{i\alpha} = \frac{A_{j\beta}^{i\alpha}}{\sum_{k\gamma\in V_m} A_{k\gamma}^{i\alpha}} \tag{2.4}$$

La deuxième stratégie est la marche aléatoire relaxée dans laquelle un paramètre $r \in [0,1]$ contrôle la probabilité de la marche aléatoire de rester dans la couche ou d'en sortir. Elle est définie par la probabilité de transition suivante :

$$P_{j\beta}^{i\alpha} = (1-r)\delta(\alpha,\beta)\frac{A_{j\alpha}^{i\alpha}}{\sum_{k\in V} A_{k\alpha}^{i\alpha}} + r\frac{A_{j\beta}^{i\beta}}{\sum_{k\in V,\gamma\in\mathcal{L}} A_{k\gamma}^{i\gamma}} \tag{2.5}$$

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Ces marches aléatoires permettent de définir la communauté locale autour d'un nœud "graine". Celui-ci est le point de départ des marche aléatoires et la communauté correspond au sous-graphe dans lequel la marche se retrouve piégée pour un grand nombre d'itérations. Les deux méthodes locales MLLCD et ACLcut nécessitent le choix initial de nœuds centraux par l'analyste.

Par ailleurs, Mux-LICOD (Hmimida and Kanawati, 2015) est aussi un algorithme de détection de communautés locales dans les graphes multicouches mais il a la particularité de choisir automatiquement les nœuds "graines" en fonction de leur degré dans le multigraphe. Pour cela, les auteurs proposent une généralisation du degré et du voisinage d'un nœud au cas des graphes multicouches. Ils font le choix de nœuds "graines" dont le degré est supérieur au degré de la majorité de leurs voisins. Par la suite, une fois les "graines" fixées, l'attribution des communautés aux nœuds se fait en fonction des longueurs des plus courts chemins qui les relient aux nœuds "graines".

Discussion

L'intérêt des méthodes locales pour la détection de communautés dépend du besoin de l'utilisateur de la méthode. Elles sont très utiles si l'utilisateur n'est intéressé que par une région particulière dans le graphe. Mais, elles nécessitent plus de ressources de calcul que les méthodes globales pour trouver un partitionnement de tous les nœuds d'un graphe car les méthodes locales doivent être exécutées à de multiples reprises avec différents nœuds graines pour obtenir un partitionnement complet du graphe. Pour nous, ces méthodes nous permettent de sélectionner le sous-graphe qui correspond au domaine d'intérêt de l'analyste à partir d'un ensemble de nœuds graines. Cette étape de sélection de sous-graphe d'intérêt sera développée au chapitre 3 dans lequel nous comparerons les résultats de deux algorithmes que nous présenterons à ceux de certaines méthodes de la section présente. Dans cette comparaison, nous nous baserons sur les temps d'exécution des algorithmes et sur la taille et l'homogénéité des sous-graphes obtenus. Nous verrons, en particulier, que les méthodes qui sélectionnent automatiquement les nœuds graines ne sont pas adaptées pour l'extraction de sous-graphes d'intérêt à cause du coût de calcul que la sélection automatique nécessite.

Mais, avant de pouvoir utiliser ces méthodes, il faut d'abord représenter le multigraphe de co-appartenance $\mathcal{G}_{ma}$ de la définition 6 comme un graphe simple pondéré ou comme un multigraphe non attribué pour correspondre aux types de graphes pris en compte par les méthodes présentées. Le graphe simple pondéré est obtenu en agrégeant les différentes couches en une seule et en associant, à chaque lien, un poids entier correspondant au nombre de couches dans lesquelles il apparaît. Le multigraphe non attribué est obtenu en mettant de côté les attributs textuels associés aux couches du multigraphe de co-appartenance.

2.3 Détection de communautés dans des multigraphes

On présente dans la suite les approches globales les plus utilisées pour la détection de communautés dans les multigraphes. Pour cela, nous utilisons principalement deux revues de littérature publiées récemment (Hanteer et al., 2019, Huang et al., 2021).

Tout d'abord, les premières méthodes proposées pour trouver des communautés dans des multigraphes se basent sur la détection de communautés dans les graphes simples. Pour

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

cela, elles nécessitent une transformation du multigraphe en un graphe simple pondéré dans lequel les poids des liens sont des entiers définis comme le nombre de couches dans lequel le lien existe dans le multigraphe (Kuncheva and Montana, 2015) (Section 2.3.1). Une alternative est de trouver les partitionnements des nœuds couche par couche en utilisant une méthode de détection de communautés dans des graphes simples puis d'agréger les différents partitionnements de façon à obtenir des communautés pour le multigraphe (Section 2.3.2). Enfin, d'autres algorithmes détectent directement les communautés dans le graphe multicouche (Section 2.3.3).

2.3.1 Méthodes basées sur la découverte de communautés dans des graphes simples

2.3.1.1 Méthodes hiérarchiques

Méthodes divisives

Les méthodes divisives cherchent un partitionnement des nœuds du graphe de sorte que la somme des poids des liens entre partitions ou "cut size" soit minimale. Ce sont des méthodes adaptées aux graphes pondérés sur les liens et basées sur la minimisation d'un critère mesurant le poids des liens inter-communautés. Parmi ces critères, nous pouvons citer le "cut size" (Kernighan and Lin, 1970) ou encore la conductance (Flake et al., 2003) et le "normalized cut" (Jianbo Shi and Malik, 2000).

Ces méthodes ont cependant certains inconvénients : la recherche d'un partitionnement optimisant un de ces critères est un problème NP-difficile (Síma and Schaeffer, 2005). Il existe cependant des méthodes heuristiques pour obtenir des solutions approximatives mais elles restent coûteuses en terme de calcul, les algorithmes dans (Kernighan and Lin, 1970) et (Jianbo Shi and Malik, 2000) ont respectivement des temps d'exécution en $O(n^3)$ et $O(mn)$; de plus, ces méthodes nécessitent le choix préalable du nombre de communautés ce qui oblige l'analyste à tester plusieurs valeurs et à comparer les partitionnements obtenus en utilisant d'autres mesures de qualité.

Une des méthodes divisives d'usage le plus courant dans la littérature scientifique est l'algorithme Girvan-Newman (Newman and Girvan, 2004). Cet algorithme a l'avantage de ne pas nécessiter de connaissance préalable sur le nombre de communautés. Les auteurs déterminent l'importance d'un lien en fonction de sa position dans le graphe et plus précisément de sa centralité. Ils proposent trois définitions pour la centralité d'un lien : la centralité des chemins les plus courts, la centralité avec marches aléatoires et la centralité avec flux de courant. La première de ces définitions correspond au ratio de participation du lien aux plus courts chemins entre paires de nœuds. La deuxième définition est basée sur la probabilité de passage d'une marche aléatoire par le lien. Cette définition est équivalente à la dernière qui considère le graphe comme un réseau de résistances et calcule la centralité d'un lien comme la moyenne du courant conduit par le lien quand une différence de voltage est appliquée à toutes les paires de nœuds du réseau. Par la suite, les liens qui possèdent les plus grandes valeurs de centralité – qui se trouvent dans un grand nombre de plus courts chemins entre paires de nœuds – sont supprimés de façon à séparer le graphe en communautés disjointes. Les auteurs ont montré que le calcul de la centralité par participation aux plus courts chemins était le plus rapide parmi les trois définitions proposées. Et, les expériences qu'ils ont menées concluent que cette définition de la centralité donne aussi les meilleurs résultats.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

La méthode de Newman and Girvan (2004) a tout de même une exécution lente en $O(nm^2)$ notamment parce que les centralités doivent être recalculées après chaque suppression de liens. Ceci limite l'utilisation de l'algorithme à des graphes peu denses et avec un nombre de nœuds de l'ordre de 10^4.

Plusieurs améliorations de l'algorithme de (Newman and Girvan, 2004) ont été développées dans le but d'accélérer son exécution. Par exemple, Wilkinson and Huberman (2004) limitent le calcul de la centralité à un sous-ensemble des nœuds du graphe, et Rattigan et al. (2007) utilisent une approximation de la mesure de centralité et proposent un algorithme avec une complexité temporelle en $O(m)$. D'autres versions ont aussi été proposées pour permettre d'assigner plusieurs communautés à un seul nœud (Gregory, 2007), chose que ne permettait pas la méthode d'origine.

Méthodes agglomératives

Les méthodes agglomératives partent de la partition discrète, dans laquelle chaque nœud constitue une communauté, puis elles fusionnent les communautés les plus proches de manière itérative. Pour ce faire, les liens du graphe sont mis de coté initialement. Par la suite, les similarités entre paires de nœuds sont calculées et les liens sont rajoutés un à un en commençant par les paires de nœuds ayant les plus grandes valeurs de similarité. La procédure peut être arrêtée à n'importe quel moment et les composantes connexes résultantes sont considérées comme les communautés du graphe. Alternativement, la totalité de la progression de la construction du graphe en commençant par le graphe ne contenant aucun lien et jusqu'à arriver au graphe complet peut être représentée par un dendrogramme. Des coupures horizontales peuvent alors être utilisées sur le dendrogramme pour obtenir les communautés. Une des méthodes agglomératives les plus utilisées est Breiger et al. (1975). Mais, dans un cadre supervisé dans lequel la vérité terrain sur les communautés est connue, les communautés retournées par cette méthode diffèrent significativement des vraies communautés dans beaucoup d'exemples de jeux de données (Newman and Girvan, 2004). Plus particulièrement, les méthodes agglomératives ont tendance à détecter les cœurs des communautés mais échouent à détecter leur périphérie (Fortunato, 2010).

La méthode L-shell proposée par (Bagrow and Bollt, 2005) est un autre exemple de méthode agglomérative pour la détection de communautés. Les communautés y sont définies localement en se basant sur un critère impliquant le nombre de liens à l'intérieur et à l'extérieur du groupe de nœuds. La procédure commence avec un nœud initial et rajoute les nœuds des enveloppes successives, où les enveloppes sont définies comme les ensembles de nœuds se trouvant à une même distance du nœud initial. A chaque étape, le ratio du nombre de liens internes sur le nombre de liens externes est calculé. Le processus est arrêté quand l'ajout d'une nouvelle enveloppe fait que le ratio passe en-dessous d'un seuil choisi. Cette approche a l'avantage d'avoir une exécution rapide. Mais, son inconvénient principal réside dans le choix du nœud initial qui a une grande influence sur le partitionnement obtenu. La solution proposée par les auteurs pour se passer du choix du nœud initial est très coûteuse en terme de calcul et s'exécute en $O(n^3)$.

La méthode Infomap (Rosvall and Bergstrom, 2008) fait aussi partie de cette famille de méthodes agglomératives. Les auteurs y proposent une méthode basée sur la théorie de l'information pour trouver des communautés dans un graphe pondéré et dirigé. Leur idée est de chercher le codage binaire le plus court possible qui permet d'encoder les nœuds du

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

graphe. La procédure simule des marches aléatoires sur le graphe puis les représente en utilisant une séquence de codes binaires. Au lieu de contraindre les nœuds à prendre des codes uniques, il est permis à ce que deux nœuds de deux communautés différentes aient un même code. En contrepartie, la procédure rajoute des codes supplémentaires pour représenter les communautés. Ces codes de communautés sont utilisés dans les séquences représentants les marches aléatoires pour signaler que la marche entre ou sort d'une communauté. Ceci permet de représenter les marches aléatoires avec des séquences plus courtes car les marches aléatoires ont tendance à rester à l'intérieur des communautés. A la fin de la procédure, la taille moyenne des séquences de codes représentants les marches aléatoires est minimale et le partitionnement qui a été utilisé pour le codage est optimal.

2.3.1.2 Méthodes par optimisation de la modularité

La modularité d'un sous-graphe mesure la différence entre le sous-graphe observé et un graphe aléatoire, ou modèle nul, qui a le même ensemble de nœuds mais dont les arêtes sont distribuées aléatoirement dans le graphe tout en respectant la distribution des degrés de départ. Un sous-graphe est considéré comme une communauté si le nombre de ses liens internes est supérieur à l'espérance du nombre de liens internes dans le cas du modèle nul. La modularité d'un partitionnement est la somme des modularités de ses communautés. Plus la modularité d'un partitionnement est élevée, meilleure est sa qualité. La modularité est définie comme suit :

$$Q = \frac{1}{2m} \sum_{\substack{v_i, v_j \in V \\ i \neq j}} \left[A_{ij} - \frac{k_i k_j}{2m}\right] \delta(c_i, c_j) \qquad \text{où} \quad m = \frac{1}{2} \sum_{i \neq j} A_{ij} \tag{2.6}$$

avec A la matrice d'adjacence du graphe. $k_i = d(v_i) = |\{v' \in V \mid (v_i, v') \in E\}|$ est le degré du nœud v_i, c_i est la communauté du nœud v_i à l'étape courante et $\delta(a, b)$ est égale à 1 si $a = b$ et 0 autrement. Le terme $\frac{k_i k_j}{2m}$ correspond à la probabilité de l'existence d'un lien entre les nœuds v_i et v_j dans le modèle nul le plus utilisé dans la littérature. Elle y est proportionnelle aux degrés k_i et k_j des extrémités du lien.

Trouver le partitionnement optimal est très coûteux car le nombre de partitionnements possibles est très grand même dans le cas où le graphe est petit. En effet, l'optimisation de la modularité est un problème NP-complet (Brandes et al., 2006), il n'est pas possible de trouver le partitionnement optimal en un temps polynomial par rapport à la taille du graphe. Mais, il existe plusieurs approches qui permettent d'obtenir, en un temps de calcul raisonnable, de bonnes approximations du partitionnement de modularité maximale. L'équation ci-dessus n'est pas l'unique définition de la modularité, il en existe plusieurs autres adaptées à des cas particuliers de graphes ou pour des besoins particuliers. Par exemple, Macon et al. (2010) ont proposé une formulation modifiée de la modularité : ils y ont inclus un paramètre de résolution qui permet de contrôler la taille des communautés dans le partitionnement obtenu. Ceci permet d'observer les différentes structures en communautés du graphe à différentes échelles. Les auteurs ont démontré l'utilité de tester plusieurs valeurs pour ce paramètre dans les algorithmes de maximisation de modularité. C'est un paramètre important pour les graphes réels car leurs communautés peuvent avoir des tailles très différentes.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

L'approche la plus populaire pour trouver le partitionnement de modularité optimale est celle des méthodes glouton (greedy algorithms). Ces méthodes construisent itérativement l'ensemble des communautés et affectent des nœuds à des communautés de façon à maximiser la modularité à chaque étape. Les algorithmes dans Blondel et al. (2008), Clauset et al. (2004), Newman (2004) suivent cette approche.

Clauset et al. (2004) utilisent le fait que la matrice d'adjacence des réseaux réels est souvent creuse et proposent d'employer des structures de données adaptées comme les listes d'adjacence et les tas pour accélérer les calculs. Ils obtiennent ainsi une méthode de complexité en $O(n \log^2(n))$ ce qui permet d'extraire les communautés de graphes de taille allant jusqu'à 10^6 nœuds.

L'algorithme dans (Blondel et al., 2008) part de la partition discrète dans laquelle chaque nœud est une communauté à part entière. Dans une première étape, la méthode parcours séquentiellement les nœuds du graphe et assigne chaque nœud à la communauté voisine de façon à avoir la plus grande augmentation possible de la modularité. Dans une deuxième étape, les communautés trouvées précédemment sont représentées par un hyper-nœud dans un hyper-graphe et reliées par un lien pondéré par la somme des poids des liens qui relient les communautés dans le niveau inférieur. Ces deux étapes sont répétées et donnent des hyper-graphes à des niveaux hiérarchiques de plus en plus élevés. L'algorithme s'arrête quand aucune modification ne permet d'augmenter la modularité. Il a une complexité de $O(m)$ ce qui lui permet d'être appliqué à des graphes de taille jusqu'à 10^9 liens.

Bien que de nombreuses méthodes se basent sur la modularité, celle-ci a des inconvénients inhérents pour la détection de communautés. De grandes valeurs pour la modularité ne sont pas forcément synonymes d'une structure en communautés réelle. Comme le montre Reichardt and Bornholdt (2006), la modularité peut prendre de grandes valeurs pour certains partitionnements de graphes aléatoires qui n'ont, en réalité, aucune structure en communautés. Pour s'assurer que le partitionnement de modularité maximale d'un graphe correspond à une vraie structure en communautés, il faut vérifier que la valeur de la modularité trouvée est significativement supérieure à la modularité maximale d'un graphe aléatoire de même taille et de même distribution de degrés. Par ailleurs, Fortunato and Barthelemy (2007) ont montré que l'optimisation de la modularité peut échouer à identifier les communautés de petite taille même si celles-ci apparaissent sans ambiguïté dans le graphe (des cliques par exemple). C'est le problème de résolution limite. Enfin, Good et al. (2010) ont inspecté visuellement la forme de la fonction de modularité en fonction des partitionnements possibles. Ils ont découvert qu'il y a un nombre exponentiel de partitionnements différents et qui ont tous une modularité proche de la valeur maximale. Ceci explique pourquoi des méthodes différentes de détection de communautés par maximisation de la modularité peuvent retourner des résultats très différents. Le souci, cependant, est qu'on ne peut décider lequel de ces partitionnements est le meilleur en l'absence d'informations complémentaires sur le graphe.

2.3.1.3 Représentation dans un espace vectoriel et clustering

L'objectif d'une méthode de représentation dans un espace vectoriel (ou plongement de graphes) est de représenter le graphe dans un espace vectoriel de dimension $d << |V|$ en préservant certaines de ses propriétés. Le graphe est alors représenté par des vecteurs de dimension d correspondants au plongement de certaines de ses parties (nœuds, liens ou sous-

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

graphes). L'approche la plus commune est celle du plongement des nœuds du graphe qui peut être réalisée par 3 techniques : la factorisation de matrices, l'apprentissage de réseaux de neurones ou encore l'optimisation de certaines métriques qui mesurent la préservation de la structure de connectivité du graphe. Une fois la représentation des nœuds dans l'espace vectoriel obtenue, l'application d'une méthode de classification non supervisée (clustering) comme k-means (Macqueen, 1967) permet d'obtenir les communautés de nœuds du graphe initial.

Factorisation de matrices

Les méthodes de cette famille trouvent un partitionnement des nœuds du graphe en utilisant les les vecteurs propres dominants de la matrice d'adjacence ou d'une matrice qui en est dérivée. L'idée est que le changement de représentation induit par les vecteurs propres dominants permet d'obtenir une représentation des nœuds du graphe dans un espace vectoriel de dimension plus petite (égale au nombre de vecteurs propres dominants). Le changement de représentation rend la séparation des nœuds en communautés plus évidente. En général, les méthodes de factorisation de matrices utilisent le Laplacien de la matrice d'adjacence. Le Laplacien L est défini comme la différence entre la matrice diagonale des degrés D et la matrice d'adjacence A : $L = D - A$. La matrice laplacienne a l'avantage d'être symétrique et semi-définie positive. Ng et al. (2001), par exemple, utilisent le Laplacien non normalisé L, tandis que Jianbo Shi and Malik (2000) factorisent le Laplacien normalisé L_{rw} défini par : $L_{rw} = D^{-\frac{1}{2}}(D-A)D^{-\frac{1}{2}}$. Pour obtenir un partitionnement des nœuds du graphe en k communautés, la première étape consiste en le calcul des vecteurs propres correspondant aux k plus petites valeurs propres du Laplacien. La matrice formée par les vecteurs propres en colonnes contient alors les coordonnées des nœuds du graphe dans la base des vecteurs propres. Cependant, le calcul exact des vecteurs propres d'une matrice peut devenir impossible si la taille du graphe est grande. Dans ce cas, certaines techniques telles que l'algorithme de Lanczos (Van Loan, 1983) permettent d'obtenir des approximations pour les vecteurs propres.

Réseaux de neurones

Les méthodes qui utilisent des réseaux de neurones pour la représentation des graphes dans un espace vectoriel peuvent être classifiées en deux groupes : les méthodes basées sur la simulation de marches aléatoires sur le graphe et celles qui utilisent un auto-encodeur.

— Marches aléatoires : Pour les méthodes de cette catégorie, la structure du graphe est représentée par un ensemble de chemins réalisés par des marches aléatoires sur le graphe. L'idée est de chercher le plongement de nœuds qui permet la meilleure reconstruction de ces chemins. A cet effet, Perozzi et al. (2014) ont proposé l'algorithme DeepWalk qui utilise des marches aléatoires tronquées pour définir les voisinages des nœuds. Le plongement est obtenu en maximisant la probabilité conditionnelle d'observer le voisinage sachant les coordonnées des nœuds dans l'espace de représentation. Les auteurs se sont inspirés des méthodes du traitement du langage et plus particulièrement du modèle Skip-gram utilisé dans word2vec (Mikolov et al., 2013a) pour représenter les mots d'un corpus dans un espace vectoriel en utilisant leurs voisinages. Un autre exemple est la méthode node2vec (Grover and Leskovec, 2016). La particularité de cette méthode est d'utiliser une marche aléatoire personnalisée et paramétrée qui associe les

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

explorations en profondeur et en largeur du graphe pour construire les voisinages des nœuds.

— Auto-encodeur : L'idée de cette famille de méthodes est que l'architecture en de multiple couches d'un réseau de neurones permet un apprentissage robuste et efficace de la structure du graphe pour pouvoir le représenter dans un espace vectoriel de petite dimension. Dans (Tian et al., 2014) et (Wang et al., 2016), les auteurs utilisent des modèles d'apprentissage profond pour représenter les nœuds de façon à préserver leurs proximités du premier et du second ordre. Les modèles ont deux composantes : un encodeur qui plonge les nœuds du graphe dans un espace vectoriel et un décodeur qui utilise les coordonnées obtenues pour reconstruire le graphe. La représentation est obtenue en minimisant l'erreur de reconstruction du graphe. GCN (Kipf and Welling, 2016) est un autre algorithme de cette famille, il utilise une version de réseaux de neurones convolutionnels capable de prendre en entrée directement la matrice d'adjacence des graphes. L'algorithme GCN a été développé pour la classification semi-supervisée des nœuds d'un graphe attribué, mais il permet aussi de calculer des représentations dans un espace vectoriel des nœuds. Ces représentations sont obtenues, après la phase d'apprentissage, dans la dernière couche cachée du réseau de neurones.

Optimisation de la préservation des liens

Les méthodes de cette famille cherchent une représentation des nœuds du graphe de sorte que les liens qui en sont déduits correspondent au maximum aux liens du graphe d'origine. Par exemple, (Zhou et al., 2017) considèrent qu'une bonne représentation d'un graphe maximise la probabilité de générer la même structure de liens observée dans le graphe. (Tang et al., 2015) considèrent que les voisinages des nœuds calculés dans l'espace de représentation doivent être les plus proches possibles des voisinages obtenus à partir des liens du graphe. Et, dans (Bordes et al., 2014), les auteurs cherchent un changement de représentation de sorte que les coordonnées d'un nœud soient similaires à ceux de ses voisins dans le graphe initial.

Les coordonnées des nœuds dans l'espace vectoriel peuvent être utilisées avec un algorithme de clustering comme *k*-means (Macqueen, 1967) pour trouver les communautés dans le graphe d'origine.

2.3.1.4 Propagation de labels

Certaines méthodes de détection de communautés dans les graphes se basent sur la simulation de propagation de labels. Par exemple, dans (Raghavan et al., 2007), les auteurs présentent un algorithme itératif basé sur la propagation de labels pour extraire les communautés depuis un graphe pondéré. A l'initialisation, chaque nœud du graphe a son propre label. Par la suite, et à chaque étape de l'exécution de l'algorithme, une stratégie de vote est adoptée par chaque nœud en prenant le label le plus populaire parmi ceux de ses nœuds voisins. La procédure s'arrête quand la communauté de chaque nœud du graphe est la communauté la plus répandue parmi celles de ses voisins. La méthode ne nécessite ni optimisation de mesure ni connaissances a priori sur le nombre ou la taille des communautés. Par la suite, Barber and Clark (2009) ont reformulé l'algorithme de propagation de labels LPA en un problème d'optimisation. Ils obtiennent de meilleures communautés en ajoutant

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

des contraintes à la procédure de propagation de labels de sorte à favoriser les partitionnements avec des grandes valeurs de modularité. Ils proposent trois formes de contraintes : une contrainte favorisant les partitionnements qui ont des communautés de même taille, et deux contraintes pour trouver des communautés qui ont un degré total similaire pour le cas des graphes simples et des graphes bipartites. Les méthodes de propagation de labels ont l'avantage d'être simples à implémenter et de s'exécuter rapidement. Mais, dans les cas où la structure en communautés du graphe n'est pas évidente, le processus de diffusion peut entraîner la construction d'une communauté triviale qui contient la quasi-totalité des nœuds du graphe. Pour éviter cette situation, Leung et al. (2009) ont proposé une modification de la procédure de propagation de labels qui utilise un score d'atténuation décroissant associé aux communautés pour contrôler leur taille et éviter qu'une d'entre elles ne devienne tellement répandue qu'elle englobe la majeure proportion des nœuds du graphe. Une autre amélioration de la propagation de labels a été introduite par (Fang et al., 2020) qui proposent d'utiliser des labels vectoriels au lieu de labels numériques pour les communautés et de guider la diffusion des labels avec une procédure d'optimisation de la modularité.

2.3.1.5 Modèles de graphes

Il existe plusieurs modèles paramétriques de graphes dans la littérature scientifique. Ils sont utilisés pour obtenir une représentation condensée d'un graphe en estimant les paramètres du modèle qui permettent d'avoir le meilleur ajustement entre le graphe observé et le modèle de graphe. Ils permettent aussi de mieux comprendre le fonctionnement et la structure d'un réseau et dans certains cas son évolution temporelle. Parmi ces modèles de graphes, le modèle en blocs SBM se base sur la structure en communautés du réseau et permet, sous réserve d'un bon ajustement des paramètres, de détecter les communautés du graphe.

Le modèle de graphe le plus ancien est le modèle d'Erdos-Renyi (Erdös and Rényi, 1959). Les auteurs se sont intéressés aux ensembles de graphes non dirigés aléatoires $E_{n,N}$ qui ont n nœuds et N liens. Ils considèrent la formation d'un graphe aléatoire comme la réalisation d'un processus stochastique. A chaque étape, un lien est tiré de manière équiprobable parmi les liens qui ne font pas partie du graphe et y est rajouté jusqu'à l'obtention des N liens.

Parmi les modèles de graphe, ce sont les modèles en blocs qui peuvent être utilisés pour la détection de communautés. Ces modèles se basent sur une structure en sous-groupes du graphe et qui, dans les cas où un bon ajustement peut être trouvé, permettent de partitionner le graphe originel en communautés. L'algorithme SBM (Stochastic Block Model) (Holland et al., 1983) est un exemple de modèles de graphes en blocs. Il utilise un modèle stochastique dans lequel les nœuds du réseau sont partitionnés en sous-groupes appelés blocs. L'estimation des paramètres du modèle est faite par maximum de vraisemblance et le partitionnement obtenu est tel que la probabilité de créer un lien dans un bloc soit grande ; et que la probabilité de créer un lien entre deux nœuds de différents blocs soit petite. Dans (Bouveyron et al., 2016), les auteurs proposent l'algorithme STBM : une extension du modèle SBM classique qui y incorpore des attributs textuels associés aux liens du graphe. Ils s'intéressent au problème de détection de communautés de nœuds cohérentes en terme de structure des interactions et de contenu sémantique.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

2.3.1.6 Discussion : avantages et inconvénients des différentes approches

Le problème de détection de communautés dans les graphes formulé comme un problème d'optimisation de critère est en général difficile à résoudre. En effet, la recherche d'un optimum global parmi les partitionnements possibles du graphe est un problème NP-difficile et l'usage d'heuristiques est nécessaire pour obtenir une approximation du partitionnement optimal.

Concernant les méthodes de changement de représentation, elles permettent d'extraire les communautés d'un graphe en appliquant un algorithme de classification non supervisée sur les coordonnées des nœuds dans l'espace vectoriel euclidien obtenu. Les ressources en calcul nécessaires pour obtenir les communautés sont alors le cumul de celles nécessaires au changement de représentation et de celles utilisées pour le clustering. De plus, la dimension de l'espace de représentation obtenu est souvent grande ($\geqslant 100$) ce qui affecte négativement les performances des algorithmes de classification à cause du "fléau de la dimensionnalité" (Steinbach et al., 2003).

Les méthodes de propagation de labels ont l'avantage d'avoir un temps d'exécution court mais elles ont l'inconvénient d'être aléatoires et de retourner des partitionnements potentiellement différents d'une exécution à une autre.

Les modèles SBM (Stochastic Block Models) permettent d'obtenir un partitionnement du graphe de sorte que la probabilité d'existence d'un lien entre une paire de nœuds d'une même partition soit supérieure à la probabilité d'existence d'un lien entre deux nœuds appartenant à des partitions différentes. Ces modèles nécessitent aussi le choix initial du nombre de partitions recherchés. Il faut alors ajuster plusieurs modèles (en faisant varier le paramètre) et les comparer pour trouver le nombre de partitions optimal.

2.3.2 Agrégation des communautés de plusieurs couches

La deuxième catégorie contient les méthodes qui définissent les communautés d'un graphe multicouche comme l'agrégation des partitionnements trouvés dans chaque couche. Pour les méthodes de cette catégorie, la détection de communautés se fait en cherchant un consensus entre les partitionnements calculés couche par couche avec une des méthodes présentées dans la section 2.3.1. L'algorithme ABACUS (Berlingerio et al., 2013) est un exemple de ces méthodes. Il utilise une méthode de détection de communautés dans les graphes simples sur chaque couche du multigraphe, puis définit les communautés multicouches comme les ensembles de nœuds qui appartiennent aux mêmes partitions dans une ou plusieurs couches. ABACUS détecte ces communautés multicouches en utilisant l'algorithme d'extraction de motifs Apriori sur le contexte formel composé des partitions obtenues dans chaque couche.

Les algorithmes MiMAG (Boden et al., 2017) et EMCD (Tagarelli et al., 2017) font aussi partie de cette famille de méthodes. L'algorithme MiMaG est basé sur l'extraction des "sous-graphes multicouches cohérents" (MLCS : Multi-layer coherent subgraph) pour la détection de coummunautés non redondantes dans des multigraphes possédant des attributs sur les liens. Les auteurs définissent d'abord les sous-graphes cohérents dans les graphes simples (à une couche) attribués sur les liens $G = (V, E, a_e)$ où a_e est une application qui retourne pour chaque lien la valeur d'attribut qui lui est associée. Ce sont les sous-ensembles de nœuds $O \subseteq V$ tels que : (1) le sous-graphe engendré par O est une 0.5-quasi-clique c'est-à-dire

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

que chaque nœud de O est connecté à au moins $0.5(|O| - 1)$ nœuds de O, et (2) toutes les paires d'attributs associés aux liens dont les extrémités sont dans O ont une distance plus petite qu'un seuil w fixé au départ. (Boden et al., 2017) généralisent cette structure pour les graphes multicouches attribués sur les liens $G = (V, E, L, a_e)$ en définissant les sous-graphes multicouches cohérents $C = (O, S)$ tels que $\forall i \in L$, si $i \in S$ alors O est un sous-graphe cohérent dans le graphe simple attribué correspondant à la couche G_i du multigraphe. Les auteurs définissent pour les sous-graphes multicouches cohérents une mesure de redondance et une fonction de qualité qui favorise ceux qui ont un grand nombre de nœuds, de couches et de liens internes. Ils proposent l'algorithme MiMaG pour trouver une solution approximative au problème de recherche d'ensembles de sous-graphes multicouches cohérents telles que les redondances de toutes les paires soient inférieures à un seuil fixé et que la somme des mesures de qualité soit maximale.

Concernant la méthode EMCD (Tagarelli et al., 2017), elle utilise aussi un consensus entre les communautés de chaque couche. Étant donnés des partitionnements couche par couche des nœuds du multigraphe, une matrice de co-association M de dimension $|V| \times |V|$ est construite de sorte à contenir pour chaque paire de nœuds le nombre de couches dans lesquelles ils sont reliés par un lien et ils appartiennent à une même communauté. Un seuil minimal θ est choisi et permet de filtrer la matrice M et d'obtenir un partitionnement des nœuds du multigraphe par consensus. Ce partitionnement permet d'obtenir un premier ensemble de communautés en utilisant les sous-graphes engendrés par les partitions. Cet ensemble de communautés est appelé C-EMCD et son graphe sous-jacent est considéré comme la borne supérieure du multigraphe. Un second partitionnement des nœuds du multigraphe est calculé en contraignant chaque communauté C-EMCD à ne contenir que les couches dans lesquelles le partitionnement individuel de la couche est en accord avec le partitionnement par consensus du multigraphe. Les communautés résultantes sont appelées CC-EMCD et leur graphe sous-jacent peut être considéré comme la borne inférieure du multigraphe. Par la suite, un algorithme de clustering guidé par une mesure de modularité définie pour les graphes multicouches permet d'obtenir le partitionnement optimal recherché pour les nœuds du multigraphe parmi les partitionnements $\hat{C}$ qui vérifient : CC-EMCD $\subseteq$ $\hat{C} \subseteq$ C-EMCD où l'inclusion est celle des ensembles de liens des communautés.

2.3.3 Détection directe de communautés dans un multigraphe

Il existe des méthodes plus récentes qui recherchent les communautés directement dans le multigraphe et ne nécessitent pas son aplatissement ni d'agrégation des partitionnements couche par couche. Ces méthodes sont basées soit sur l'optimisation de la modularité, sur la propagation de labels ou encore sur le changement de représentation.

2.3.3.1 Approche par optimisation de grandeurs

Les méthodes de cette approche utilisent des fonctions de qualité appropriées aux partitionnements des graphes multicouches et trouvent les communautés qui permettent d'optimiser ces fonctions.

Parmi ces méthodes, celles qui sont basées sur l'optimisation de la modularité (Liu et al., 2014, Mucha et al., 2010, Pramanik et al., 2017) sont les plus répandues. Dans (Mucha et al., 2010), les auteurs définissent une mesure de modularité pour les graphes multicouches

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

$\mathcal{G} = (V, E, \mathcal{L})$ en utilisant les dynamiques laplaciennes pour la définition du modèle nul auquel sont comparés les partitionnements. La formule de la modularité est donnée par :

$$Q_{multicouche} = \frac{1}{2\mu} \sum_{ijsr} \left\{ (A_{ijs} - \gamma_s \frac{k_{is} k_{js}}{2m_s}) \delta_{sr} + \delta_{ij} C_{jsr} \right\} \delta(g_{is}, g_{jr})$$

où i et j correspondent à des nœuds et s et r à des couches dans $\mathcal{G}$. A_{ijs} dénote l'élément d'indice (i, j) dans la matrice d'adjacence A_s de la couche s, k_{is} est le degré du nœud i dans la couche s, δ indique si les éléments qui lui sont passés sont égaux, m_s est le nombre de liens dans la couche s, C_{jsr} correspond au poids du lien de couplage du nœuds j entre les couches s et r. Et, $2\mu = \sum_{jr} \kappa_{jr}$ où $\kappa_{jr} = k_{jr} + cjr$ avec c_{jr} le degré de couplage du nœud $j : c_{js} = \sum_r C_{jsr}$.

Carchiolo et al. (2010) utilisent cette même formule de modularité multicouche et présentent une procédure pour accélérer l'exécution de l'algorithme de recherche du partitionnement de plus grande modularité dans un graphe multicouche. Ils proposent de réduire la taille du graphe sans affecter la modularité pour obtenir, avec un moindre coût, le partitionnement du multigraphe réduit qui soit équivalent à celui du graphe originel.

Pramanik et al. (2017) introduisent une autre adaptation de la mesure de modularité au cas des graphes multicouches notée Q_M. La modularité Q_M est applicable au cas général des graphes multicouches pour lesquels le couplage n'est pas forcément diagonal c'est-à-dire que les liens de couplage ne se limitent pas aux liens entre les occurrences d'un même nœud dans différentes couches. Ils définissent la modularité Q_M explicitement pour les multigraphes contenant 2 couches de façon à favoriser les communautés avec les propriétés suivantes :

(i) grande cohésion des nœuds dans les graphes simples représentant les couches et dans le graphe bipartite contenant les liens de couplage entre les couches
(ii) les liens de couplage doivent connecter une grande partie des nœuds des deux couches

Les auteurs ont montré que leur algorithme de détection de communautés qui optimise la modularité Q_M est capable de détecter à la fois les communautés dispersées sur plusieurs couches et les communautés constituées de nœuds d'une même couche. Les auteurs présentent aussi une méthodologie pour la construction de graphes multicouches aléatoires pour l'évaluation de la qualité et la comparaison des méthodes de détection de communautés.

Par ailleurs, dans (Liu et al., 2014), les auteurs développent une nouvelle mesure de modularité pour les graphes multicouches et hétérogènes (contenant plusieurs types de nœuds) : la modularité composite définie comme la somme pondérée des modularités classiques (Newman, 2004) pour les couches contenant un seul type de nœuds et des modularités k-partites (Murata and Ikeya, 2010) pour les couches contenant plusieurs types de nœuds. La modularité composite d'un partitionnement des nœuds C est donnée par :

$$Q(C) = \sum_{y=1}^{|\mathcal{L}|} \frac{1}{w^{[y]}} Q^{[y]}(C)$$

où $w^{[y]}$ représente l'importance relative de la couche y et $Q^{[y]}$ est la modularité classique pour les couches unipartites et la modularité k-partite pour les couches contenant plusieurs types de nœuds.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Parmi les autres méthodes par optimisation, il y a les algorithmes basés sur la connectivité comme la percolation des cliques pour les graphes multicouches (Afsarmanesh and Magnani, 2016). Pour cette approche, les auteurs introduisent le concept de *k*-*m*-clique qui est un ensemble de nœuds de taille k complètement connectés entre eux dans au moins m couches du multigraphe. L'algorithme qu'ils proposent extrait les *k*-*m*-cliques maximales et rassemble celles qui sont adjacentes pour former des communautés. Nous pouvons aussi citer l'algorithme Infomap pour les multigraphes (Edler et al., 2017) qui cherche l'encodage optimal pour les nœuds permettant d'obtenir la représentation la plus courte des flux simulés par des marches aléatoires dans le multigraphe. A la différence de la première version de l'algorithme (Rosvall and Bergstrom, 2008) qui a été développée pour les graphes à une couche, cette version pour les multigraphes utilise des marches aléatoires d'ordre supérieur à 1.
Pour finir, l'algorithme dans (Boutemine and Bouguessa, 2017) utilise une stratégie de propagation de labels pour guider l'optimisation. Cette approche a l'avantage de ne pas nécessiter de paramètres à fixer par l'utilisateur. Elle permet aussi d'identifier les couches non informatives pour la détection des communautés dans le multigraphe et de caractériser chaque communauté de nœuds extraite avec les couches les plus pertinentes dans lesquelles ils apparaissent. Ils définissent les couches les plus pertinentes comme celles dans lesquelles les nœuds de la communauté ont les plus grands degrés.

2.3.3.2 Représentation dans un espace vectoriel et clustering

Enfin, il y a une catégorie supplémentaire qui peut être ajoutée aux méthodes globales de détection de communautés dans les multigraphes : ce sont les algorithmes de changement de représentation qui permettent de calculer les coordonnées des nœuds d'un graphe multicouche dans un espace vectoriel euclidien en préservant la structure topologique du graphe original (Cen et al., 2019, Zhang et al., 2018). Pour ces méthodes, appliquer un algorithme de classification non supervisée sur les coordonnées des nœuds dans l'espace vectoriel (avec l'algorithme k-means (Macqueen, 1967) par exemple) revient à partitionner le multigraphe en communautés. Parmi les méthodes de cette catégorie, l'algorithme MNE (Zhang et al., 2018) permet d'obtenir la représentation vectorielle des nœuds en préservant leur voisinage couche par couche. Les coordonnées v_n^i d'un nœud n de la couche i sont de la forme : $\forall n \in V, v_n^i = b_n + w^i X^{i\intercal} u_n^i$ où $b_n \in \mathbb{R}^d$ (avec $d << |\mathcal{V}|$) est la représentation vectorielle du nœud n commune à toutes ses occurrences dans les différentes couches du multigraphe, w^i est un poids représentant l'importance de la couche i, $u_n^i \in \mathbb{R}^s$ (avec $s << d$) est la représentation vectorielle de n dans la couche i et la matrice X^i est une matrice de transformation qui permet d'aligner b_n et u_n^i. Des marches aléatoires sont simulées dans les couches du multigraphe pour générer des séquences de nœuds et construire les voisinages à préserver. Par la suite, une procédure d'optimisation par descente de gradient permet de calculer les coordonnées v_n^i des nœuds qui permettent la plus grande préservation des voisinages. Plus tard, (Cen et al., 2019) ont proposé l'algorithme GATNE qui généralise la méthode MNE (Zhang et al., 2018) au cas des multigraphes hétérogènes (contenant différents types de nœuds) et attribués sur les nœuds.

Pour répondre à notre objectif de détection de communautés dans les multigraphes de co-appartenance, les méthodes présentées dans cette partie ont certaines limites :

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

— Les attributs textuels associés aux couches ne sont pas pris en compte dans le calcul des communautés
— Dans la partie expérimentale, ces méthodes ont nécessité beaucoup de ressources de calcul car leur complexité temporelle croît très rapidement avec le nombre de couches du multigraphe

2.4 Évaluation de la qualité d'un partitionnement

Une fois la détection des communautés d'un graphe accomplie, l'étape suivante est l'évaluation de la qualité des communautés obtenues. L'évaluation est plus simple à réaliser quand la vérité terrain de la structure en communautés du graphe est connue par avance. Dans ces cas, plusieurs métriques peuvent être utilisées pour estimer la correspondance entre le partitionnement trouvé par la méthode de détection de communautés et la vérité terrain. Autrement, quand les vraies communautés ne sont pas connues, les différents critères utilisés dans les approches par optimisation (Sections 2.3.1.2, 2.3.1.1 et 2.3.3) peuvent aussi être calculés sur les partitionnements obtenus à des fins d'évaluation mais il faut faire attention à ce que le critère choisi ne soit optimisé par aucune des méthodes à comparer.

2.4.1 Graphes synthétiques

Les graphes synthétiques générés par ordinateur sont utiles pour évaluer la qualité d'une méthode de détection de communautés car la vraie structure en communautés est connue à l'avance pour ces graphes. Ils permettent ainsi d'évaluer la qualité d'un partitionnement en communautés et de comparer les performances de différentes méthodes d'extraction de communautés. Dans (Girvan and Newman, 2002), les auteurs proposent une procédure simple pour la construction de graphes synthétiques : tout d'abord, les nœuds sont divisés aléatoirement en communautés ; ensuite, les paires de nœuds appartenant à une même communauté sont reliées par une arête avec une probabilité p_{in} et les paires de nœuds appartenant à deux communautés différentes avec une probabilité p_{out}. La qualité d'un partitionnement trouvé par une méthode de détection de communautés est estimée en calculant la proportion de nœuds correctement classifiés. Plus tard, Lancichinetti et al. (2008) ont relevé certains inconvénients liés à l'utilisation des graphes synthétiques classiques pour tester la performance des algorithmes de détection de communautés. Ils expliquent que les modèles de graphes aléatoires existants dans la littérature ont des caractéristiques structurelles différentes des réseaux concrets rencontrés dans le monde réel. Ils introduisent une nouvelle classe de graphes pour réaliser cette tâche. Ces graphes sont plus appropriés pour comparer les performances des algorithmes de détection de communautés car ils prennent en compte les distributions des degrés des nœuds et des tailles de communautés.

2.4.2 Comparaison avec les communautés réelles

Plusieurs métriques peuvent être utilisées pour évaluer la distance entre le partitionnement trouvé par la méthode et la vérité-terrain des communautés. Certaines métriques sont basées sur le décompte des paires de nœuds pour lesquels le partitionnement calculé et la vérité-terrain des communautés sont en accord. Par exemple, les critères proposées par Wallace (1983) estiment la probabilité qu'une paire de nœuds appartenant à une même

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

communauté dans un partitionnement soient dans une même communauté dans l'autre partitionnement. Une autre possibilité est l'indice de Jaccard (Ben-Hur et al., 2002) qui calcule le ratio de paires de nœuds pour lesquelles les deux partitionnements sont en accord. Soient $\mathcal{P}_1$ et $\mathcal{P}_2$ deux partitionnements des nœuds V d'un graphe $G = (V, E)$. Nous pouvons en dériver les matrices $C^{(1)}$ et $C^{(2)}$ telles que :

$$C_{ij} = \begin{cases} 1 & \text{si } V_i \text{ et } V_j \text{ sont dans la même communauté et } i \neq j \\ 0 & \text{sinon} \end{cases}$$

Ensuite, en définissant le produit scalaire entre les matrices de représentation des partitionnements :

$$\langle C^{(1)}, C^{(2)} \rangle = \sum_{i,j} C^{(1)}_{ij} . C^{(2)}_{ij}$$

Le coefficient de Jaccard 'écrit alors :

$$J(\mathcal{P}_1, \mathcal{P}_2) = \frac{\langle C^{(1)}, C^{(2)} \rangle}{\langle C^{(1)}, C^{(1)} \rangle + \langle C^{(2)}, C^{(2)} \rangle - \langle C^{(1)}, C^{(2)} \rangle}$$

D'autres métriques sont basées sur l'appariement des communautés. La méthode proposée par Meil and Heckerman (2001) cherche la meilleure correspondance entre les communautés des deux partitionnements et définit la distance qui les sépare comme la proportion de nœuds classés dans des communautés non correspondantes. Un autre exemple est le critère symétrique proposé par Dongen (2000) qui calcule le nombre de nœuds classés dans des communautés différentes pour les deux partitionnements considérés.

D'autres métriques pour la comparaison de partitionnements sont basées sur la théorie de l'information. Par exemple, dans (Lancichinetti et al., 2009), les auteurs développent un algorithme de détection de communautés chevauchantes et estiment la qualité des partitionnements trouvés en les comparant à la vérité-terrain des communautés à l'aide de l'information mutuelle normalisée (Danon et al., 2007). Cette métrique est basée sur le calcul de la matrice de confusion N dans laquelle les lignes correspondent aux vrais communautés A et les colonnes aux communautés à évaluer B. L'élément N_{ij} contient le nombre de nœuds de la vraie communauté i qui apparaissent dans la communauté à évaluer j.Nous avons alors :

$$NMI(A, B) = \frac{-2 \sum_{i=1}^{c_A} \sum_{j=1}^{c_B} N_{ij} log(\frac{N_{ij} n}{N_{i.} N_{.j}})}{\sum_{i=1}^{c_A} N_{i.} log(\frac{N_{i.}}{n}) + \sum_{i=1}^{c_B} N_{.j} log(\frac{N_{.j}}{n})}$$

où c_A et c_B représentent les nombres de communautés dans A et B, n est le nombre de nœuds. $N_{i.}$ est la somme des valeurs de la ligne i de la matrice N et $N_{.j}$ est la somme des valeurs de la colonne j de N.

Nous pouvons aussi citer la variation d'information (VI) introduite dans (Meila, 2007). Cette mesure a des propriétés intéressantes : elle vérifie les conditions d'une métrique sur l'espace des partitionnements et, ses valeurs sont comparables même dans des conditions expérimentales différentes. Par ailleurs, Yang and Leskovec (2015) utilisent des exemples de graphes pour lesquels les communautés réelles sont connues à l'avance. Ils comparent les communautés obtenues par 13 approches différentes chacune basée sur l'optimisation d'une grandeur spécifique. Pour cela, ils utilisent 4 métriques d'évaluation : la séparabilité,

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

la densité, la cohésion et le coefficient de clustering. Ils analysent aussi la robustesse des approches aux perturbations : une grandeur à optimiser est robuste si elle prend de grandes valeurs pour les vraies communautés et que ces valeurs baissent significativement quand des perturbations aléatoires sont appliquées.

2.4.3 Métriques d'évaluation

Pour pouvoir comparer les résultats de différentes méthodes de détection de communautés en utilisant un graphe $G = (V, E)$ pour lequel les vraies communautés ne sont pas connues, plusieurs critères peuvent être calculés sur les ensembles de communautés à condition qu'aucune des méthodes à comparer ne soit basée sur l'optimisation du critère d'évaluation. Parmi ces critères, nous pouvons citer pour un partitionnement $C = \{C_1, C_2...C_p\}$:

— La modularité $Q(C)$ (équation 2.6) peut être utilisée comme une métrique d'évaluation de la qualité d'un partitionnement et pour comparer les résultats de plusieurs méthodes d'extraction de communautés à condition que ces méthodes ne soient pas basées sur l'optimisation de la modularité.

— La performance $Perf(C)$:

$$Perf(C) = \frac{2|\{(u,v) \in E, c(u) = c(v)\}| + |\{(u,v) \notin E, c(u) \neq c(v)\}|}{n(n-1)}$$

où $c(u)$ retourne la communauté de C à laquelle appartient le nœud u et $n = |V|$ est le nombre de nœuds dans G.La performance mesure le ratio de paires de nœuds correctement interprétées sur le nombre total de paires de nœuds. Une paire de nœuds est considérée comme correctement interprétée si les deux nœuds sont adjacents et appartiennent à une même communauté ou si les deux nœuds ne sont pas adjacents et apparaissent dans des communautés différentes.

— La couverture $Cov(C)$:

$$Cov(C) = \frac{m_{in}}{m}$$

où m_{in} est la cardinalité du sous-ensemble de liens dont les deux extrémités appartiennent à une même communauté et m est le nombre de liens dans G. La couverture d'un partitionnement est le ratio du nombre de liens intra-communautés sur le nombre total de liens.

— La conductance $Cond(C)$:

$$Cond(C) = \min_{C_i \in C} Cond(C_i) \quad \text{avec} \quad Cond(C_i) = \frac{Ext(C_i)}{\min(\sum_{v \in C_i} d_v, \sum_{v \in V \setminus C_i} d_v)}$$

où $Ext(C_i)$ est le nombre de liens sortants de la communauté C_i et d_v indique le degré du nœud v. La conductance d'un partitionnement est le minimum de la conductance de ses communautés. La conductance d'une communauté est définie comme le ratio du nombre de liens sortants sur le minimum entre le nombre de liens dans la communauté et le nombre de liens dans le reste du graphe.

— La densité interne $d_{int}(C_i)$:

$$d_{int}(C_i) = \frac{2m(C_i)}{n(C_i)(n(C_i)-1)}$$

avec $m(C_i)$ le nombre de liens internes à la communauté C_i et $n(C_i)$ est le nombre de

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

nœuds de C_i. La densité interne est définie pour une communauté, c'est le ratio du nombre de ses liens internes sur le nombre de liens possibles qu'elle peut contenir.

— Le ratio de participation aux triangles $RPT(C_i)$ d'une communauté :

$$RPT(C_i) = \frac{|\{u \in C_i | \exists v, w \in C_i \text{ t.q. } (u,v) \in E, (v,w) \in E, (w,u) \in E\}|}{n(C_i)}$$

C'est le ratio de nœuds qui participent à une triade incluse dans la communauté sur le nombre de nœuds de la communauté.

— Le cut ratio d'une communauté $Cut(C_i)$:

$$Cut(C_i) = \frac{2Ext(C_i)}{n(C_i)(n(C_i)-1)}$$

C'est le ratio du nombre de liens sortants de la communauté sur le nombre de liens possibles dans la communauté

Lee and Cunningham (2014) proposent une méthode d'évaluation alternative basée sur la capacité d'un classifieur à inférer les attributs des nœuds étant données leurs assignations aux communautés. De bonnes performances du classifieur impliquent un plus grand score d'évaluation de la méthode de détection de communautés.

Dans Kaminski et al. (2019), les auteurs assignent "un score de divergence" aux représentations dans un espace vectoriel des nœuds du graphe obtenues par un algorithme d'embedding. Ce score est basé sur le modèle de Chung-Lu et permet de comparer la qualité des représentations obtenues par différentes méthodes.

2.5 Applications aux réseaux sociaux

Dans cette thèse, le principal cas d'application des méthodes proposées pour l'analyse des multigraphes est celui des réseaux sociaux. Ces derniers sont caractérisés par leur grand nombre d'utilisateurs dont l'activité génère une très grande quantité de données. Le réseau social Facebook comptabilise plus de 2.8 milliards d'utilisateurs actifs par mois, et chaque année, 350.10^9 Go de données y sont générées [1]. Ces données sont de différents types : les données structurées correspondants aux informations données par les utilisateurs à leur inscription dans le réseau social (pseudonyme, âge, sexe, région, profession...), et les données non structurées qui peuvent être des textes (messages, commentaires ...), des images ou d'autres types de contenus multimédia partagés sur le réseau social. Une autre caractéristique de ces réseaux sociaux est leur rapide évolution temporelle : chaque seconde, 1074 nouvelles images sont postées sur le réseau social dédié au partage d'images Instagram [2].

En particulier, nous nous intéressons à la plateforme Twitter qui a été créée en 2006. C'est l'outil de microblogging le plus populaire sur le Web. Le microblogging est défini comme une forme de blogging qui permet aux utilisateurs de publier des textes courts et de les partager avec leurs connaissances et/ou avec des observateurs intéressés via une messagerie instantanée, par e-mail ou directement sur des plate-formes dédiées sur le Web. Beaucoup de travaux de recherche se sont intéressés à l'analyse de la structure des graphes d'interactions générés par les activités (messages, publications, création de liste ...) des utilisateurs sur le

1. https ://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/
2. https ://www.statista.com/topics/1882/instagram/

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

réseau social. Ils se basent sur l'idée que les interactions dans la plate-forme virtuel sont un miroir des interactions qu'entretiennent les utilisateurs dans le monde réel.

Les réseaux sociaux sont des portes d'accès à des informations difficilement accessibles par d'autres moyens à cause de contraintes spatiales ou temporelles. En effet, il n'y a pas de limites géographiques à l'accès aux données sur les réseaux sociaux, et dans plusieurs d'entre eux les données sont stockées et restent accessibles durant de longues périodes. Ceci a motivé les chercheurs à développer des méthodes spécifiques pour l'analyse des réseaux sociaux. Dans (Java et al., 2007), les auteurs ont conduit une étude sur la structure topologique et géographique du graphe de suivi de Twitter. C'est un graphe dirigé qui contient un lien dirigé $u \rightarrow v$ si l'utilisateur u suit l'utilisateur v. Ils ont montré que la structure de ce graphe a des propriétés semblables à celles des graphes des pages Web avec notamment une distribution cumulative des degrés des nœuds en loi de puissance avec des paramètres similaires. L'analyse géographique a montré que la probabilité d'existence d'un lien entre deux nœuds est inversement proportionnelle à la distance géographique qui les deux utilisateurs correspondants. Enfin, les auteurs se sont intéressés aux intentions des utilisateurs sur Twitter. Ils ont identifié trois rôles que peuvent jouer les utilisateurs sur le réseau social : sources d'informations, amis, chercheurs d'informations. Ils ont aussi montré que ces rôles n'étaient pas exclusifs, c'est-à-dire qu'un seul utilisateur peut en même temps être source d'informations pour une partie du graphe et chercheur d'informations pour une autre. Enfin, l'application de l'algorithme de détection de communautés CPM (Palla et al., 2005) a permis d'observer que les utilisateurs qui partagent des intentions similaires sont plus susceptibles de créer des liens entre eux de façon à constituer des communautés.

Kim et al. (2010) se sont intéressés aux listes Twitter pour inférer les préférences et les intérêts des utilisateurs du réseau social. Leurs expériences ont montré le bon fonctionnement de leur méthode même pour des intérêts qui n'apparaissent pas directement dans le profil de l'utilisateur. Ces informations sont obtenues en calculant l'ensemble de mots dont la mesure d'exceptionnalité χ^2 est maximale dans le corpus de documents contenant les publications des membres des listes dans lesquels apparaît l'utilisateur. Une approche semblable est utilisée par Yamaguchi et al. (2011) qui utilisent les listes Twitter pour associer aux utilisateurs des "tags" qui caractérisent leurs sujets de publication. Greene et al. (2012) tirent aussi profit de l'information collective contenue dans les listes Twitter créées par différentes personnes pour construire des communautés représentant un intérêt ou un événement commun à leurs membres. Bhattacharya et al. (2014) proposent une méthode sémantique pour identifier des communautés d'utilisateurs Twitter autour d'un sujet d'intérêt. La méthode des auteurs analyse aussi l'organisation des utilisateurs dans ces communautés et identifie les rôles qu'ils y jouent pour la diffusion de l'information : diffusion de l'information par les experts et consommation de l'information par les chercheurs. Rakesh et al. (2014) présentent une méthode de recommandation de listes Twitter aux utilisateurs en fonction de leurs centres d'intérêt. Leur travail comporte deux contributions : ListRec et ListPageRank. ListRec permet de modéliser les préférences des utilisateurs Twitter pour les listes en fonction du contenu de leurs publications (tweets), du réseau d'interactions entre utilisateurs et de la popularité des listes. ListPageRank permet de recommander de nouvelles listes aux utilisateurs en se basant sur leurs centres d'intérêt et sur le graphe de similarité entre listes Twitter.

Une analyse de la structure de différents réseaux de publication a permis à Wang et al. (2011) d'observer que les utilisateurs peuvent être séparés en deux groupes selon leur degré d'influence et leur comportement. Ces deux groupes sont celui des influenceurs dont le

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

nombre est petit mais qui produisent la majeure partie du contenu ; et les amateurs qui, à l'inverse, sont nombreux et ne produisent que très peu de contenu. Les auteurs présentent deux algorithmes d'optimisation pour extraire les cœurs des communautés, généralement composés d'influenceurs, à partir des réseaux sociaux de grande taille comme Twitter.

Velichety and Ram (2013) présentent une analyse exploratoire de la relation entre les trois formes d'usage de l'information sur Twitter : le suivi d'utilisateurs, la création et souscription à une liste. les auteurs montrent que, pour un même utilisateur, l'information qu'il reçoit sur le graphe suiveurs-suivis n'est pas la même que celle qui est diffusée par les utilisateurs des listes auxquelles il est souscrit ou qu'il a créées.

Garimella et al. (2018) s'intéressent au phénomène de chambres d'écho dans les réseaux sociaux, c'est-à-dire aux situations dans lesquelles un utilisateur se retrouve uniquement au contact d'opinions avec lesquels il est d'accord. Ils identifient différents types d'acteurs qui favorisent ce phénomène par leurs comportements dans le réseau social.

2.6 Bilan

Dans ce chapitre, nous avons commencé par définir formellement trois types de graphes pour modéliser la relation de co-appartenance d'un ensemble d'entités à un ensemble de groupes : le graphe de co-appartenance attribué $\mathcal{G}_a$, le multigraphe de co-appartenance $\mathcal{G}_{ma}$ et le multigraphe de co-appartenance dynamique $\mathcal{G}_{mad}$. Comme les contributions de cette thèse portent sur l'analyse des multigraphes de co-appartenance, nous avons fait un état de l'art des méthodes de détection de communautés dans les multigraphes et avons présenté les métriques qui permettent d'évaluer leur performance et de comparer leurs résultats. Nous avons fait la distinction entre les méthodes locales et globales pour la détection de communautés dans les multigraphes.

Les méthodes locales permettent de délimiter les frontières d'une seule communauté à partir d'un ensemble de nœuds graines. Ces méthodes cherchent le sous-ensemble de nœuds dans le voisinage des nœuds graines qui optimise une métrique de connectivité locale. Elles sont utiles pour sélectionner des sous-graphes d'intérêt dans les multigraphes de co-appartenance. Mais, les résultats des expériences ne correspondent pas à ce qui est attendu pour les sous-graphes d'intérêt : les communautés locales retournées ne contiennent qu'un nombre très limité de nœuds et aucune méthode locale ne prend en compte les attributs textuels associés aux couches. Pour ces raisons, nous présenterons dans le prochain chapitre deux procédures de sélection de sous-graphe d'intérêt depuis un multigraphe de co-appartenance. Ces procédures ont elles aussi comme point de départ un ensemble de nœuds graines mais les sous-graphes qu'elles extraient sont plus grands que ceux retournés par les méthodes locales de détection de communautés, et leur extraction prend en compte les attributs textuels associés aux couches du multigraphe de co-appartenance.

Les méthodes globales de détection de communautés permettent de partitionner tous les nœuds d'un multigraphe. Elles se basent sur une des trois approches suivantes : aplatissement du multigraphe puis partitionnement du graphe à une couche résultant, consensus entre les partitionnements calculés couche par couche, ou détection directe dans le multigraphe en prenant en compte toutes les couches simultanément. Cependant, aucune de ces méthodes n'utilise l'information sémantique contenue dans les attributs textuels du multigraphe de co-appartenance pour le calcul du partitionnement des nœuds en commu-

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

nautés. Ceci nous a motivé pour développer les méthodes de fouille de données présentées aux chapitres 4 et 5. Ces méthodes d'extraction de motifs symboliques et numériques sous contraintes permettent de détecter les communautés d'un multigraphe en prenant en compte sa structure de connectivité et ses attributs textuels. Nous illustrerons l'utilité de l'information sémantique des attributs textuels pour détecter les communautés d'un multigraphe de co-appartenance en nous comparant aux méthodes de l'état de l'art avec deux exemples : le multigraphe de co-appartenance des utilisateurs Twitter aux listes Twitter et celui de la co-citation des pages Wikipedia pour lequel les vraies communautés sont connues.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Chapitre 3

Sélection de sous-graphes d'intérêt

3.1 Introduction

Nous avons vu dans le chapitre précédent deux familles de méthodes pour l'analyse de la structure des multigraphes. La première famille est celle des méthodes locales de détection de communautés qui permettent de sélectionner un sous-graphe d'intérêt depuis un multigraphe à partir d'un ensemble de nœuds graines. La deuxième famille comporte les méthodes globales de détection de communautés qui sont utilisées pour extraire les communautés i.e. les sous-groupes de nœuds fortement connectés dans le multigraphe. Dans ce chapitre, nous nous intéresserons plus en détails à la sélection de sous-graphes d'intérêt depuis les multigraphes de co-appartenance. Ces multigraphes, que nous utilisons pour modéliser la relation transitive d'appartenance commune d'un ensemble d'entités à un ensemble de groupes, ont des propriétés spécifiques que nous soulignerons dans la suite du chapitre.

Dans, ce chapitre nous introduirons formellement les multigraphes de co-appartenance et présenterons leurs propriétés. Ces multigraphes sont, en particulier, de très grande taille et par conséquent délicats à analyser. Nous proposerons par la suite deux approches pour délimiter des régions liées à des domaines d'intérêt dans ces multigraphes, et nous expliquerons leurs avantages par rapport aux méthodes de la littérature pour la détection de communautés locales dans les multigraphes. Les approches que nous proposons sont basées sur les préférences de l'utilisateur de l'outil à qui il revient de choisir le domaine qui l'intéresse en sélectionnant l'ensemble de nœuds graines qu'il juge pertinents pour ce domaine. Nous illustrons ces méthodes avec un exemple de multigraphe de co-appartenance associé aux listes Twitter à partir duquel nous sélectionnons trois sous-graphes liés à trois domaines d'intérêt : 'l'impact social des organisations à but non lucratif', la 'scène politique française' et 'l'intelligence artificielle'.

3.2 Les multigraphes de co-appartenance

Nous nous intéressons aux multigraphes de co-appartenance tels que présentés dans la définition 6. Voici un exemple de la construction d'un tel graphe à partir des données de co-appartenance $(\mathcal{V}, \mathcal{K}, \mathcal{L}, \mathbf{v}, \mathbf{k}, \mathbf{t})$ telles qu'elles ont été définies en introduction du chapitre 2.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Exemple 3.1. La figure 3.1 représente le multigraphe de co-appartenance issu de l'exemple de données de co-appartenance $(\mathcal{V}, \mathcal{K}, \mathcal{L}, \mathbf{v}, \mathbf{k})$ tel que :

- $\mathcal{V} = \{a, b, c, d, e, f\}$
- $\mathcal{L} = \{L_1, L_2, L_3\}$
- $\mathcal{K} = \{'V', 'R', 'B'\}$
- $\mathbf{v}(L_1) = \{d, e, f\}$
- $\mathbf{v}(L_2) = \{a, e, f\}$
- $\mathbf{v}(L_3) = \{a, b, c, d\}$
- $\mathbf{k}(L_1) = \{'V'\}$
- $\mathbf{k}(L_2) = \{'R'\}$
- $\mathbf{k}(L_3) = \{'B'\}$

où les lettres $'V'$, $'R'$ et $'B'$ correspondent respectivement aux couleurs vert, rouge et bleu.

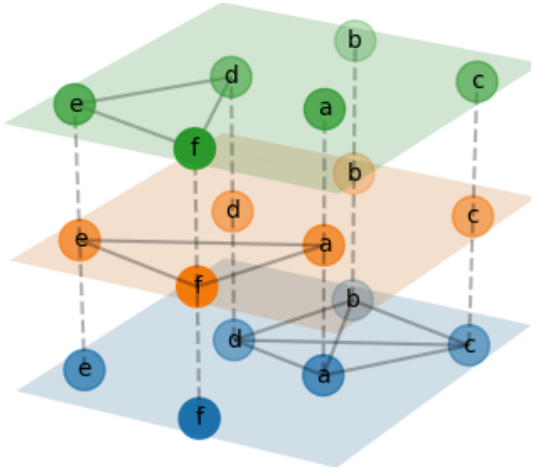


FIGURE 3.1 – Exemple simple de multigraphe de co-appartenance.

Les multigraphes de co-appartenance ont des propriétés qui leur sont propres et qui les distinguent des multigraphes classiques :

(i) ils ont des attributs textuels associés à leurs couches. En effet, l'application $\mathbf{k}$ permet d'associer un sous-ensemble du vocabulaire $\mathcal{K}$ à chaque couche du multigraphe.

(ii) les relations entre nœuds dans chaque couche sont transitives : les nœuds d'une même couche forment une clique dans laquelle chaque nœud est relié à tous les autres nœuds de la couche. La densité des multigraphes de co-appartenance est, par conséquent, grande

(iii) les exemples de multigraphes de co-appartenance, que nous verrons par la suite, contiennent un très grand nombre de nœuds et de couches (d'ordre de grandeur supérieur à 10^6)

(iv) ils possèdent bien plus de couches que les multigraphes classiques ce qui a pour résultat que le ratio $r = \frac{|\mathcal{V}|}{|\mathcal{L}|}$ est bien plus petit pour les multigraphes de co-appartenance.

Notre objectif est d'identifier des communautés dans ces multigraphes de co-appartenance. Mais avant de pouvoir le faire, il est nécessaire de cibler une partie du graphe qui est intéressante pour l'utilisateur. Pour cela, nous présenterons deux procédures pour délimiter, à partir d'un ensemble de nœuds graines, des sous-graphes correspondants à des domaines d'intérêt dans les multigraphes de co-appartenance.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

3.3 Présentation des jeux de données

Nous considérerons, dans les parties expérimentales, les exemples de multigraphes de co-appartenance suivants :

— Listes Twitter [1] : Etant donné un ensemble de listes d'utilisateurs Twitter, un multigraphe de co-appartenance peut être construit en considérant les utilisateurs comme les nœuds du multigraphe et les listes comme ses couches. De plus, le titre et la description des listes Twitter sont à l'origine des attributs textuels associés aux couches du multigraphe.

— Pages Wikipedia [2] : Etant donnés un ensemble de pages Wikipedia et leurs citations, le multigraphe de co-citation associé est défini en disposant chaque page dans une couche et en considérant chaque citation comme un nœud de sorte que les citations d'une même page appartiennent à la même couche et forment une clique. Le titre et le paragraphe d'introduction de chaque page constituent l'attribut textuel de la couche correspondante.

— Articles scientifiques CitHepTh [3] : Similairement au multigraphe de co-appartenance issu des pages Wikipedia, les articles correspondent aux couches et les citations aux nœuds. Ici, les attributs textuels des couches sont issus des résumés des articles scientifiques.

— Acteurs et films IMDb [4] : Soit un jeu de données de films comportant pour chacun son synopsis et la liste des acteurs qui y ont participé. Le multigraphe de co-appartenance qui en est issu a autant de couches que de films. Aussi, dans chaque couche, les acteurs du film correspondant forment une clique et le synopsis du film est considéré comme attribut textuel de la couche.

Graphe	#nœuds	#arêtes	#couches	densité	$r = \frac{\lvert\mathcal{V}\rvert}{\lvert\mathcal{L}\rvert}$
Twitter	18 922 253	_	7 162 384	_	2.64
Wikipedia	1 791 489	28 511 807	1 618 192	15.9	1.11
CitHepTh	27 770	352 807	21 497	12.7	1.29
IMDb	297 705	1 662 429	85 855	5.6	3.47

TABLE 3.1 – Tableau récapitulatif des statistiques des exemples de multigraphes de co-appartenance.

3.4 Méthodologie

Nous avons remarqué précédemment que les exemples de graphes de co-appartenance rencontrés dans la réalité contiennent un très grand nombre de nœuds et de couches : de l'ordre de plusieurs millions. Ceci peut constituer un défi pour leur stockage, leur analyse et notamment pour la détection de leurs communautés. De plus, l'analyste n'est, en général,

1. https://developer.twitter.com/en/docs/twitter-api
2. https://snap.stanford.edu/data/wiki-topcats.html
3. https://snap.stanford.edu/data/cit-HepTh.html
4. https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

intéressé que par une région particulière du graphe de co-appartenance. Nous proposons alors, ici, deux procédures pour construire un sous-graphe $\mathcal{G}_U$ spécifique au domaine d'intérêt de l'analyste. Ce sont des procédures d'expansion qui, à partir d'un ensemble U d'un ou plusieurs nœuds graines a priori représentatifs du domaine d'intérêt, délimitent les frontières du sous-graphe d'intérêt $\mathcal{G}_U$ dans le graphe de co-appartenance $\mathcal{G}$. La délimitation se fait en sélectionnant le sous-ensemble de couches ou de nœuds proches des nœuds graines U et donc en relation avec le domaine d'intérêt choisi par l'analyste.

Les approches locales de détection de communautés dans les graphes simples et dans les graphes multicouches présentées à la section 2.2.2 du chapitre 2 semblent appropriées pour cette tâche de sélection de sous-graphe depuis un multigraphe de co-appartenance. Parmi ces méthodes, il y a notamment celles qui sont basées sur l'expansion d'un ensemble de nœuds "graines" et qui sont présentées dans la section 2.2 du chapitre précédent. Cependant, parmi ces méthodes, celles qui font une sélection automatique des nœuds graines ne peuvent pas être utilisées car l'étape de sélection nécessite le calcul de certaines grandeurs qui impliquent le graphe entier comme la distribution des densités des nœuds ou les longueurs des plus courts chemins. Pour les multigraphes de co-appartenance, la très grande taille du graphe empêche que la sélection automatique ne se fasse en un temps raisonnable.

Parmi les travaux introduits dans l'état de l'art du chapitre 2, les algorithmes qui peuvent réaliser la tâche de sélection de sous-graphes d'intérêt depuis les multigraphes de co-appartenance sont :

— les méthodes locales de détection de communautés dans les graphes simples appliquées à la projection sur une couche du multigraphe de co-appartenance (section 2.2.1)

— les méthodes locales de détection de communautés dans les multigraphes (section 2.2.2)

Mais, à travers les expériences réalisées sur des exemples de multigraphes de co-appartenance, les résultats de ces méthodes ne correspondent pas aux sous-graphes d'intérêt attendus. En effet, ces algorithmes retournent de très petites communautés de nœuds à la différence des procédures qui seront présentées dans la suite du chapitre qui permettent d'extraire des sous-graphes d'intérêt contenant plusieurs centaines de nœuds. Une autre limite des algorithmes de l'état de l'art est qu'ils ne prennent pas en compte conjointement la multiplicité des couches et la présence d'attributs textuels dans la détection des communautés locales.

3.4.1 Pré-traitement des attributs textuels

Pour analyser les attributs textuels en calculant les fréquences des mots employés, un pré-traitement du langage naturel est souvent nécessaire. Dans une première étape, les noms propres sont identifiés et ne sont pas modifiés. Puis, les signes de ponctuation, les majuscules et les mots vides (stop-words) sont mis de côté. Ces derniers sont les mots communs qu'il est inutile d'identifier du fait que leur distribution est uniforme dans n'importe quel document de texte ; ils incluent les prépositions, les articles et les pronoms. Pour finir, une procédure de lemmatisation est appliquée aux mots des attributs textuels pour extraire leur racine et regrouper ceux qui sont d'une même famille. Le vocabulaire résultant $\mathcal{K}$ contient alors les noms propres et les racines des noms communs.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

3.4.2 Première stratégie : *Double couronnes*

L'algorithme *Double couronnes* permet de sélectionner, à partir de l'ensemble de nœuds graines U, un sous-graphe d'intérêt $\mathcal{G}_U$ depuis un multigraphe de co-appartenance $\mathcal{G}$. La procédure cherche les deux couronnes de couches entourant les nœuds graines U et filtre l'ensemble des couches de la deuxième couronne pour ne garder que celles qui sont homogènes avec les nœuds graines et les couches de la première couronne. La procédure se fait en quatre étapes :

(i) Sélection des nœuds graines : Choix par l'analyste d'un ou plusieurs nœuds a priori pertinents pour son domaine d'intérêt : Le point de départ de la construction du sous-graphe associé à un domaine d'intérêt est la sélection d'un petit ensemble de nœuds (entités) jugés, par l'analyste, comme étant au coeur du domaine. Le résultat de cette procédure est sensible au choix initial de ces nœuds graines. Cette sensibilité est un choix assumé car elle permet de reproduire les résultats (en reprenant le même ensemble de nœuds graines), et de laisser le choix à l'analyste de décider et d'explorer différentes options (un ensemble de nœuds graines différent donne une nouvelle représentation d'un point de vue différent du même domaine d'intérêt). Dans tous les cas, la sensibilité est mitigée par le choix de plusieurs nœuds graines. Néanmoins, l'analyste doit respecter certaines lignes directrices pendant le choix de ces nœuds de départ :

- Choisir ces nœuds en se basant sur un consensus d'experts ayant une ample connaissance du domaine d'intérêt
- Préférer les nœuds très connectés dont le degré est grand dans le multigraphe
- Choisir des nœuds graines répartis dans le sous-graphe de sorte à correspondre aux différentes sous-parties du domaine d'intérêt
- Éviter les nœuds graines dont l'appartenance au domaine d'intérêt est vague ou diffuse et privilégier les nœuds qui appartiennent exclusivement au domaine d'intérêt

(ii) Groupes de la première couronne : Sélection d'un sous-ensemble de couches $L_1(U) \subseteq \mathcal{L}$ contenant un ou plusieurs nœuds graines U : $\forall l \in L_1(U),\ \ |\mathbf{v}(l) \cap U| > 0$.

(iii) Calcul des scores : Calcul de la distribution de score $D_{L_1} = \{\mathbf{f}(l), \forall l \in L_1(U)\}$ pour les couches de la première couronne. La fonction score $\mathbf{f} : \mathcal{L} \rightarrow \mathbb{R}$ mesure les fréquences des nœuds et des mots relativement à $L_1(U)$, $\forall l \in \mathcal{L}$:

$$\mathbf{f}(l) = \sum_{u \in \mathbf{v}(l)} \frac{|\{l' \in L_1(U); u \in \mathbf{v}(l')\}|}{|L_1(U)|} + \sum_{k \in \mathbf{k}(l)} \frac{|\{l' \in L_1(U); k \in \mathbf{k}(l')\}|}{|L_1(U)|}$$

(iv) Groupes de la deuxième couronne : Sélection des couches $L_2(U) \subseteq \mathcal{L}$ ayant au moins un nœud en commun avec les couches de $L_1(U)$ et dont le score est supérieur à la médiane de la distribution pour les couches de la première couronne D_{L_1} : i.e. telles que $\forall l \in L_2(U), \mathbf{f}(l) \geqslant median(D_{L_1})$. En particulier, si un nœud ou un mot n'apparaît dans aucune couche de $L_1(U)$, sa fréquence est égale à 0. Ceci permet de ne garder dans la deuxième couronnes $L_2(U)$ que les couches dont les nœuds et les mots utilisés dans les attributs textuels sont similaires à ceux de la première couronne $L_1(U)$ et d'éventuellement négliger les couches éloignées des nœuds graines choisis par l'analyste.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Le sous-graphe $\mathcal{G}_U \subset \mathcal{G}$ obtenu avec la procédure *Double couronnes* est défini par le sous-ensemble de couches $\mathcal{L}_U = L_1(U) \cup L_2(U)$. Aussi, nous notons $\mathcal{V}_U$ le sous-ensemble de nœuds qui apparaissent dans au mois une liste de $\mathcal{L}_U$, et $\mathcal{K}_U$ le vocabulaire des mots apparaissant dans les attributs textuels des listes $\mathcal{L}_U$.

Algorithm 1: DOUBLE COURONNES

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L}, \mathbf{k}), U$
Output: $\mathcal{G}_U = (\mathcal{V}_U, \mathcal{E}_U, \mathcal{L}_U, \mathbf{k})$

$L_1, L_2 = \{\}$
for $l \in \mathcal{L}$ **do**
  if $|U \cap \mathbf{v}(l)| > 0$ **then**
    L_1.add(l)
scores $= \{\mathbf{f}(l), \ \forall l \in L_1\}$ $V_1 = \{\mathbf{v}(l), \ \forall l \in L_1\}$ **for** $l \in \mathcal{L}$ **do**
  if $|V_1 \cap \mathbf{v}(l)| > 0$ **then**
    if $\mathbf{f}(l) > median(scores)$ **then**
      L_2.add(l)
$\mathcal{L}_U = L_1 \cup L_2$
$\mathcal{V}_U = \{\mathbf{v}(l), \ \forall l \in L_U\}$
$\mathcal{E}_U = \{e = (v_1, v_2, l) \in \mathcal{E} | l \in L_U\}$

Le sous-graphe $\mathcal{G}_U$ ainsi construit est lui-même un multigraphe de co-appartenance de plus petite taille que le graphe original $\mathcal{G}$. Un contexte formel $C_{\mathcal{G}_U}$ peut être défini pour $\mathcal{G}_U$ à l'aide du sous-ensemble de couches $\mathcal{L}_U$ et des sous-ensembles de nœuds $\mathcal{V}_U$ et de vocabulaire $\mathcal{K}_U$ qui y sont associés. Dans les prochains chapitres, nous considérerons le cas général d'un multigraphe de co-appartenance que ce soit un sous-graphe local ou le graphe complet.

3.4.3 Deuxième stratégie : *Marche aléatoire*

La procédure *Marche aléatoire* est une alternative à *Double couronnes* pour la sélection d'un sous-graphe d'intérêt depuis le multigraphe de co-appartenance. A la différence de *Double couronnes*, *Marche aléatoire* ne filtre pas les couches du multigraphe mais plutôt les nœuds. Elle utilise une marche aléatoire dont le point de départ est un nœud graine et filtre l'ensemble des nœuds pour ne garder que ceux qui sont les plus visités par la marche aléatoire. La procédure *Marche aléatoire* est constituée de quatre étapes :

(i) Sélection des nœuds a priori pertinents pour le domaine d'intérêt (nœuds graines) U en respectant les mêmes lignes directrices que pour *Double couronnes*

(ii) Pour chaque nœud graine $u \in U$, identifier l'ensemble des nœuds similaires $\mathcal{V}_u$ et construire le sous-graphe engendré par ces nœuds $\mathcal{G}_u : \mathcal{V}_u$ se compose de l'ensemble de nœuds $\mathcal{V}_u^1$ partageant k_{min} listes en commun avec u et de l'ensemble de nœuds $\mathcal{V}_u^2$ partageant au moins k_{min} listes en commun avec les nœuds de $\mathcal{V}_u^1$. Ce choix du nombre minimal de listes communes permet de filtrer les nœuds faiblement connectés aux nœuds graines et de réduire le bruit présent dans les données. Dans nos expériences avec le multigraphe de co-appartenance issu des listes Twitter, nous avons choisi $k_{min} = 3$ mais d'autres valeurs peuvent être testées pour les différents exemples de

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

multigraphes de co-appartenance. Le sous-graphe $\mathcal{G}_u$ associé à un nœud graine $u \in U$ est le sous-graphe engendré par la restriction du graphe de co-appartenance $\mathcal{G}$ aux nœuds $\mathcal{V}_u = \mathcal{V}_u^1 \cup \mathcal{V}_u^2$.

(iii) Débruiter, à l'aide de marches aléatoires partant du nœud graine, les sous-graphes $\mathcal{G}_u$ trouvés à l'étape précédente : les nœuds de $\mathcal{G}_u$ ont des structures de connectivité différentes, certains sont plus ou moins éloignés du nœud graine que d'autres. Les nœuds les plus proches ont plus de chance d'être pertinents par rapport au domaine d'intérêt. A cette étape, nous cherchons alors à garder les nœuds les plus connectés au nœud graine et à retirer ceux qui le sont moins. Pour cela, nous calculons, pour chaque nœud de $\mathcal{G}_u$, le nombre de fois qu'il a été visité par une marche aléatoire de longueur 2 et dont le point de départ est le nœud graine. Les nœuds visités moins de fois que le nombre moyen de visites par nœud sont enlevés de $\mathcal{G}_u$. Nous estimons le nombre de visites pour chaque nœud $v \in \mathcal{V}_u$ en calculant le nombre de chemins de longueur inférieure ou égale à 2 qui partent du nœud graine et qui arrivent à v. Ceci est obtenu en prenant la matrice d'adjacence A de $\mathcal{G}_u$ et en calculant la somme vectorielle $a_u = A_u + A_u^2$ où A_u est la colonne de A correspondant au nœud graine u. Le vecteur a_u contient alors, pour chaque nœud, le nombre de fois qu'il a été visité par la marche aléatoire. En notant $\bar{a}_u$ la moyenne des valeurs dans a_u, nous pouvons filtrer l'ensemble de nœuds $\mathcal{V}_u$ pour ne garder que ceux qui ont un nombre de visites plus grand que $\bar{a}_u$. Ce sous-ensemble de nœuds noté $\mathcal{V}_u^f$ est alors utilisé pour définir un sous-graphe $\mathcal{G}_u^f \subseteq \mathcal{G}_u$ tel que $\mathcal{G}_u^f = (\mathcal{V}_u^f, \mathcal{E}_u^f, \mathcal{L}_u^f, \mathbf{k})$.

(iv) Fusionner les sous-graphes débruités $\mathcal{G}_u^f, \forall u \in U$ de sorte à obtenir un plus grand sous-graphe qui fournit une vision globale des acteurs peuplant le domaine d'intérêt. Comme la densité des multigraphes de co-appartenance est grande et que les nœuds U sont centraux au même domaine d'intérêt, nous nous attendons à ce qu'il y ait beaucoup de chevauchement entre les sous-graphes $\mathcal{G}_u^f$ qui leur sont associés et que le graphe $\mathcal{G}_U = \bigcup_{u \in U} \mathcal{G}_u^f$ engendré par leur jointure soit connexe.

3.5 Résultats et comparaisons

Le réseau social Twitter met à disposition de ses utilisateurs une API publique pour récupérer une partie des données générées par l'activité sur le réseau social. Parmi ces données, celles qui sont relatives aux listes Twitter constituent une source d'information de valeur pour obtenir une caractérisation des utilisateurs faite par un ensemble de tiers. En effet, la fonctionnalité des listes Twitter permet à tout utilisateur du réseau social de rassembler d'autres utilisateurs dans des listes auxquelles il associe un titre et une description. Une date de création est aussi automatiquement attribuée aux listes. Des données de co-appartenance $(\mathcal{V}, \mathcal{K}, \mathcal{L}, \mathbf{v}, \mathbf{k}, \mathbf{t})$ (section 3.2) peuvent alors être définies pour les listes Twitter. L'ensemble des entités $\mathcal{V}$ est celui des utilisateurs du réseau social, l'ensemble des groupes $\mathcal{L}$ correspond à celui des listes Twitter et le vocabulaire $\mathcal{K}$ se compose des mots utilisés dans les titres et descriptions des listes Twitter. Les applications $\mathbf{v}$, $\mathbf{k}$ et $\mathbf{t}$ permettent respectivement de récupérer les utilisateurs appartenant à une listes, de retourner le titre et la description d'une liste et de retourner la date de création d'une liste Twitter. A partir de ces données de co-appartenance, un multigraphe attribué de co-appartenance des utilisateurs Twitter aux

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Algorithm 2: *Marche aléatoire*

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L}, \mathbf{k}), U$
Output: $\mathcal{G}_U = (\mathcal{V}_U, \mathcal{E}_U, \mathcal{L}_U, \mathbf{k})$

$subgraphs = []$
for $seed \in U$ **do**
  $V_1, L_1, V_2, V_f = \{\}$
  $L_{seed} = \{l \in \mathcal{L} \mid seed \in \mathbf{v}(l)\}$ **for** $v \in \mathcal{V}$ **do**
    $L_v = \{l \in \mathcal{L} \mid v \in \mathbf{v}(l)\}$ **if** $\mid L_{seed} \cap L_v \mid \geqslant 3$ **then**
      $V_1.add(v)$
      $L_1 = L_1 \bigcup L_v$
  for $v \in \mathcal{V}$ **do**
    $L_v = \{l \in \mathcal{L} \mid v \in \mathbf{v}(l)\}$ **if** $\mid L_1 \cap L_v \mid \geqslant 3$ **then**
      $V_2.add(v)$
  $\mathcal{G}_{seed} = \textsc{Reduce}(\mathcal{G}, V_1 \cup V_2)$
  $A_{seed} = \textsc{GetAdjMat}(\mathcal{G}_{seed})$
  $\bar{a_u} = \textsc{Mean}(A_{seed}[: seed]$ **for** $v, val \in (\mathcal{V}, A_{seed}[: seed])$ **do**
    if $val \geqslant \bar{a_u}$ **then**
      $V_f.add(v)$
  $subgraphs.append(\textsc{Reduce}(\mathcal{G}_{seed}, V_f))$
$\mathcal{G}_U = \textsc{MergeGraphs}(subgraphs)$

listes Twitter $\mathcal{G}_{ma}^{Twitter} = (\mathcal{V}, E, \mathcal{L}, \mathbf{k_l})$ peut être construit en utilisant la définition 6, de même qu'un multigraphe de co-appartenance dynamique $\mathcal{G}_{mad}^{Twitter} = (\mathcal{V}, E, \mathcal{L}, \mathbf{k_l}, \mathbf{t_l})$ à l'aide de la définition 7.

A l'aide de l'API publique de Twitter, nous avons créé un jeu de données contenant les informations relatives à un ensemble de 10^6 listes dans lesquelles apparaissent yy utilisateurs du réseau social. Le multigraphe de co-appartenance attribué $\mathcal{G}_{ma}^{Twitter}$ construit à partir de ces données est de très grande taille et l'application directe d'un algorithme de détection de communautés serait très coûteuse en termes de calcul ou infaisable avec les ressources disponibles. Nous illustrons, dans la suite, l'utilisation des algorithmes de sélection de sous-graphes d'intérêt sur l'exemple du multigraphe de co-appartenance Twitter.

3.5.1 *Double couronnes*

On construit le sous-graphe d'intérêt $\mathcal{G}_{U_1}$ de la 'scène politique française' en utilisant l'ensemble U_1 contenant trois nœuds graines recouvrant le spectre des partis politiques en France : @laurentwauquiez (droite), @gerardcollomb (centre), @alexiscorbiere (gauche). Nous filtrons ce sous-graphe pour se débarrasser des couches non-informatives : nous ne gardons que celles qui contiennent entre 20 et 200 membres. Le multigraphe résultant est composé de 38, 306 nœuds et 7, 845, 652 liens répartis dans 2, 836 couches.

Ce multigraphe de co-appartenance dépend du choix initial des nœuds graines U_1. Dans un premier temps, nous évaluons la sensibilité de la procédure de sélection de sous-graphes au choix de ces nœuds. Pour cela, nous comparons - à l'aide des indices de Jaccard - les sous-graphes obtenus en modifiant un ou plusieurs éléments de l'ensemble U_1. Les résultats sont

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

présentés dans la Table 3.2. Nous observons que les multigraphes pour lesquels un seul nœud graine a été modifié ont des ensembles de nœuds et des ensembles de couches similaires. Les deux sous-graphes qui n'ont aucun nœud graine en commun ont une similarité de Jaccard de 32% pour les couches et 28% pour les nœuds.

Seeds			Layers similarity	Nodes similarity
@jlmelenchon	@ccastaner	@alainjuppe	0.32	0.28
@alexiscorbiere	@ccastaner	@alainjuppe	0.72	0.62
@jlmelenchon	@ccastaner	@laurentwauquiez	0.55	0.45
@jlmelenchon	@gerardcollomb	@alainjuppe	0.34	0.30
@jlmelenchon	@gerardcollomb	@laurentwauquiez	0.52	0.45
@alexiscorbiere	@ccastaner	@laurentwauquiez	0.90	0.83
@alexiscorbiere	@gerardcollomb	@alainjuppe	0.77	0.70

TABLE 3.2 – Similarités Jaccard entre $\mathcal{G}_{U_1}$ et les multigraphes de co-appartenance obtenus en remplaçant un ou plusieurs nœuds graines.

Par la suite, nous construisons un second sous-graphe d'intérêt $\mathcal{G}_{U_2}$, cette fois pour le domaine de 'l'intelligence artificielle'. L'ensemble U_2 des nœuds graines comporte trois chercheurs de renommée mondiale dans le domaine de l'intelligence artificielle : @ylecun, @geoffreyhinton and @JeffDean. Le multigraphe résultant contient 145, 309 nœuds et 23, 836, 272 liens répartis dans 8, 784 couches.

Afin de visualiser ces multigraphes, nous les projetons sur une seule couche où chaque lien est pondéré avec le nombre de couches dans lesquelles il apparaît. Les nœuds faiblement connectés sont écartés en supprimant les liens de poids inférieur ou égal à 2 et en prenant la plus grande composante connexe du graphe. Les représentations sur une couche des deux exemples de multigraphes de co-appartenance aux listes Twitter ainsi que les communautés Louvain associées sont représentés dans les Figure 3.2 et 3.3.

La projection de $\mathcal{G}_{U_1}$ sur une seule couche résulte en un graphe contenant 6.264 nœuds et 323.560 liens.

En appliquant l'algorithme Louvain, le graphe est partitionné en 14 clusters représentés à la figure 3.2. La légende montre les six clusters les plus importants et donne le pourcentage de nœuds recouverts par chacun d'entre eux.

Les trois nœuds graines utilisés pour la construction du sous-graphe de la 'scène politique française' sont séparés dans trois différents clusters (numérotés 3, 5 et 8), chacun correspondant à la tendance politique du nœud graine qui y appartient comme ça apparaît dans leur caractérisation dans la légende de la figure 3.2. La communauté numéro 8 contient le nœud graine @laurentwauquiez et correspond aux politiciens conservateurs de droite avec le mot-clé "droite" qui fait partie de l'ensemble des mots les plus fréquents associés à la communauté. Le nœud graine @alexiscorbiere appartient au cluster numéro 3 qui correspond à la tendance politique d'extrême gauche comme montré par le nom de son parti politique qui apparaît dans la liste des mots caractérisants la communauté 3. Enfin, le troisième nœud graine @gerardcollomb appartient à la communauté numéro 5 des politiciens de gauche modérée.

Le reste des communautés contiennent des utilisateurs qui sont connectés à la scène politique française mais sans en faire partie (journalistes, société civile, politique internationale).

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

#	%	Color	Keywords	Seeds
0	19.94	Green	journaliste, politique, ex, france, chef	
1	4.45	Orange	député, ministre, van, bruxelles, président	
3	4.18	Rose	gauche, parti, #jlm2017, #franceinsoumise, conseiller	@alexiscorbiere
4	4.18	Green1	sénateur, maire, président, commission, vice	
5	18.14	Blue	président,maire, député, conseiller, paris	@gerardcollomb
8	10.01	Black	président, député, maire, conseiller, droite	@laurentwauquiez
9	36.03	Purple	compte, président, france, politique, député	

FIGURE 3.2 – Sous-graphe construit à partir des nœuds graines de la 'scène politique française'. Les différentes couleurs correspondent aux communautés Louvain. Les mots fréquents dans les biographies des utilisateurs sont utilisés pour caractériser les communautés. Certains mots originalement en français ont été traduits en anglais et sont représentés en italique.

La projection de $\mathcal{G}_{U_2}$ sur une seule couche résulte en un graphe de 8, 414 nœuds et 262, 947 liens.

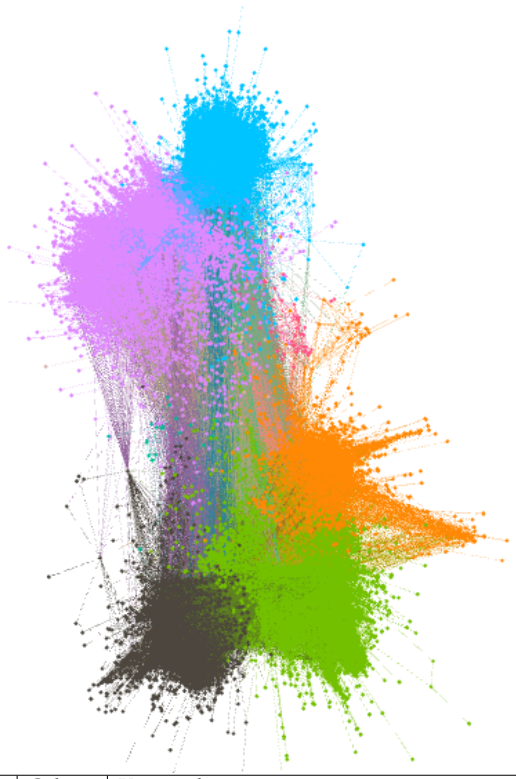
En appliquant l'algorithme Louvain, le graphe est partitionné en 8 clusters représentés dans la figure 3.3. la légende contient les cinq plus grandes communautés accompagnées du pourcentage de nœuds du graphe qu'elles recouvrent.

Pour cet exemple, les trois nœuds graine sont tous regroupés dans un même cluster Louvain : le numéro 2. Ce cluster est celui qui correspond le plus au domaine d'intérêt de 'l'intelligence artificielle'. Sa caractérisation avec les mots les plus fréquents qui lui sont associés comporte les mots-clés : "ai", "machine learning" et "research". Les autres communautés correspondent à des domaines en relation avec l'intelligence artificielle tels que l'analyse de données (cluster numéro 1) et la visualisation des données (cluster numéro 5).

3.5.2 *Marche aléatoire*

Le sous-graphe $\mathcal{G}_{U_3}$ correspond au domaine d'intérêt de 'l'impact social des organisations à but non lucratif' et est extrait du multigraphe de co-appartenance Twitter en utilisant la procédure *Marche aléatoire* et avec un ensemble U_3 de sept nœuds graines : @3ieNews, @MandE_NEWS, @RockefellerFdn, @Keystone_Acc, @GiveWell, @CWTips et @gatesfoundation. L'exécution de l'algorithme Louvain sur ce sous-graphe produit un partitionnement en 30 clusters. Chaque cluster est caractérisé par les 10 mots les plus fréquents utilisés dans

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés



#	%	Color	Keywords	Seeds
0	29.89	Rose	news,tech,technology,founder,author,editor	
1	24.11	Green	data,analytics,science,#bigdata,big,founder	
2	14.7	Orange	learning,machine, ai, research, data, intelligence	All
3	15.41	Blue	science,news,space,research,author,nasa	
5	14.95	Black	data,visualization,open,journalism,editor	

FIGURE 3.3 – Représentation sur une couche du sous-graphe de 'l'intelligence artificielle'. Différentes couleurs sont utilisées pour les communautés Louvain. Les mots fréquents dans les biographies des utilisateurs sont utilisés pour la caractérisation des communautés.

les biographies de ses utilisateurs. La figure 3.4 contient une représentation sur une couche de ce multigraphe avec des couleurs différentes utilisées pour les nœuds de différents clusters. Nous observons que six des sept nœuds graines se retrouvent dans la communauté numéro 27. De plus, la caractérisation sémantique de cette communauté semble très apparentée au domaine d'intérêt. Elle contient des mots comme 'nonprofit', 'charity', 'foundation', 'philanthropy', 'change', 'fundraising'...

Dans (Benabdelkrim et al., 2020a), nous avons proposé une application de l'algorithme *Marche aléatoire* à l'étude des organisations, et plus particulièrement à l'identification des acteurs d'un domaine d'activité. Nous avons utilisé le multigraphe de co-appartenance aux listes Twitter, et nous avons développé une méthodologie qui combine l'algorithme *Marche aléatoire*, l'algorithme de détection de communautés Louvain et des outils de visualisation de graphes dans le but de conduire une analyse empirique du domaine d'activité. L'algorithme *Marche aléatoire* permet de délimiter, dans le multigraphe de co-appartenance, les frontières du sous-graphe correspondant au domaine d'activité. L'algorithme Louvain et les outils de visualisation permettent d'identifier les différentes composantes du domaine d'activité et

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

d'analyser la structure des interactions entre leurs principaux acteurs. L'annexe D donne plus de détails sur l'objectif et les étapes de cette méthodologie.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

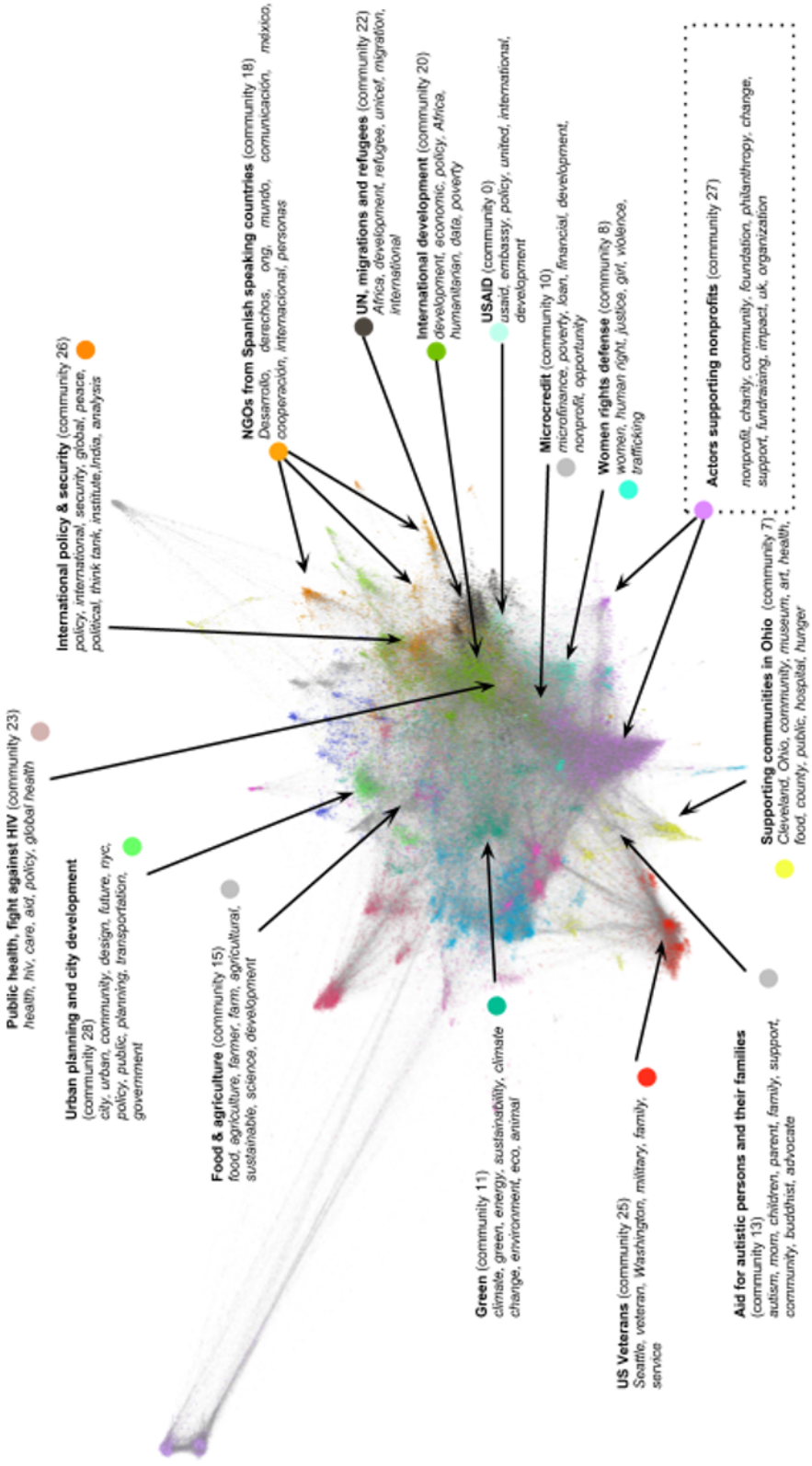


FIGURE 3.4 – Représentation sur une couche du sous-graphe de 'l'impact social des organisations à but non lucratif'. Différentes couleurs sont utilisées pour les communautés Louvain. Les mots fréquents dans les biographies des utilisateurs sont utilisés pour la caractérisation des communautés. Ceci nous permet de mieux comprendre les communautés et de les nommer (en gras).

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

3.5.3 Comparaison des deux approches

Tout d'abord, nous notons que les deux premières étapes sont les mêmes pour les deux procédures : ce sont les étapes de choix des nœuds graines représentatifs du domaine d'intérêt, et de sélection des deux couronnes de nœuds qui les englobent dans le multigraphe de co-appartenance. C'est au cours des étapes suivantes que les procédures diffèrent. L'algorithme *Double couronnes* permet de filtrer les nœuds des couronnes pour ne garder que ceux qui sont très connectés aux nœuds graines. A cet effet, la matrice d'adjacence A qui représente les liens entre les nœuds des deux couronnes est construite ; et les nombres de chemins, de longueur inférieure ou égale à 2, entre les nœuds graines et les nœuds des couronnes sont calculés à l'aide de la matrice $A + A^2$. En fixant un seuil minimal pour ce nombre de chemins, nous obtenons l'ensemble des nœuds les plus fortement connectés aux nœuds graines. Pour l'algorithme *Marche aléatoire*, l'idée de base permettant de filtrer les nœuds des couronnes est quelque peu différente. Elle se base sur la définition d'un profil-type (ensemble de nœuds et de mots du vocabulaire qui caractérisent le domaine d'intérêt représenté par les nœuds graines) à partir des couches de la couronne 1 : les couches qui contiennent les nœuds graines. Ensuite, des scores d'homogénéité entre les couches et le profil-type sont calculés. La médiane de la distribution des scores des couches de la couronne 1 est utilisée comme seuil minimal de score que doivent avoir les couches de la couronne 2 pour être maintenues dans le sous-graphe d'intérêt. Les couches dont le score est inférieur au seuil sont mises de coté car elles sont très différentes des couches contenant les nœuds graines.

Dans le but de comparer leurs performances, nous utilisons les procédures *Double couronnes* et *Marche aléatoire* pour sélectionner un sous-graphe d'intérêt associé au domaine d'intérêt 'Tennis' depuis le multigraphe de co-appartenance Wikipedia. Tout d'abord, nous choisissons un ensemble de nœuds graines représentatifs de ce domaine d'intérêt : 'Wimbledon Championships', 'Rafael Nadal' et 'ATP' puis nous sélectionnons les nœuds et les couches des deux premières couronnes autour de ces nœuds graines. Les sous-graphes obtenus par les deux procédures sont tels que :

Algorithme	temps d'exéc. (s)	#nœuds	#couches	homogénéité sém. h
Double couronnes	2206	29 931	1 344	0.092
Marche aléatoire	1749	31 682	1 674	0.065
ML-LCD (Interdonato et al., 2017)	7265	84	6	0.32
ACLcut (Jeub et al., 2017)	2811	317	14	0.247

Table 3.3 – Tableau récapitulatif des sous-graphes d'intérêt obtenus avec *Double couronnes* et *Marche aléatoire*

L'homogénéité sémantique h est donnée par : $h = 1 - r$ où r est le rapport entre le nombre de mots distincts et le nombre total de mots utilisés dans les attributs des couches du sous-graphe.

Nous notons, tout d'abord, qu'aucune méthode locale de détection de communautés ne prend en compte les attributs textuels contenus dans le multigraphe de co-appartenance. Seule la procédure *Double couronnes* intègre la similarité sémantique entre les descriptions des couches dans le calcul des scores. C'est pour cette raison que l'homogénéité sémantique du sous-graphe d'intérêt obtenu avec *Double couronnes* est supérieure à celle du sous-graphe obtenu avec *Marche aléatoire*.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Nous constatons aussi que les sous-graphes retournés par les méthodes locales de détection de communautés sont de petite taille et très ciblés. Par exemple, Le nombre de nœuds dans les sous-graphes calculés avec MLLCD est en moyenne égale à 84 et le nombre de couches à 6. A l'inverse, les procédures que nous proposons retournent des sous-graphes de taille bien plus grande. Ces derniers seront utilisés pour représenter le sous-graphe représentant le domaine d'intérêt associé aux nœuds graines et identifier ses communautés.

3.5.4 Conclusion

Les multigraphes de co-appartenance sont, en pratique, de très grande taille et contiennent souvent des acteurs issus de différents domaines d'intérêt imbriqués entre eux. Dans la majorité des cas, l'analyste n'est intéressé que par un domaine d'intérêt particulier. Il est alors utile de sélectionner le sous-graphe qui correspond à ce qu'il cherche. Ceci offre l'avantage de rendre les calculs moins fastidieux et les résultats plus compréhensibles et conformes aux attentes de l'analyste. Nous avons observé que les sous-graphes obtenus par les méthodes locales de détection de communautés de l'état de l'art n'ont pas les caractéristiques recherchées pour les sous-graphes d'intérêt : la taille des communautés locales est très petite et leur détermination ne prend pas en compte l'information sémantique contenue dans les attributs des couches du multigraphe de co-appartenance. A l'inverse, les procédures proposées *Double couronnes* et *Marche aléatoire* retournent des sous-graphes d'une plus grande taille. De plus, l'algorithme *Double couronnes* utilise les attributs textuels pour la sélection du sous-graphe d'intérêt. Pour cette raison, ce sera cet algorithme qui sera utilisé dans la suite. Dans les prochains chapitres, nous développerons des méthodes de détection de communautés que nous appliquerons à ces sous-graphes d'intérêt pour identifier les acteurs principaux du domaine d'intérêt et la manière avec laquelle ils s'organisent dans le multigraphe de co-appartenance.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Chapitre 4

Détection de communautés par extraction de motifs symboliques

4.1 Introduction

La sélection de sous-graphes d'intérêt avec les algorithmes du chapitre 3 engendre des multigraphes de co-appartenance de plus petite taille que le graphe originel. Dans ce type de graphes, tous les nœuds d'une même couche sont connectés et forment des cliques : les relations entre les nœuds d'une même couche sont transitives et une association entre les ensembles de couches et de nœuds peut être faite. De plus, comme les attributs textuels sont associés aux couches du multigraphe, une association peut aussi être faite entre le vocabulaire de mots issu des attributs textuels et l'ensemble des couches. Il est alors possible de définir une relation binaire entre l'ensemble des couches d'un coté et l'ensemble des nœuds et le vocabulaire de l'autre. Cette relation binaire peut être représentée dans un jeu de données à attributs booléens dans lequel les objets sont les couches du multigraphe de co-appartenance et les attributs déterminent l'appartenance d'un nœud à une couche ou l'utilisation d'un mot du vocabulaire dans l'attribut textuel associé à la couche. Dans le formalisme de l'Analyse Formelle de Concepts (AFC), ce type de jeux de données à attributs booléens est appelé contexte formel. Dans ces contextes formels, les motifs sont définis comme des sous-ensembles d'attributs, nous parlons alors de motifs symboliques. De plus, ces motifs peuvent être ordonnés avec la relation d'ordre partiel de l'inclusion d'ensembles $\subseteq$. L'ensemble des motifs symboliques muni de sa relation d'ordre est appelé le langage de motifs du contexte formel.

Dans le contexte formel issu d'un multigraphe de co-appartenance, les motifs symboliques sont des paires (V, K) avec $V \subseteq \mathcal{V}$ et $K \subseteq \mathcal{K}$. Ces motifs correspondent à des sous-graphes sous forme d'ensembles de couches dans le multigraphe de co-appartenance. L'objectif est de chercher ceux qui correspondent à des communautés de nœuds homogènes : c'est-à-dire des nœuds qui appartiennent aux mêmes couches ou à des couches dont les attributs textuels ont une certaine similarité sémantique. A cet effet, nous définissons une fonction de qualité sur les motifs pour estimer leur homogénéité, puis nous en déduisons une contrainte de qualité minimale que nous utilisons pour filtrer l'ensemble de motifs et ne garder que ceux qui correspondent à des communautés dans le multigraphe de co-appartenance.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Nous présenterons, dans la suite du chapitre, les principes de l'extraction de motifs symboliques sous contraintes et les algorithmes de la littérature qui permettent de réaliser cette tâche. Nous introduirons aussi les propriétés de fermeture qui permettent à la fois d'accélérer l'énumération des motifs en limitant leur nombre et de filtrer l'ensemble de motifs résultant en diminuant sa redondance. Ensuite, nous présenterons en détails la méthode de construction du contexte formel à partir d'un multigraphe de co-appartenance, et les algorithmes exhaustifs et d'échantillonnage proposés qui permettent d'en extraire des motifs symboliques. Les approches exhaustives permettent d'extraire tous les motifs fermés vérifiant la contrainte de qualité minimale, ou bien le sous-ensemble des top-k motifs fermés et diversifiés. Les méthodes d'échantillonnage sont utilisées quand la taille du contexte formel ne permet pas d'énumérer la totalité des motifs. Elles permettent de tirer aléatoirement, avec un coût de calcul très faible, des motifs symboliques selon une distribution de probabilité proportionnelle à la mesure de qualité. Dans la section 4.4, nous présenterons les résultats et nous introduirons les métriques utilisées pour l'évaluation de la performance des algorithmes et pour les comparaisons aux algorithmes de détection de communautés de la littérature.

4.2 Extraction de motifs symboliques sous contraintes

Nous rappelons, dans cette première section, les fondamentaux de l'Analyse Formelle de Concepts (AFC) et son utilisation pour l'extraction de motifs symboliques sous contraintes. Nous renvoyons à l'annexe B pour les définitions des treillis et de leurs propriétés et leur relation avec les motifs symboliques. Nous présentons un opérateur de fermeture pour ces motifs et soulignons son utilité pour limiter la redondance dans les résultats. Puis, à l'aide d'une relation d'ordre partiel entre les motifs, nous construisons le langage de motifs symboliques fermés. Enfin, nous mentionnons des propriétés de contraintes comme la monotonicité et l'antimonotonicité qui permettent d'accélérer l'extraction du sous-ensemble de motifs vérifiant les contraintes depuis le langage de motifs symboliques fermés.

Tout d'abord, les motifs symboliques sont extraits de jeux de données à attributs booléens.

Définition 8. *Un **jeu de données binaire** D est une paire $D = (G, M)$ où G est un ensemble d'objets et M est un ensemble d'attributs booléens de domaine $\{0, 1\}$. Un attribut $m \in M$ est une application $m : G \to \{0, 1\}$ qui associe à chaque objet g sa valeur pour l'attribut m notée $m(g) \in \{0, 1\}$.*

Un jeu de données binaire permet de définir un contexte formel en utilisant une relation binaire pour regrouper les associations entre les objets et les attributs booléens du jeu de données.

Définition 9. *Un **contexte formel** $C = (G, M, I)$ est composé de deux ensembles G et M et d'une relation binaire I entre G et M représentée par le jeu de données. G et M sont appelés les ensembles **objets** et **attributs** respectivement du contexte C. gIm ou $(g, m) \in I$ veut dire que l'objet g possède l'attribut m.*

Exemple 4.1. On considère un exemple de jeu de données binaire $D_1 = (G_1, M_1)$où M_1 est un ensemble d'attributs booléens et tel que représenté dans la table 4.1. Les attributs booléens $m \in M_1$ sont à valeur dans $\{0, 1\}$. La relation binaire I_1 déduite du jeu de données D_1 permet de définir un contexte formel $C_1 = (G_1, M_1, I_1)$.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

	m_1	m_2	m_3
g_1	0	1	1
g_2	0	0	1
g_3	1	1	1
g_4	1	1	0

TABLE 4.1 – D_1 : Jeu de données à attributs booléens

Deux opérateurs découlent de la définition d'un contexte formel et font le lien entre les sous-ensembles d'objets et les sous-ensembles d'attributs. Ce sont les opérateurs d'extension et d'intention :

Définition 10. ***L'opérateur d'extension****, noté ext, associe à chaque ensemble d'attributs $B \subseteq M$ l'ensemble d'objets qui contiennent ces attributs :*

$$ext : \mathcal{P}(M) \rightarrow \mathcal{P}(G), B \rightarrow \{g \in G | \forall m \in B, gDm\}$$

Définition 11. ***L'opérateur d'intention****, noté int, associe à chaque ensemble d'objets $A \subseteq G$ l'ensemble des attributs communs à ces objets :*

$$int : \mathcal{P}(G) \rightarrow \mathcal{P}(M), A \rightarrow \{m \in M | \forall g \in A, gDm\}$$

Exemple 4.2. Dans le contexte formel $C_1 = (G_1, M_1, I_1)$ (voir jeu de données de la table 4.1), nous avons les relations suivantes :

$$ext(\{m_3\}) = \{g_1, g_2, g_3\}$$
$$ext(\{m_1, m_2, m_3\}) = \{g_3\}$$
$$int(\{g_2, g_4\}) = \emptyset$$
$$int(\{g_1, g_2\}) = \{m_3\}$$

4.2.1 Le langage de motifs symboliques

Étant donné un contexte formel $C = (G, M, I)$, des motifs symboliques peuvent être extraits à partir de l'ensemble $\mathcal{P}(M)$.

Définition 12. *Un **motif symbolique** B dans un contexte formel $C = (G, M, I)$ est un sous-ensemble d'attributs : $B \subseteq M$.*
***Le langage de motifs** est l'ensemble des parties de M muni de la relation d'ordre partielle de l'inclusion d'ensembles $\subseteq$, il est noté $(\mathcal{P}(M), \subseteq)$.*
***L'image** d'un motif B est l'ensemble $A \subseteq G$ obtenu en appliquant l'opérateur ext : $ext(B) = A$, c'est-à-dire telle que $\forall a \in A$ et $\forall b \in B$, $(a, b) \in D$.*

Exemple 4.3. Les motifs $p_1 = \{m_1, m_2\} \in \mathcal{P}(M_1)$ et $p_2 = \{m_1, m_2, m_3\} \in \mathcal{P}(M_1)$ sont des motifs symboliques pour le contexte formel C_1 (voir jeu de données de la table 4.1).

Les motifs d'un contexte formel sont très nombreux et redondants. Il est possible de définir des classes d'équivalence de motifs en rassemblant ceux qui ont la même image.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

L'élément maximal (par rapport à la relation $\subseteq$) d'une classe d'équivalence de motifs est appelé un motif fermé.

Exemple 4.4. Dans le contexte formel C_1, soient les motifs p et p' tels que $p = \{m_1\}$ et $p' = \{m_1, m_2\}$, nous avons $ext(p) = ext(p') = \{g_3, g_4\}$. Les deux motifs ont les mêmes images, nous pouvons alors définir la classe d'équivalence de motifs $\{p, p'\}$.

La composition des deux opérateurs d'extension et d'intention permet de définir deux opérateurs de fermeture pour les motifs symboliques d'un contexte formel.

Définition 13. *Les applications ext∘int et int∘ext sont des opérateurs de fermeture pour les ensembles G et M respectivement. Les points fixes de ext ∘ int (c'est-à-dire $A \subseteq G$ tels que ext ∘ int(A)=A) sont appelés* ***des extensions****. Et, les points fixes de int ∘ ext sont appelés* ***des intentions*** *ou* ***des motifs fermés****.*

Exemple 4.5. Le motif symbolique $p = \{m_1\}$ n'est pas un motif fermé :

$$int \circ ext(p) = int(\{g_3, g_4\}) = \{m_1, m_2\} \neq p$$

Le motif symbolique $p' = \{m_1, m_2\}$ est fermé : $int \circ ext(p') = int(\{g_3, g_4\}) = p'$.

Les motifs fermés et leurs images constituent les concepts formels d'un contexte.

Définition 14. *Un* ***concept formel*** *dans un contexte $C = (G, M, D)$ est une paire (A, B) où B est un motif fermé, et A est l'image de B : $int(A) = B$. Les ensembles A et B sont appelés, respectivement,* ***l'extension*** *et* ***l'intention*** *du concept formel (A, B).*

Exemple 4.6. Dans le contexte formel C_1, $(\{g_3, g_4\}, \{m_1, m_2\})$ est un concept formel.

En définissant une relation d'ordre sur les concepts formels, nous obtenons un langage de motifs symboliques fermés pour le contexte formel $C = (G, M, D)$

Définition 15. *L'ensemble des concepts formels du contexte C forme un treillis complet appelé* ***treillis des concepts*** *de C et noté $\underline{\mathfrak{B}}(C)$. Les concepts sont ordonnés par la relation $\leqslant$:*

$$(A_1, B_1) \leqslant (A_2, B_2) \iff A_1 \subseteq A_2 \iff B_2 \subseteq B_1$$

Le langage de motifs du contexte formel $C = (G, M, D)$ est alors le treillis de concepts $\underline{\mathfrak{B}}(C)$.

Exemple 4.7. Pour l'exemple de contexte formel C_1 (table 4.1), le treillis de concepts contient 5 éléments avec la relation d'ordre partiel $\leqslant$ et est représenté dans la figure 4.1.
Les deux motifs restants $\{m_1\}$ et $\{m_1, m_3\}$ ne sont pas fermés dans C_1 et ne font donc pas partie du treillis de concepts.

4.2.2 Les contraintes

Après avoir choisi de définir le langage des motifs d'un contexte formel $C = (G, M, I)$ comme le treillis des concepts $\underline{\mathfrak{B}}(C)$, l'étape suivante pour l'extraction de motifs symboliques consiste à identifier parmi les motifs du langage ceux qui sont intéressants et utiles. L'intérêt et l'utilité dépendent des besoins de l'utilisateur de l'outil d'extraction de motifs et ceux-ci peuvent être exprimés sous forme de contraintes.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

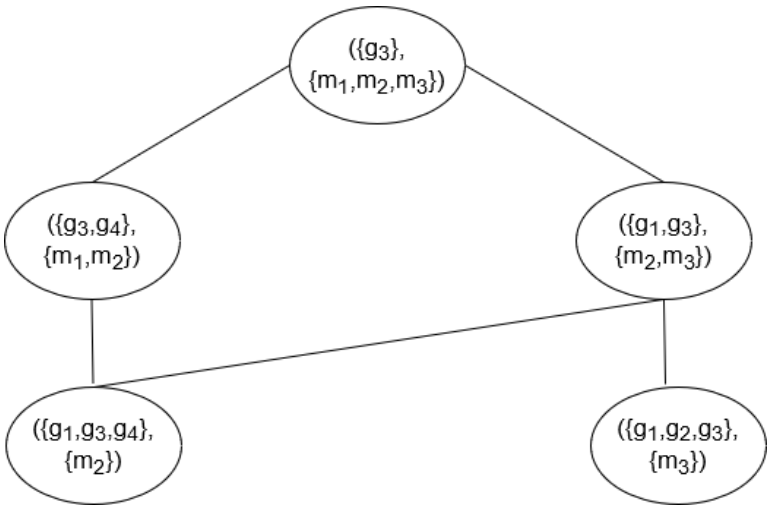


FIGURE 4.1 – Treillis de concepts $\underline{\mathfrak{B}}(C_1)$.

Contraintes de fréquence

La tâche d'extraction de motifs symboliques fréquents est une des tâches les plus communes en data mining. Elle consiste à compter le nombre de fois que le motif est présent dans les données. Pour pouvoir réaliser cette tâche, nous définissons le support d'un motif :

Définition 16. *Dans un contexte $C = (G, M, I)$, le support d'un motif $p \subseteq M$ est noté $sup(p)$ et est défini comme le nombre d'objets contenant p dans leur description :*

$$sup(p) = |ext(p)| = |\{g \in G | \forall m \in p, gIm\}|$$

Exemple 4.8. Dans le contexte formel C_1 (voir jeu de données de la table 4.1), le support des motifs fermés $p_1 = \{m_1, m_2\}$ et $p_2 = \{m_1, m_2, m_3\}$ est :

$$sup(p_1) = |ext(p_1)| = |\{g_3, g_4\}| = 2 \qquad sup(p_2) = |ext(p_2)| = |\{c\}| = 1$$

Le support d'un motif est la taille de son image. Il permet de définir deux contraintes sur l'ensemble des motifs : une contrainte de support minimal qui rend possible l'extraction des motifs fréquents, et une contrainte de support maximal qui permet de définir des motifs rares. Ces deux contraintes nécessitent de fixer un paramètre représentant le seuil de support minimal (maximal) au-delà duquel un motif est considéré fréquent (rare). Pour un seuil de support minimal $\min_{supp}$ et un seuil de support maximal $\max_{supp}$, nous pouvons définir des contraintes de fréquence et de rareté P_{fr} et P_{ra} pour les motifs du langage $p \in \underline{\mathfrak{B}}(C)$

$$P_{fr} = sup(.) \geqslant min_{supp} \qquad P_{ra} = sup(.) \leqslant max_{supp}$$

Exemple 4.9. Dans le contexte formel C_1, et avec $min_{supp} = 2$ le motif fermé $p_1 = \{m_1, m_2\}$ est un motif fréquent pour lequel la contrainte P_{fr} est vraie. Le motif fermé $p_2 = \{m_1, m_2, m_3\}$ n'est pas un motif fréquent, son support est strictement inférieur à min_{supp} et il ne vérifie pas P_{fr}.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Contrainte de discrimination de classes

Pour l'extraction des motifs discriminants, nous assumons que les objets G du contexte $C = (G, M, I)$ appartiennent à deux classes G^+ et G^-. De plus, nous définissons, pour un motif p, une fonction $f(n^+(p), n^-(p))$ où n^+ est le nombre d'objets de G^+ couverts par p et n^- le nombre d'objets de G^- dans l'extension de p. Cette fonction retourne un score qui mesure à quel point le motif p permet de discriminer les objets de G^+ des objets de G^-, f peut être la mesure χ^2 par exemple. La contrainte qui en découle permet d'extraire les motifs qui caractérisent une des deux classes et qui sont absents dans l'autre. Elle est définie, en utilisant un seuil minimal θ, comme suit :

$$P_{dis} = f(|ext(p) \cap G^+|, |ext(p) \cap G^-|) \geqslant \theta$$

Exemple 4.10. On reprend le contexte formel C_1 dont le jeu de données est représenté dans la table 4.1 et nous supposons que les deux premiers objets g_1 et g_2 sont dans la classe $+$ et que les deux autres g_3 et g_4 sont dans la classe $-$. Nous fixons le seuil $\theta = 2$ et nous choisissons $f = \chi^2$, nous pouvons alors sélectionner le sous-ensemble de motifs fermés discriminants pour lesquels la contrainte P_{dis} est vraie. Soient les motifs $p = \{m_1, m_2\}$ et $p' = \{m_2, m_3\}$, nous avons :

$$f(n^+(p), n^-(p)) = \chi^2(0, 2) = 4 \geqslant \theta \Rightarrow p \text{ vérifie } P_{dis}$$

$$f(n^+(p'), n^-(p')) = \chi^2(1, 1) = 0 \leqslant \theta \Rightarrow p' \text{ ne vérifie pas } P_{dis}$$

Propriétés des contraintes

L'extraction de motifs symboliques sous contraintes peut être accélérée en tirant profit de certaines propriétés des contraintes.

Définition 17. *Une contrainte est **monotone** si et seulement si quand elle est vraie pour un motif symbolique S, elle est vrai pour n'importe quel motif S′ plus grand que S pour la relation d'ordre $\subseteq$: tel que $S \subseteq S'$.*

Exemple 4.11. La contrainte de rareté P_{ra} est monotone. En effet, $\forall p \subseteq p' \subseteq M$, nous avons $sup(p) \geqslant sup(p')$. Par ailleurs, si P_{ra} est vraie pour p alors $sup(p) \leqslant max_{supp}$. Nous en déduisons que $sup(p') \leqslant max_{supp}$ et donc que p' vérifie P_{ra}.

Définition 18. *Une contrainte est **anti-monotone** si et seulement si quand elle est vraie pour un motif symbolique S, elle est vrai pour n'importe quel motif S′ plus petit que S pour la relation d'ordre $\subseteq$: tel que $S' \subseteq S$.*

Exemple 4.12. La contrainte de fréquence P_{fr} est anti-monotone. En effet, $\forall p \subseteq p' \subseteq M$, nous avons $sup(p) \geqslant sup(p')$. Par ailleurs, si P_{fr} est vraie pour p' alors $sup(p') \geqslant min_{supp}$. Nous en déduisons que $sup(p) \geqslant min_{supp}$ et donc que p vérifie P_{fr}.

L'utilisation d'une de ces deux propriétés diminue le coût de calcul de l'extraction de motifs sous contraintes. Elles permettent de déduire rapidement la vérification d'une contrainte par un motif en utilisant la relation d'ordre partiel du langage de motifs fermés.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

4.2.3 Les algorithmes de calcul

4.2.3.1 Calcul exhaustif de motifs

Les opérateurs de fermeture pour les motifs symboliques présentés à la définition 13 sont utilisés pour énumérer d'une manière exhaustive et non redondante les sous-groupes d'objets dans un contexte formel. En effet, à chaque sous-groupe d'objets correspond un unique motif symbolique fermé. Dans (Kuznetsov and Obiedkov, 2002), les auteurs fournissent une liste d'algorithmes d'énumération des motifs fermés d'un contexte formel et les classifient selon les quatre critères :

(1) Ordre de complétion :
   - incrémental : la génération des motifs fermés par l'algorithme respecte un ordre préétabli sur les objets du contexte formel. A chaque étape i de son exécution, l'algorithme produit l'ensemble des concepts générés par les i premiers objets du contexte ;
   - par batchs : l'algorithme produit l'ensemble des motifs fermés dans un ordre indifférent à celui des objets dans le contexte

(2) Sens de complétion :
   - Une approche de bas en haut qui construit le treillis de concepts en commençant par le plus petit concept (selon l'ordre partiel associé aux concepts) qui est celui dont l'intention est la plus générale de sorte que son extension comporte tous les objets du contexte. les concepts qui ont les ensembles d'extension les plus petits.
   - Une approche de haut en bas qui construit le treillis en commençant par le concept le plus grand (selon l'ordre partiel associé aux concepts) qui est celui dont l'intention est la plus spécifique de sorte que son extension comporte le moins d'objets en comparaison aux autres concepts extraits du contexte.

(3) Comment générer un prochain concept à partir du concept courant ?
   - Intersection de l'intention du concept courant avec la description d'un objet
   - Intersection des descriptions d'un ensemble d'objets donné
   - Intersection entre les intentions de concepts générés aux étapes précédentes
   - Rajout de nouveaux attributs aux intentions des concepts générés aux étapes précédentes

(4) Comment éviter de multiples générations d'un même concept ?
   - Maintien de structures de données spécifiques pour le stockage des concepts
   - Stockage des concepts dans des structures de données disjointes pour accélérer leur récupération
   - Utilisation de fonctions de hashage pour une recherche plus efficace dans l'ensemble des concepts générés aux étapes précédentes
   - Cache d'attributs : l'unicité d'un concept est déterminée par le calcul de son intersection avec des attributs enregistrés dans un cache
   - Test de canonicité basé sur un ordre lexicographique choisi pour les objets ou les attributs du contexte

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Les auteurs comparent ces algorithmes en terme de temps d'exécution en utilisant des jeux de données synthétiques générés aléatoirement et des contextes issus de sources de données "classiques" (couramment utilisées dans la communauté de l'analyse des données) comme par exemple le jeu de données SPECT Heart disponible en ligne sur le UCI Machine Learning Repository [1]. Ils identifient les méthodes les plus performantes en fonction de la taille et de la densité du contexte considéré. L'algorithme Godin (Godin et al., 1995) est le plus rapide pour les contextes parcimonieux de petite taille. Les algorithmes Norris (Norris, 1978), NextClosure (Ganter et al., 1987) et Close-by-One (Kuznetsov, 1993) sont, eux, performants dans les cas où le contexte est grand et dense.

L'algorithme Godin (Godin et al., 1995) est une méthode incrémentale qui parcourt l'ensemble des concepts d'un contexte de bas en haut. Il génère les nouveaux concepts en calculant les intersections entre les intentions de concepts déjà découvertes aux étapes précédentes et des descriptions d'objets. La redondance y est évitée en stockant les concepts générés précédemment dans des structures de données disjointes. De plus, l'algorithme utilise une fonction de hashage pour accélérer la récupération des données. L'algorithme Norris (Norris, 1978) est aussi incrémental et il utilise la même approche que Godin pour la génération des nouveaux concepts. L'algorithme parcourt de haut en bas le treillis des concepts et enregistre les concepts générés dans une liste de sorte à éviter la génération d'un même concept à de multiples reprises. L'algorithme NextClosure (Ganter et al., 1987) trouve de nouveaux concepts en parcourant les sous-ensembles d'objets dans le contexte. L'intention de chaque concept est alors calculée comme l'intersection des descriptions des objets du sous-ensemble. NextClosure évite les générations multiples d'un même motif en utilisant un test de canonicité basé sur un ordre lexicographique choisi pour les objets du contexte. L'algorithme Close-by-one (ou CbO) (Kuznetsov, 1993) est une des méthodes les plus utilisées pour l'énumération exhaustive des motifs d'itemsets fréquents fermés d'un contexte. Il permet de faire un parcours en profondeur du treillis de concepts formels. A chaque étape de son exécution, l'algorithme génère un nouveau motif fermé en calculant l'intersection entre l'intention du concept de l'étape précédente avec la description d'un objet du contexte. La répétition des mêmes concepts est évitée en utilisant un test de canonicité semblable à celui de NextClosure. Par la suite, d'autres versions de cet algorithme ont été proposées dans la littérature. Par exemple, la méthode FCbO (Outrata and Vychodil, 2012) est une version parallélisable de CbO. Elle nécessite moins de ressources de calcul et utilise un test de canonicité amélioré pour réduire le nombre de concepts formels visités à de multiples reprises. InCLose (Andrews, 2011) est une autre implémentation haute performance de l'algorithme CbO qui tire profit de nouvelles techniques d'optimisation et de pré-traitement des données.

4.2.3.2 Calcul heuristique

Quand l'énumération des concepts d'un contexte est infaisable, il existe certaines méthodes heuristiques qui permettent d'obtenir un ensemble de motifs non exhaustif en un temps raisonnable. Ces méthodes utilisent différentes heuristiques pour diriger la recherche dans l'espace de motifs et retournent un résultat sans effectuer un parcours complet de tous les éléments de l'espace. Nous pouvons distinguer deux approches : la recherche en faisceaux et les algorithmes génétiques (ou évolutifs).

1. https ://archive.ics.uci.edu/ml/datasets/SPECT+Heart

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

La recherche en faisceaux (Leeuwen and Knobbe, 2012) est basée sur un parcours incomplet du treillis de concepts. A chaque niveau de ce treillis, seul un sous-ensemble contenant des motifs diversifiés et dont la mesure de qualité est supérieure à un certain seuil est maintenu pour explorer les prochains niveaux. A cet effet, des fonctions de qualité pour les concepts sont définies et trois stratégies pour la sélection de concepts diversifiés sont présentées. Les stratégies proposées sont : limiter la redondance des intentions des concepts, ou limiter la redondance des extensions des concepts, ou encore en choisissant le sous-ensemble de concepts qui permet d'obtenir la meilleure compression du jeu de données. Par ailleurs, dans (Lavrač et al., 2004, Meeng et al., 2014), les auteurs proposent de rendre l'utilisation de la recherche en faisceaux plus accessible aux non-experts en utilisant la mesure ROC. En effet, la recherche en faisceaux classique nécessite de choisir des valeurs pour un certain nombre de paramètres et ces choix ont une grande influence sur les résultats obtenus. Les auteurs remplacent, dans ce travail, les paramètres du beam-search par une procédure d'optimisation de la mesure ROC. Cette procédure permet de trouver automatiquement un équilibre entre exploration et exploitation du treillis de concepts.

Les algorithmes génétiques (ou évolutifs) permettent aussi d'extraire des motifs qui satisfont certaines contraintes sans passer par l'énumération exhaustive des motifs fermés d'un contexte. Les méthodes de cette famille utilisent des idées issues de la génétique pour explorer d'une manière efficace des espaces complexes de très grande taille. La méthode présentée dans (Del Jesus et al., 2007) est une approche génétique qui utilise comme langage de motifs les règles floues disjonctives sur les attributs. A chaque étape de l'évolution, ces règles évoluent en subissant des mutations aléatoires ce qui permet une exploration efficace de l'espace de recherche des motifs. Comme pour les autres algorithmes génétiques, la méthode nécessite une population d'individus à faire évoluer, qui sont ici les règles floues disjonctives, et une fonction de fitness qui permet d'estimer la qualité des individus. Des opérateurs d'évolution comme la mutation ou la recombinaison sont utilisés pour générer de nouveaux individus à partir des individus existants. (Lucas et al., 2017) est un autre exemple de cette famille de méthodes évolutives, l'algorithme a été conçu pour l'extraction des top-k motifs discriminants d'un contexte formel. Plus de détails sur le fonctionnement des algorithmes génétiques sont disponibles dans (Freitas, 2002).

4.2.3.3 Calcul par échantillonnage

L'échantillonnage de motifs est une autre alternative à l'énumération exhaustive. Les méthodes d'échantillonnage permettent de tirer aléatoirement les motifs d'itemsest d'un contexte selon une distribution de probabilité proportionnelle à la mesure de qualité définie pour les motifs. Les plus anciens algorithmes d'échantillonnage de motifs comme (Boley et al., 2010) utilisent le cadre formel des MCMC (Markov chain Monte Carlo). Ils définissent une chaîne de Markov sur l'espace d'états constitué des concepts d'itemsets et utilisent une procédure de génération pour parcourir cet espace. La chaîne de Markov est définie de façon à ce qu'elle possède une distribution stationnaire proportionnelle à la mesure de qualité. Sous réserve que le nombre de transitions d'états effectuées dans la chaîne de Markov soit assez grand pour garantir sa convergence vers la distribution stationnaire, l'algorithme proposé dans (Boley et al., 2010) permet d'échantillonner des motifs d'itemsets avec une distribution de probabilité proportionnelle à la mesure de qualité.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Les autres méthodes d'échantillonnage qui n'utilisent pas le carde MCMC sont des méthodes heuristiques. Parmi elles, l'algorithme 2-step sampling (Boley et al., 2011, 2012) construit en deux temps une distribution de probabilité sur les motifs symboliques proportionnelle à leur mesure de qualité. La distribution est construite de façon à favoriser le tirage des motifs avec les caractéristiques recherchées : fréquence, support, surface, fréquence disjonctive ... La méthode se décompose en deux étapes : une première étape dans laquelle nous tirons aléatoirement une ou plusieurs objets du contexte puis une seconde étape où nous extrayons un motif commun aux objets échantillonnés à la première étape. L'algorithme Maximal sampling (Moens and Goethals, 2013) permet aussi l'échantillonnage de motifs d'itemsets. Il est basé sur un tirage itératif d'attributs pour construire des motifs symboliques maximaux, c'est-à-dire des motifs qui ne peuvent être augmentés avec aucun autre attribut sans mettre à mal les propriétés recherchées pour les motifs. Enfin, l'algorithme FLEXICS (Dzyuba et al., 2016) est un autre exemple de méthode d'échantillonnage de motifs d'itemsets. Il est basé sur la programmation par contraintes : elle considère les problèmes d'extraction de motifs comme des problèmes de satisfaction de contraintes. Elle utilise des contraintes XOR aléatoires pour partitionner l'espace de recherche de sorte à obtenir des cellules de plus petite taille. L'échantillonnage d'un motif se fait alors en tirant aléatoirement une cellule puis en exécutant un algorithme d'extraction de motifs sur les éléments de la cellule. La méthode est flexible en terme de fonctions de qualité et de contraintes qu'elle peut prendre en charge, et elle offre des garanties sur la précision et l'efficacité de l'échantillonnage.

Ces méthodes seront comparées entre elles et avec l'énumération exhaustive dans la tâche d'extraction de communautés depuis un multigraphe de co-appartenance dans la Section 4.4.

4.2.4 Filtrage des motifs

L'analyste qui utilise un outil d'extraction de motifs exhaustif se retrouve, en sortie, avec un très grand nombre de motifs redondants ce qui rend sa compréhension des résultats compliquée voire impossible. La même chose est valable pour un algorithme d'apprentissage conçu pour utiliser les motifs extraits comme variables explicatives. En effet, une abondance de motifs ne facilite pas forcément la tâche à l'algorithme, car un grand nombre de variables explicatives entraîne souvent du sur-apprentissage.

Tout d'abord, dans (Boulicaut et al., 2006), les auteurs présentent les différentes approches permettant d'avoir une représentation plus courte des motifs d'itemsets d'un contexte formel. Ils s'intéressent particulièrement aux motifs d'itemsets fréquents et proposent trois types de représentations condensées : les motifs maximaux qui correspondent aux éléments de la frontière dans le treillis de motifs, les motifs fermés qui sont les motifs maximaux dans les classes d'équivalence associées aux opérateurs de fermeture, et les représentations condensées approximatives utilisées pour les grands jeux de données et qui sont calculées sur un sous-échantillon des données d'origine. Par ailleurs, (Knobbe and Ho, 2006) ont proposé le concept de "pattern teams" ou sous-ensemble des top-k motifs. Ils cherchent le sous-ensemble de motifs de cardinalité fixe k et qui optimise une mesure de qualité Φ définie sur les ensembles de motifs et qui permet par exemple de garantir un niveau de diversité dans les motifs retournés. D'une façon similaire, (Bjorn et al., 2010) ont proposé une procédure d'extraction de sous-ensembles de motifs en fonction de l'intérêt de l'utilisateur de l'outil. Leur idée est que le meilleur sous-ensemble de motifs est celui qui permet d'obtenir

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

une partition des objets la plus similaire de celle obtenue avec l'ensemble complet de tous les motifs. Enfin, la méthode Krimp proposée dans (Vreeken et al., 2011) utilise le concept du MDL (Minimum Description Length) pour la sélection d'un sous-ensemble de motifs. Krimp cherche le plus petit sous-ensemble de motifs qui permet d'obtenir la représentation la plus succincte du jeu de données. La méthode est inspirée de la théorie de l'information et utilise des tables d'encodage sur les motifs. Le sous-ensemble de motifs retourné par Krimp est tel que la somme de la taille de la table d'encodage correspondante et de la taille de la représentation du jeu de données obtenue avec cette table soit minimale.

4.3 Communautés dans un multigraphe de co-appartenance

L'objectif de ce chapitre est la détection de communautés de nœuds ayant une certaine homogénéité structurelle et sémantique dans un multigraphe de co-appartenance $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L}, \mathbf{k})$. Dans cette partie, nous reformulons cet objectif en un problème d'extraction de motifs fermés, ou concepts, depuis le contexte d'itemsets $C_{\mathcal{G}} = (\mathcal{L}, (\mathcal{V}, \mathcal{K}), \mathcal{D}_{\mathcal{L}})$ dérivé du multigraphe de co-appartenance $\mathcal{G}$. $\mathcal{D}_{\mathcal{L}}$ est un jeu de données binaires qui contient les associations entre les couches et les nœuds et entre les couches et le vocabulaire. En reprenant le multigraphe de co-appartenance de l'exemple 3.1 du chapitre 3, nous pouvons en déduire son contexte formel $C = (\mathcal{L}, (\mathcal{V}, \mathcal{K}), \mathcal{D}_{\mathcal{L}})$ où $\mathcal{D}_{\mathcal{L}}$ est le jeu de données binaires suivant :

Couches	a	b	c	d	e	f	'Vert'	'Rouge'	'Bleu'
L_1				x	x	x	x		
L_2	x				x	x		x	
L_3	x	x	x	x					x

TABLE 4.2 – Jeu de données binaires du contexte formel associé au multigraphe de co-appartenance de l'exemple 3.1.

Par la suite, le principe de l'extraction se base sur la recherche des sous-ensembles d'objets du contexte qui ont un maximum d'attributs en commun. En d'autres termes, nous cherchons des sous-ensembles de couches $L \subseteq \mathcal{L}$ tels que les deux proportions de nœuds communs et de mots communs soient grandes.

4.3.1 Le langage des sous-graphes

Le langage de motifs du contexte $C_{\mathcal{G}} = (\mathcal{L}, (\mathcal{V}, \mathcal{K}), \mathcal{D}_{\mathcal{L}})$ contient des motifs fermés de la forme $(L, (V, K))$ où $L \subseteq \mathcal{L}$, $V \subseteq \mathcal{V}$ et $K \subseteq \mathcal{K}$ et vérifient $\forall l \in L$ et $\forall (v, k) \in (V, K)$, $v \in \mathbf{v}(l)$ et $k \in \mathbf{k}(l)$. L'énumération des éléments du langage de motifs équivaut à l'énumération des sous-graphes du multigraphe de co-appartenance $\mathcal{G}$. La définition d'une fonction de qualité pour les motifs permettra de sélectionner les sous-graphes qui correspondent à des communautés de nœuds homogènes dans le multigraphe. De plus, l'ensemble de mots K parmi les mots du vocabulaire $\mathcal{K}$ permet d'obtenir une caractérisation sémantique pour la communauté de nœuds.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

4.3.2 Des communautés partageant des nœuds et des attributs textuels

Nous proposons ainsi d'évaluer la qualité d'un motif $(L, (V, K))$ en mesurant la qualité de l'ensemble de couches $L \subseteq \mathcal{L}$ avec une fonction de qualité Q de la forme :

$$Q^j(L) = \alpha\, Q^j_{\mathrm{v}}(L) + (1-\alpha)\, Q^j_{\mathrm{k}}(L) \qquad j \in \{1,2,3,4\} \tag{4.1}$$

où Q^j_{v} et Q^j_{K} mesurent respectivement l'homogénéité en terme de nœuds et de mots pour les couches L. Ces fonctions sont proportionnelles au nombre de nœuds et de mots que les couches ont en commun. Parmi les dénominateurs de normalisation possibles, les deux suivants possèdent des propriétés utiles pour l'énumération exhaustive des motifs d'itemsets :

$$Q^1_{\mathrm{v}}(L) = \frac{|\bigcap_{l\in L} \mathbf{v}(l)|}{\max_{l\in L} |\mathbf{v}(l)|}, \quad \text{et} \quad Q^1_{\mathrm{k}}(L) = \frac{|\bigcap_{l\in L} \mathbf{k}(l)|}{\max_{l\in L} |\mathbf{k}(l)|}$$

Q^1 est normalisée par le nombre maximal de nœuds ou de mots que les couches L contiennent.

$$Q^2_{\mathrm{v}}(L) = \frac{|\bigcap_{l\in L} \mathbf{v}(l)|}{|\bigcup_{l\in L} \mathbf{v}(l)|}, \quad \text{et} \quad Q^2_{\mathrm{k}}(L) = \frac{|\bigcap_{l\in L} \mathbf{k}(l)|}{|\bigcup_{l\in L} \mathbf{k}(l)|}$$

Ici, nous normalisons par la taille de l'union des ensembles de nœuds et de mots.

4.3.2.1 Propriétés des contraintes

La contrainte P^1 issue de Q^1 et définie par $P^1 \equiv Q^1(S) \geqslant \lambda$ est antimonotone : si P^1 est vérifiée par un ensemble L alors P^1 est vérifiée par tout sous-ensemble $L' \subseteq L$.

Proposition 1. $P^1 : Q^1(.) \geqslant \lambda$ est une contrainte antimonotone : $\forall X \subseteq Y$, si $Q^1(Y) \geqslant \lambda$ alors $Q^1(X) \geqslant \lambda$.

Démonstration. Soient X et Y deux sous-ensembles d'objets du contexte $C_{\mathcal{G}}$ tels que $X \subseteq Y$. Nous avons directement $\bigcap_{l\in Y}\mathbf{v}(l) \subseteq \bigcap_{l\in X}\mathbf{v}(l)$ par propriété de l'intersection et $\max_{l\in X} |\mathbf{v}(l)|$ $\leqslant \max_{l\in Y} |\mathbf{v}(l)|$ par propriété du maximum. Nous pouvons en déduire l'inégalité $\frac{|\bigcap_{l\in Y}\mathbf{v}(l)|}{\max_{l\in Y} |\mathbf{v}(l)|} \leqslant \frac{|\bigcap_{l\in X}\mathbf{v}(l)|}{\max_{l\in X} |\mathbf{v}(l)|}$. Suivant le même raisonnement nous pouvons déduire que $\frac{|\bigcap_{l\in Y}\mathbf{k}(l)|}{\max_{l\in Y} |\mathbf{k}(l)|} \leqslant \frac{|\bigcap_{l\in X}\mathbf{k}(l)|}{\max_{l\in X} |\mathbf{k}(l)|}$. En multipliant par les coefficients positifs α et $(1-\alpha)$ et en sommant les deux inégalités nous trouvons $Q^1(X) \geqslant Q^1(Y)$. □

Proposition 2. $P^2 : Q^2(.) \geqslant \lambda$ est une contrainte antimonotone : $\forall X \subseteq Y$, si $Q^2(Y) \geqslant \lambda$ alors $Q^2(X) \geqslant \lambda$.

Démonstration. En notant que $\forall X \subseteq Y$, nous avons $|\bigcup_{l\in X} \mathbf{v}(l)| \leqslant |\bigcup_{l\in Y} \mathbf{v}(l)|$ et en reprenant les étapes de la dernière démonstration, nous trouvons le résultat $Q^2(X) \geqslant Q^2(Y)$. □

On choisit pour la suite la contrainte de qualité P^1 associée à la mesure de qualité Q^1 pour l'extraction des motifs d'itemsets. Nous noterons la mesure de qualité Q et la contrainte correspondante P.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

4.3.2.2 Fermeture

Définition 19. *Soient les applications* $\phi : \mathcal{P}(\mathcal{L}) \rightarrow \mathcal{P}(\mathcal{V}) \times \mathcal{P}(\mathcal{K})$ *et* $\psi : \mathcal{P}(\mathcal{V}) \times \mathcal{P}(\mathcal{K}) \rightarrow \mathcal{P}(\mathcal{L})$ *telles que :*

$$\forall L \subseteq \mathcal{L}, \quad \phi(L) = \ \bigcap_{l \in L} \mathbf{v}(l), \bigcap_{l \in L} \mathbf{k}(l)\big\} \tag{4.2}$$

$$\forall (V,K) \subseteq \mathcal{V} \times \mathcal{K}, \quad \psi((V,K)) = \ l \in \mathcal{L} | \forall v \in V, v \in \mathbf{v}(l) \wedge \forall k \in K, k \in \mathbf{k}(l)\big\} \tag{4.3}$$

On peut alors définir les **opérateurs de fermeture** σ et σ' pour les motifs du contexte $C_{\mathcal{G}}$:

$$\sigma = \psi \circ \phi \qquad \sigma' = \phi \circ \psi$$

Démonstration. (1) L'opérateur σ vérifie les 3 conditions d'un opérateur de fermeture (voir Définition 13) :

extensité : $\forall L \subseteq \mathcal{L}, \ L \subseteq \sigma(L)$:
Notons $\phi(L) = (V_s(L), K_s(L))$ avec $V_s(L) = \bigcap_{l \in L} \mathbf{v}(l)$ et $K_s(L) = \bigcap_{l \in L} \mathbf{k}(l)$.
Nous pouvons distinguer deux cas pour l'ensemble $L' = \mathcal{L} \backslash L$:

- 1er cas : $\nexists l' \in L'$ tel que $(\forall v \in V_s(L), v \in \mathbf{v}(l')) \wedge (\forall k \in K_s(L), k \in \mathbf{k}(l'))$.
  Dans ce cas, $\sigma(L) = \psi((V_s(L), K_s(L)))$ ne contient que les éléments de L (par éq. 4.3). Et, $L = \sigma(L)$
- 2ème cas : $\exists l' \in L'$ tel que $(\forall v \in V_s(L), v \in \mathbf{v}(l')) \wedge (\forall k \in K_s(L), k \in \mathbf{k}(l'))$.
  Ici, nous avons $l' \notin L$ et $l' \in \psi(V_s(L), K_s(L)) = \sigma(L)$ d'où $L \subset \sigma(L)$

A partir des deux possibilités, nous pouvons conclure la propriété d'extensité pour l'opérateur σ.

monotonicité : $\forall (L, L') \subseteq \mathcal{L}, \quad L' \subseteq L \Rightarrow \sigma(L') \subseteq \sigma(L)$:
Soient $L, L' \subseteq \mathcal{L}$ tels que $L' \subseteq L$, nous en déduisons $V_s(L) \subseteq V_s(L')$ et $K_s(L) \subseteq K_s(L')$ par propriété de l'intersection. D'où $\phi(L) \subseteq \phi(L')$.
Par ailleurs, supposons $V, V' \subseteq \mathcal{V}$ et $K, K' \subseteq \mathcal{K}$ tels que : $V' \subseteq V$ et $K' \subseteq K$, alors par application de la définition de ψ (équation 4.3), nous avons $\psi((V,K)) \subseteq \psi((V',K'))$. En remplaçant dans cette dernière relation (V', K') par $\phi(L)$ et (V, K) par $\phi(L')$, nous avons bien $\phi(L) \subseteq \phi(L')$ d'où $\psi(\phi(L')) \subseteq \psi(\phi(L))$ c'est-à-dire $\sigma(L') \subseteq \sigma(L)$.

idempotence : $\forall L \in \mathcal{L}, \ \sigma(\sigma(L)) = \sigma(L)$:
Nous avons la propriété d'extension pour l'opérateur $\sigma : L \subseteq \sigma(L)$. Nous pouvons alors écrire $\sigma(L) = L \bigcup L'$ avec $\forall l' \in L', \forall v \in V_s(L), v \in \mathbf{v}(l')$ et $\forall k \in K_s(L), k \in \mathbf{k}(l')$. En d'autres termes, $V_s(L) \subseteq V_s(L')$ et $K_s(L) \subseteq K_s(L')$. Nous avons alors :

$$\begin{aligned}
\phi(\sigma(L)) =& \phi(L \bigcup L') \\
=& (V_s(L \bigcup L'), K_s(L \bigcup L')) \\
=& (V_s(L) \bigcap V_s(L'), K_s(L) \bigcap K_s(L')) \\
=& (V_s(L), K_s(L)) \\
=& \phi(L)
\end{aligned}$$

et en introduisant ψ des deux cotés de l'égalité, nous trouvons :

$$\sigma(\sigma(L)) = \sigma(L)$$

(2) L'opérateur σ' vérifie les 3 conditions d'un opérateur de fermeture :

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

extensité : $\forall (V,K) \subseteq \mathcal{V} \times \mathcal{K}, \quad (V,K) \subseteq \sigma'((V,K))$:
Nous distinguons deux possibilités pour l'ensemble $(V',K') = ((\mathcal{V}\backslash V) \cup \{\varnothing\}, (\mathcal{K}\backslash K) \cup \{\varnothing\})$:

— 1er cas : $\nexists (v',k') \in (V',K')$ tel que $(v',k') \neq (\varnothing,\varnothing)$ et $\forall l \in \psi((V,K))$, nous avons $v' \in \mathbf{v}(l)$ et $k' \in \mathbf{k}(l)$. Dans ce cas, $(V,K) = \sigma((V,K))$.

— 2ème cas : $\exists (v',k') \in (V',K')$ tel que $(v',k') \neq (\varnothing,\varnothing)$ et $\forall l \in \psi((V,K))$, nous avons $v' \in \mathbf{v}(l)$ et $k' \in \mathbf{k}(l)$. Ici, il y a au moins un élément (v',k') en plus dans $\sigma'((V,K))$ qui n'est pas dans (V,K). D'où $(V,K) \subset \sigma'((V,K))$.

Il en résulte $(V,k) \subseteq \sigma'((V,K))$

monotonicité : $\forall (V,K) \subseteq \mathcal{V} \times \mathcal{K}, \quad (V',K') \subseteq (V,K) \Rightarrow \sigma'((V',K')) \subseteq \sigma'((V,K))$:
Par la définition de l'opérateur ψ (équation 4.3), nous pouvons déduire que $(V',K') \subseteq (V,K) \Rightarrow \psi((V,K)) \subseteq \psi((V',K'))$. Aussi, par définition de l'opérateur ϕ et en utilisant la propriété de l'intersection, nous avons $L' \subseteq L \Rightarrow \phi(L) \subseteq \phi(L')$. En remplaçant dans cette dernière équation L' par $\psi((V,K))$ et L par $\psi((V',K'))$, nous avons bien $\psi((V,K)) \subseteq \psi((V',K'))$, d'où $\phi(\psi((V',K'))) \subseteq \phi(\psi((V,K)))$ et $\sigma'((V',K')) \subseteq \sigma'((V,K))$.

idempotence : $\forall (V,K) \subseteq \mathcal{V} \times \mathcal{K}, \quad \sigma'(\sigma'((V,K)) = \sigma'((V,K))$:
Nous avons $(V,K) \subseteq \sigma'((V,K))$ (extensité de σ'). Nous pouvons alors écrire $\exists (V',K') \in ((\mathcal{V}\backslash V) \cup \{\varnothing\}, (\mathcal{K}\backslash K) \cup \{\varnothing\})$ tel que $\sigma'((V,K)) = (V \cup V', K \cup K')$ et où $\forall l \in \psi((V,K))$, nous avons $\forall v \in V', v \in \mathbf{v}(l)$ et $\forall k \in K', k \in \mathbf{k}(l)$.
Nous avons alors, $\psi(\sigma'((V,K))) = \psi((V \cup V', K \cup K')) = \psi((V,K)) \cap \psi((V',K'))$.
Or, nous avons aussi $\psi((V,K)) \subseteq \psi(V',K'))$.
D'où $\psi(\sigma'((V,K))) = \psi((V,K))$,
et $\sigma'(\sigma'(V,K)) = \sigma'((V,K))$.

□

Fonctions de qualité et concepts

Il est à noter que deux motifs partageant le même ensemble d'attributs ne recouvrent pas forcément le même ensemble d'objets. Parmi eux, c'est celui qui a le plus grand nombre d'objets qui sera préféré. Il est obtenu en utilisant les opérateurs de fermeture σ et σ'. Ce motif fermé $(L,(V,K))$ est alors appelé concept et il vérifie : $\sigma(L) = L$ et $\sigma'((V,K)) = (V,K)$.

Proposition 3. $Q(L) \geqslant Q(\sigma(L))$

Démonstration. En utilisant les définitions précédentes, nous pouvons écrire :

$$\begin{aligned} Q(\sigma(L)) =& \alpha \frac{|\bigcap_{l\in\sigma(L)} \mathbf{v}(l)|}{\max_{l\in\sigma(L)} |\mathbf{v}(l)|} + (1-\alpha)\frac{|\bigcap_{l\in\sigma(L)} \mathbf{k}(l)|}{\max_{l\in\sigma(L)} |\mathbf{k}(l)|} \\ =& \alpha \frac{|\bigcap_{l\in L} \mathbf{v}(l)|}{\max_{l\in\sigma(L)} |\mathbf{v}(l)|} + (1-\alpha)\frac{|\bigcap_{l\in L} \mathbf{k}(l)|}{\max_{l\in\sigma(L)} |\mathbf{k}(l)|} \end{aligned}$$

Puisque σ est un opérateur de fermeture alors il est extensif. D'où

$$\max_{l\in\sigma(L)} |\mathbf{v}(l)| \geqslant \max_{l\in L} |\mathbf{v}(l)|, \text{ and } \max_{l\in\sigma(L)} |\mathbf{k}(l)| \geqslant \max_{l\in L} |\mathbf{k}(l)|$$

Ce qui donne $Q(\sigma(L)) \leqslant Q(L)$. □

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

4.3.3 Différents algorithmes de calcul de communautés

Soit le multigraphe de co-appartenance $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{L}, \mathbf{k})$ où l'application $\mathbf{k} : \mathcal{L} \rightarrow \mathcal{P}(\mathcal{K})$ retourne les attributs textuels associés aux couches du graphe et $\mathcal{K}$ est le vocabulaire des mots utilisés dans ces attributs. Dans ce graphe, les nœuds d'une même couche forment une clique dans laquelle les relations entre nœuds sont transitives. Nous pouvons ainsi définir une application $\mathbf{v} : \mathcal{L} \rightarrow \mathcal{P}(\mathcal{V})$ qui retourne pour chaque couche les nœuds qui y appartiennent. Dans $\mathcal{G}$, il y a alors une correspondance entre l'ensemble des couches d'un coté et l'ensemble des nœuds et du vocabulaire des attributs textuels de l'autre. En notant $\mathcal{D}_{\mathcal{L}}$ le dataset qui contient ces relations de correspondance $\mathbf{v}$ et $\mathbf{k}$, nous définissons le contexte formel $C_{\mathcal{G}} = (\mathcal{L}, (\mathcal{V}, \mathcal{K}), \mathcal{D}_{\mathcal{L}})$ dans lequel l'ensemble des objets est l'ensemble des couches $\mathcal{L}$, et l'ensemble des attributs est la composition des deux ensembles distincts des nœuds $\mathcal{V}$ et du vocabulaire $\mathcal{K}$. Les motifs de ce contexte sont de la forme $(L, (V, K))$ où $L \subseteq \mathcal{L}, V \subseteq \mathcal{V}$ et $K \subseteq \mathcal{K}$. La recherche des motifs peut se faire dans l'espace des objets $\mathcal{P}(\mathcal{L})$ ou dans celui des attributs $\mathcal{P}(\mathcal{V}) \times \mathcal{P}(\mathcal{K})$.

4.3.3.1 Par énumération exhaustive

La mesure de qualité Q est utilisée pour identifier l'ensemble C des communautés de nœuds du multigraphe de co-appartenance $\mathcal{G}$. A cette fin, nous cherchons l'ensemble de motifs $\mathcal{M} = \{M_i = (L_i, (V_i, K_i)), i \in [1..N]\}$ tel que la mesure de qualité des sous-ensembles de couches $Q(L_i)$ soit supérieure à un seuil λ défini par l'analyste. Nous proposons de résoudre ce problème de deux façons. D'abord, nous présentons un algorithme exhaustif qui retourne l'ensemble des motifs fermés du contexte d'itemsets $C_{\mathcal{G}}$. Ensuite, nous introduisons une variante de cet algorithme qui permet de sélectionner un sous-ensemble de k motifs tel que la somme des qualités des motifs et leur diversité soient maximales.

Ensemble complet de motifs

On énumère les concepts du contexte formel $C_{\mathcal{G}}$ en utilisant une stratégie de recherche en profondeur. Étant donné l'ensemble de couches $L \subseteq \mathcal{L}$ exploré actuellement et l'indice i de la prochaine couche à visiter, l'algorithme Exhaustif retourne tous les sur-ensembles de L de mesure de qualité supérieure à λ et qui sont fermés : $\{X \mid S \subset X \subseteq \mathcal{L}, Q(X) \geqslant \lambda \text{ et } \sigma(X) = X\}$.

A l'initialisation, L est l'ensemble vide et $i = 1$. Exhaustif tire profit de l'antimonotonicité de Q. Cette propriété est utilisée à la ligne 4 de l'algorithme 3 pour arrêter la recherche quand plus aucun concept ne peut être obtenu dans la branche de recherche actuelle. Afin d'éviter de générer un même concept à de multiples reprises, nous utilisons un ordre arbitraire $\ll$ pour l'ensemble des couches $\mathcal{L}$ tel que $l_i \ll l_j$ ssi $i \leqslant j$. Nous généralisons cet ordre pour les ensembles de couches : $\forall X, Y \subseteq \mathcal{L}$, $X \ll Y \Leftrightarrow \forall x \in X$ et $\forall y \in Y \backslash X$, $x \ll y$. Nous vérifions alors à chaque exécution de l'algorithme que l'opération de fermeture ne rajoute que des couches pas encore énumérées (ligne 6). Même si n'importe quel ordre peut être utilisé pour éviter la redondance des concepts, nous choisissons d'ordonner l'ensemble des couches $\mathcal{L}$ en fonction du nombre de nœuds et de mots clés qu'elles contiennent (deux couches $x, y \in \mathcal{L}$ sont telles que $x \ll y$ ssi $|\mathbf{v}(x) \cup \mathbf{k}(x)| \leqslant |\mathbf{v}(y) \cup \mathbf{k}(y)|$). La fonction nextIndex retourne l'indice de la première couche plus grande que les couches dans L selon l'ordre $\ll$, et telle que la couche ne soit pas dans L'.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Algorithm 3: EXHAUSTIF

Input: L l'ensemble des couches qui appartiennent à la communauté en cours de construction, i l'index de la dernière couche à ajouter à la communauté, λ le seuil pour la mesure de qualité Q, R l'ensemble des communautés calculées.

Output: $R = \{X \mid L \subseteq X,\ \sigma(X) = X \text{ and } Q(X) \geqslant \lambda\}$

```
if (i = |𝓛| + 1) and (Q(L) ⩾ λ) then
    R ← R ∪ {L}
else
    if Q(L ∪ l_i) ⩾ λ then
        L' ← σ(L ∪ l_i)
        if L ≪ L' then
            j ← nextIndex(L, L')
            Exhaustif(L', j, λ, R)
        Exhaustif(L, i + 1, λ, R)
```

A la fin de l'exécution, EXHAUSTIF retourne un ensemble de motifs fermés qui dépend du choix de paramètres λ et α et qu'on note $\mathcal{M}_{\text{EXHAUSTIF}}(\mathcal{G}, \lambda, \alpha) = \{(L_i, (V_i, K_i))\}$. Avec de petites valeurs pour λ, $\mathcal{M}_{\text{EXHAUSTIF}}$ contient de nombreux concepts qui sont redondants entre eux. A l'inverse, une large valeur pour λ restreint l'énumération à ne retourner qu'un petit nombre de concepts qui ont une mesure de qualité proche de 1 et dont les couches sont très similaires en termes de nœuds et de mots. Une possibilité pour résoudre ce problème est de chercher une valeur optimale λ^* pour λ tel que $\mathcal{M}_{\text{EXHAUSTIF}}(\mathcal{G}, \lambda^*, \alpha)$ soit le plus informatif avec le nombre de nœuds recouverts par ses concepts qui est maximal et la similarité entre eux qui est minimale. En d'autres termes, la valeur optimale λ^* est telle que les concepts générés par EXHAUSTIF recouvrent un maximum de nœuds avec un minimum de redondance. Le recouvrement des nœuds par les concepts, noté $co(\lambda)$, est calculé comme le ratio du nombre de nœuds apparaissant dans les concepts $\mathcal{M}_{\text{EXHAUSTIF}}(\mathcal{G}, \lambda^*, \alpha)$ sur le nombre total de nœuds dans le multigraphe de co-appartenance $\mathcal{G}$. Et, la redondance est estimée à l'aide de la distribution des similarités de Jaccard entre concepts. Le troisième quartile de cette distribution, $q_3(\lambda)$, est la valeur de redondance assignée à l'ensemble de concepts. La formule pour calculer le score $sc(\lambda)$ permet d'équilibrer le recouvrement et la redondance : $sc(\lambda) = \log \frac{co(\lambda)}{q_3(\lambda)}$. La valeur optimale λ^* est celle dont le score maximal.

Sous-ensemble des top-k motifs diversifiés

Une alternative à l'optimisation du seuil de qualité λ est de chercher, parmi l'ensemble des concepts du contexte $C_{\mathcal{G}}$, un sous-ensemble de cardinal k qui maximise la somme des mesures de qualité des concepts tout en étant diversifié. Cette approche, appelée TOP-K, a deux paramètres : k le nombre de concepts, et δ le seuil de similarité tel que toutes les paires de concepts de l'ensemble ont une mesure de similarité inférieure à δ. L'ensemble de concepts retourné par cette approche est appelé l'ensemble des top-k motifs diversifiés et est noté $\mathcal{M}_{\text{TOP-K}}(\mathcal{G}, k, \delta)$. L'implémentation de cette méthode (voir Algorithme 4) nécessite de modifier l'algorithme EXHAUSTIF pour limiter la taille de l'ensemble des concepts à k, et ne garder que les concepts avec une qualité maximale et une redondance minimale. La fonction

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Top-K (Algorithme 5) est celle qui permet d'obtenir l'ensemble des top-k motifs diversifiés. Elle traverse la liste ordonnée des k concepts trouvés auparavant et enregistre ceux qui sont similaires au concept courant *Closed* dans la variable *similarPatterns*. Si un de ces concepts similaires a une mesure de qualité Q supérieure à celle du concept *Closed*, la boucle est arrêtée car le concept *Closed* ne sera pas inséré dans la liste P des top-k motifs diversifiés. Si la variable *similarPatterns* ne contient aucun concept alors deux cas sont possibles :

(1) la liste P est pleine : elle contient k concepts, dans ce cas, *Closed* remplacera le concept de plus petite qualité et le seuil minimal de la qualité min_Q sera mis à jour,

(2) la liste P contient moins de k concepts, alors le concept *Closed* sera simplement rajouté à P.

Enfin, si *Closed* a des concepts similaires dans *similarPatterns* et que sa mesure de qualité est supérieure à toutes celles des concepts similaires alors l'ensemble de ces concepts sont retirés de P et remplacés par *Closed*. Le seuil de qualité $\min_q$ n'est mis à jour que si la liste des top-k motifs diversifiés P est pleine c'est-à-dire si elle contient k concepts.

Algorithm 4: Top-K

Input: S, i, R
Output: $R = \{X \mid S \subseteq X,\ \sigma(S) = S \text{ and } Q(S) \geqslant \lambda\}$

```
1 if (i = |ℒ| + 1) and (Q(S) ⩾ λ) then
2 |   TopKDiv(R, S, k, δ, min_Q)
3 else
4 |   if Q(S ∪ L_i) ⩾ λ then
5 |   |   S' ← σ(S ∪ L_i)
6 |   |   if S ≪ S' then
7 |   |   |   j ← nextIndex(S, S')
8 |   |   |   Exhaustif(S', j, R)
9 |   |   Exhaustif(S, i + 1, R)
```

4.3.3.2 Par échantillonnage

En général, l'extraction de tous les concepts d'un contexte issu d'un graphe de co-appartenance nécessite la capacité de traiter de très grands volumes de données. En effet, la restriction de l'extraction à une région particulière du graphe n'est pas toujours souhaitée par l'analyste et peut elle-même engendrer un sous-graphe de grande taille. Dans ces cas, la recherche exhaustive de tous les concepts est coûteuse en terme de calcul car le nombre de concepts croît exponentiellement avec le nombre de couches, de nœuds et de mots clés. Il est alors possible d'utiliser des méthodes d'échantillonnage pour trouver plus rapidement un sous-ensemble de motifs dont la mesure de qualité est grande. Ces méthodes offrent plusieurs avantages :

(1) la réduction du coût de calcul

(2) l'identification de motifs intéressants par rapport à la mesure de qualité

(3) la possibilité de distribuer les calculs sur plusieurs machines.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Algorithm 5: TopKDiv(P, *Closed*, k, δ, $\min_Q$)

Input: P l'ensemble d'au plus k motifs diversifiés ordonnés selon Q.
Output: P la liste mise à jour ainsi que le seuil $\min_Q$.

```
similarPatterns ← ∅
i ← 0
isBetter ← true
while i < |P| and isBetter do
    J ← P[i]
    if Jaccard(J, Closed) ⩾ δ then
        insert(J, similarPatterns)
        if Q(J) > Q(Closed) then
            isBetter ← false
    i ← i + 1
if (|similarPatterns| = 0) then
    if (Q(Closed) ⩾ δ) then
        if (|P| = k) then
            remove(P[0]), P)
            insert(Closed, P)
            min_Q ← Q(P[0])
        else
            insert(Closed, P)
else
    if isBetter then
        forall J ∈ similarPatterns do
            remove(J, P)
        insert(Closed, P)
        if (|P| = k) then
            min_Q ← Q(P[0])
```

Nous distinguons deux familles de méthodes d'échantillonnage. La première est basée sur les méthodes MCMC qui réalisent des marches aléatoires dans un graphe d'états dans lequel les états correspondent aux concepts et les liens entre états les probabilités d'atteindre le prochain concept à partir du concept courant. Ceci peut être fait avec la garantie que les concepts seront tirés aléatoirement selon une distribution de probabilité proportionnelle à leur mesure de qualité. Cependant, le coût de calcul pour les méthodes de cette famille est grand car le graphe d'états doit être matérialisé dans les deux directions : celle de l'ajout d'un objet au concept et celle de l'ajout d'un attribut. Il existe d'autres approches qui relâchent cette contrainte. Dans la suite, deux approches heuristiques basées sur l'échantillonnage de motifs et concepts maximaux sont adaptées pour extraire des motifs dans un contexte $C_{\mathcal{G}}$. Enfin, une approche par échantillonnage direct en deux étapes est également présentée.

Méthode de Monte-Carlo par chaînes de Markov – MCMC

Dans (Boley et al., 2010), les auteurs utilisent une méthode MCMC pour échantillonner des motifs fermés selon une distribution de probabilité proportionnelle à une mesure de qualité. Ils définissent une chaîne de Markov sur l'espace d'états des motifs fermés. Les

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

liens entre deux états – correspondant à deux concepts $C = \big(L, (V,K)\big)$ et $C' = \big(L', (V',K')\big)$ – existent s'il y a au moins en élément générateur x tel qu'il est possible de passer d'un état à l'autre avec les relations $\sigma\Big(\big(L \cup x, (V,K)\big)\Big) = C'$ or $\sigma'\big(\big(L, (V,K) \cup x\big)\big) = C'$. Pour appliquer la méthode MCMC et s'assurer de la convergence de la chaîne de Markov vers sa distribution stationnaire, celle-ci doit être ergodique, c'est-à-dire que tous ses états doivent être atteignables depuis n'importe quel autre état. Cette propriété est obtenue à l'aide d'un mélange aléatoire de deux chaînes de Markov définies sur le même espace d'états, et dont les probabilités de transition sont proportionnelles au nombre d'éléments générateurs. La première chaîne va d'un concept C à son successeur C' par l'ajout d'un objet (une couche du graphe pour $C_{\mathcal{G}}$) et elle a pour probabilité de transition entre états :

$$P_s(C,C') \propto |\{L \in \mathcal{L} \mid \phi(S \cup L) = (V',K')\}|$$

Inversement, la deuxième chaîne va d'un concept C à son prédécesseur C' par l'ajout d'un attribut (nœud ou mot pour $C_{\mathcal{G}}$). Sa probabilité de transition entre états est telle que :

$$P_s(C,C') \propto |\{v \in \mathcal{V}_{\mathcal{L}} \mid \psi(V \cup v, K) = S'\}| + |\{k \in \mathcal{K}_{\mathcal{L}} \mid \psi(V, K \cup k) = S'\}|$$

Cependant, ce mélange de chaînes de Markov n'est pas symétrique. Nous l'utilisons alors dans le cas général de l'algorithme de Metropolis-Hastings en rajoutant une probabilité de rejet du prochain état : le successeur C' de C associé à la probabilité P_S est maintenu avec une probabilité

$$\min\left(\frac{P_s(C',C)}{P_s(C,C')} \times \frac{Q(S')}{Q(S)}, 1\right)$$

La chaîne de Markov résultante a une probabilité de transition entre états de la forme :

$$P(C,C') = P_s(C,C') \min\left(\frac{P_s(C',C)}{P_s(C,C')} \times \frac{Q(S')}{Q(S)}, 1\right)$$

Cet algorithme, dénommé ConceptSampling dans la suite, ne passe pas à l'échelle car le nombre d'itérations de la chaîne de Markov doit être significativement supérieur à la taille de l'espace d'états pour garantir la convergence de la chaîne vers sa distribution stationnaire. De plus, pour notre cas, chaque itération nécessite beaucoup de calculs car elle requiert d'appliquer l'opérateur de fermeture à de multiples reprises pour calculer toutes les probabilités de transition.

Échantillonnage biaisé pour les motifs maximaux

Cette méthode, présentée dans (Chaoji et al., 2008), échantillonne des motifs maximaux mais sans garantie de l'uniformité de la probabilité d'échantillonnage des motifs. En retour, elle est très efficace même dans le cas où le contexte est de grande taille.

L'approche, détaillée dans l'algorithme 6, commence par un unique objet du contexte puis rajoute itérativement de nouveaux objets tirés aléatoirement selon une distribution de probabilité p proportionnelle à la mesure de qualité Q évaluée sur le concept obtenu après l'ajout de l'objet : $p(L) = \frac{Q(L)}{Z}$ où Z est une constante de normalisation. Le processus s'arrête quand plus aucun objet du contexte ne peut être ajouté au motif courant sans que la mesure de qualité du motif engendré ne soit inférieure au seuil λ.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Algorithm 6: MaximalSampling

Input: $\mathcal{G}_{\mathcal{L}} = (\mathcal{V}_{\mathcal{L}}, \mathcal{E}_{\mathcal{L}}, \mathcal{K}_{\mathcal{L}}, \mathcal{V}, \mathcal{L})$
Output: Une communauté locale S.

1 **draw** a $\sim u([0, |\mathcal{L}|])$
2 $S \leftarrow \{L_a\}$
3 $C \leftarrow \{x \in \mathcal{L} \backslash S\}$
4 **while** $|C| > 0$ **do**
5 $i \leftarrow 0$
6 **for** $x \in C$ **do**
7 $p[i] \leftarrow Q(S \cup x)$
8 $i \leftarrow i + 1$
9 **draw** $x^\star$ from C proportionally to p
10 $S \leftarrow \{S \cup x^\star\}$
11 $C \leftarrow \{x \in C \backslash S \mid Q(S \cup x) \geqslant \lambda\}$

A noter que les motifs maximaux ne sont pas nécessairement fermés à cause de la propriété 3 et du seuil sur la qualité Q (ligne 11). Pour pouvoir échantillonner des concepts (motifs fermés), nous modifions l'algorithme MaximalSampling en ajoutant l'opérateur de fermeture σ dans les lignes 7, 10 et 11 de l'algorithme 6. Cette version est appelée Closed-MaximalSampling.

Échantillonnage direct

Afin d'échantillonner des concepts avec une probabilité proportionnelle à une mesure de qualité, (Boley et al., 2011) proposent une procédure en deux étapes permettant de tirer aléatoirement des concepts sans passer par une étape de simulation de processus stochastiques. Comme expliqué par le pseudo-code de l'algorithme 7, la procédure tire aléatoirement un objet du contexte selon une première distribution de probabilité w_1. Ensuite, dans un deuxième temps, un sous-ensemble des attributs de cet objet est sélectionné aléatoirement avec une probabilité w_2. Les distributions w_1 et w_2 dépendent de la mesure de qualité utilisée. La combinaison des deux étapes suit la distribution désirée (proportionnelle à Q). Cet algorithme est appelé DirectSampling dans la suite. Pour notre cas, nous cherchons à favoriser et à échantillonner plus fréquemment les concepts dont les objets ont des attributs similaires : les concepts dont les couches ont une grande similarité en termes de nœuds et de mots équilibrée avec le paramètre α. La distribution w_2 dépend dont de α. Pour un objet l tiré aléatoirement à l'étape 1, nous avons :

— pour $\alpha = 0$, DirectSampling échantillonne un sous-ensemble de mots avec $w_2 = u(\mathcal{P}(\mathbf{k}(l)))$

— pour $\alpha = 1$, DirectSampling échantillonne un sous-ensemble de nœuds avec $w_2 = u(\mathcal{P}(\mathbf{v}(l)))$

— pour $\alpha = 0$, DirectSampling échantillonne un sous-ensemble de nœuds et de mots avec $w_2 = u(\mathcal{P}(\mathbf{v}(l) \bigcup \mathbf{k}(l)))$

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Algorithm 7: DIRECTSAMPLING

Input: $\mathcal{G}_{\mathcal{L}} = (\mathcal{V}_{\mathcal{L}}, \mathcal{E}_{\mathcal{L}}, \mathcal{K}_{\mathcal{L}}, \mathcal{V}, \mathcal{L})$
Output: A local community S.
1 Let weights w_1 be defined by $w_1(L) = 2^{|K_L \cup V_L|}$ for all $L \in \mathcal{L}$
2 **draw** $L \sim w_1(\mathcal{L})$
3 **draw** $(V, K) \sim w_2(K_L \cup V_L)$
4 **return** $\psi(V, K)$

Top-k motifs diversifiés pour l'échantillonnage

L'approche de filtrage de l'ensemble de motifs 'top-k motifs diversifiés' est applicable aux méthodes par échantillonnage dans une étape de post-traitement. Elle nécessite de calculer les sommes des mesures de qualité et les similarités entre concepts de tous les sous-ensembles de motifs de taille k.

4.4 Résultats et comparaisons

Les deux multigraphes de co-appartenance Twitter et Wikipedia que nous étudions sont de très grande taille : ils contiennent des millions de nœuds répartis sur des centaines de milliers de couches. Une première étape est alors de délimiter les frontières de sous-graphes d'intérêt $\mathcal{G}_U$ en utilisant l'un des deux algorithmes présentés dans le chapitre 3 initialisé avec un ensemble U de nœuds centraux au domaine d'intérêt. Ensuite, nous appliquons les algorithmes d'extraction de motifs présentés ci-dessus sur les contextes $\mathcal{C}_{\mathcal{G}_U}$ pour obtenir grâce à chacune des méthodes un ensemble de motifs $\mathcal{M} = \{(L_i, (V_i, K_i))\}$ tel que $\forall L_i, Q(L_i) \geqslant \lambda$. Pour cela, nous effectuons les étapes suivantes dans nos expériences :

(0) Construction du sous-graphe d'intérêt $\mathcal{G}_U$ à partir de l'ensemble de nœuds centraux U identifiés par l'analyste ;

(1) Exécution des algorithmes d'énumération de concepts :
   — EXHAUSTIF pour trouver l'ensemble exhaustif des concepts de mesure de qualité supérieure à λ
   — TOP-K pour trouver l'ensemble de concepts de cardinalité k et telle que la somme des qualités de ses concepts est maximale et leur redondance est minimale ;

(2) Exécution des algorithmes d'échantillonnage afin de les comparer à l'énumération EXHAUSTIF ;

(3) Évaluation quantitative des communautés obtenues dans le multigraphe de co-appartenance et caractérisation sémantique des communautés ;

(4) Évaluation qualitative des communautés détectées par nos méthodes puis comparaison entre elles et avec des méthodes de l'état de l'art.

L'étape (0) a été exposée dans le chapitre 3 et les algorithmes d'énumération de l'étape (1) sont présentés en détails dans la section 4.3.3.1. Nous décrivons maintenant les trois étapes suivantes.

Étape (2) : Nous comparons les méthodes d'échantillonnage en terme de temps d'exécution. A cette fin, nous représentons les distributions des temps d'échantillonnage de motifs de

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

chaque méthode. Aussi, pour comparer les résultats de ces méthodes à ceux de l'énumération Exhaustif, l'évolution au cours de l'exécution de l'algorithme d'échantillonnage de la proportion de motifs distincts échantillonnés par rapport au nombre total de motifs est tracée. Soient t_e le temps nécessaire à l'exécution de l'énumération exhaustive et n_e le nombre de motifs obtenus. Les algorithmes d'échantillonnage retournent p motifs d'indice $i \in [1..p]$ avec des temps d'échantillonnage cumulés jusqu'au premier tirage du motif $i \in [1..p]$ notés $t_1..t_p$. Nous comparons les méthodes à l'aide des courbes représentant l'évolution de $\frac{i}{n_e}$ en fonction de $\frac{t_i}{t_e}$ avec $i \in [1..p]$ pour estimer la performance des méthodes d'échantillonnage.
Étape (3) : Étant donné un motif fermé $m = (L_i, (V_i, K_i))$ extrait par une de nos méthodes depuis le contexte $C_{\mathcal{G}}$, ce motif correspond dans le multigraphe de co-appartenance à une communauté définie par le sous-ensemble de nœuds V_i. Par ailleurs, les attributs textuels associés aux couches du multigraphe peuvent être utilisés pour caractériser sémantiquement la communauté de nœuds V_i avec les mots fréquents dans K_i. La caractérisation peut être enrichie avec les attributs textuels associés aux nœuds du multigraphe de co-appartenance quand de tels attributs sont disponibles. Nous définissons alors $\mathcal{K}'$ le vocabulaire des mots qui apparaissent dans les attributs textuels des nœuds et l'application $\mathbf{k}' : \mathcal{V} \rightarrow \mathcal{K}'$ qui associe à un nœud $v \in \mathcal{V}$ un ensemble de mots $K'_i \subseteq \mathcal{K}'$. Cette information sémantique a l'avantage d'être indépendante de celle utilisée pour l'extraction de motifs.

Évaluation d'un ensemble de communautés à l'aide de la sémantique

Étape (4) : Dans les multigraphes de co-appartenance, les nœuds d'une même couche sont tous reliés entre eux et forment une clique. Les critères topologiques utilisés habituellement pour évaluer la qualité d'une communauté dans un graphe sont difficilement applicables dans ce cas. Par conséquent, nous proposons d'estimer la qualité des communautés avec des critères purement sémantiques en utilisant les attributs textuels présents dans le multigraphe. Nous préfèrons nous servir de l'information sémantique associée aux nœuds du multigraphe quand c'est possible en utilisant l'application $\mathbf{k}'$. Ceci nous permet d'évaluer la qualité des communautés avec des données qui n'ont pas été utilisées dans les algorithmes de détection de communautés. Pour les deux exemples de multigraphes de co-appartenance, cette source d'information indépendante est disponible grâce aux biographies des utilisateurs Twitter et aux introductions des pages Wikipedia.

Exceptionnalité sémantique évaluée par le TF-IDF moyen et la divergence de Kullback-Leibler

Les données indépendantes utilisées sont des attributs textuels associés aux nœuds du multigraphe de co-appartenance. Ils permettent d'évaluer la performance des méthodes de détection de communautés en mesurant l'exceptionnalité des attributs des nœuds d'une communauté par rapport aux attributs des nœuds en dehors de la communauté. Pour cela, nous comparons la distribution des mots dans le document composé des attributs des nœuds de la communauté à celle des mots dans le document composé des attributs de tous les nœuds du multigraphe. Pour chaque motif trouvé $m = (L_i, (V_i, K_i))$ et sa communauté de nœuds V_i, nous calculons la moyenne des TF-IDF des mots dans K_i par rapport au corpus composé de deux documents : un document correspondant à la caractérisation sémantique de la

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

communauté (les attributs textuels de ses nœuds), et un deuxième constitué des attributs textuels des nœuds extérieurs à la communauté. De manière similaire, nous calculons la divergence de Kullback-Leibler entre les distributions de probabilité correspondantes.

Plus formellement, étant donnée une communauté de nœuds c détectée par une de nos méthodes, nous construisons D_c le document composé des attributs textuels associés aux nœuds de $c : D_c = \{\bigcup_{v \in c} \mathbf{k}'(v)\}$. Nous construisons aussi son complémentaire $\bar{D}_c$ qui contient les attributs textuels des nœuds de $\mathcal{G}_U$ en dehors de c. Nous calculons la moyenne des TF-IDF des mots de D_c sur le corpus $D_c \cup \bar{D}_c$. Le terme IDF est régularisé pour éviter qu'il ne s'annule dans les cas où un mot existe dans les deux documents : $idf(w)$ où $n_w = 2$ si w appartient aux deux documents et $n_w = 1$ s'il n'apparaît que dans un seul document. Le terme TF est calculé avec les fréquences augmentées pour prévenir le biais envers le document de plus grande taille $\bar{D}_c : tf(w)$ où $f_d(w)$ est la fréquence du mot w dans le document d.

On calcule la divergence de Kullback-Leibler $K(P_c \| P)$ qui mesure la différence entre la distribution de probabilité P_c des mots du document D_c et la distribution de probabilité P des mots apparaissant dans le corpus entier $D_c \bigcup \bar{D}_c$.

$$K(P_c \| P) = \sum_{w \in D_c \cup \bar{D}_c} P_c(w) \log \frac{P_c(w)}{P(w)} .$$

Plus la divergence de Kullbak-Leibler $K(P_c \| P)$ est grande, plus la communauté c est exceptionnelle du point de vue de sa sémantique. Ces mesures d'exceptionnalité sémantique sont utilisées pour comparer nos méthodes Exhaustif, DirectSampling et MaximalSampling aux méthodes de l'état de l'art suivantes :

— Louvain (Blondel et al., 2008), méthode de détection de communautés purement topologique appliquée sur la représentation en une seule couche du multigraphe de co-appartenance ;

— ML-LCD, méthode proposée dans (Interdonato et al., 2017) pour la détection de communautés locales dans les graphes multicouches ;

— LPV (Greene et al., 2012), méthode spécifique au réseau social Twitter et qui utilise la co-appartenance des utilisateurs aux listes pour extraire des communautés thématiques dans le multigraphe de co-appartenance ;

— Infomap (Edler et al., 2017), méthode basée sur des marches aléatoires qui décompose un multigraphe en modules en cherchant la représentation binaire du flux d'informations dans le graphe la plus condensée ;

— Mucha, la méthode proposée par (Mucha et al., 2010) basée sur une généralisation de la détection de communautés par maximisation de la modularité au cas des graphes multicouches.

4.4.1 Étude du réseau social Twitter

La procédure expérimentale commence avec la sélection des sous-graphes d'intérêt $\mathcal{G}_{U_1}$ et $\mathcal{G}_{U_2}$ qui correspondent aux domaines d'intérêt 'scène politique française' et 'intelligence artificielle' respectivement. Ils sont obtenus avec l'algorithme *Double couronnes* initialisé avec les nœuds graines U_1 et U_2. Les détails de cette étape initiale sont donnés dans la partie 3.5.1.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Étape (1) : La figure 4.2 montre le nombre de concepts trouvés et les temps d'exécution de l'algorithme Exhaustif en fonction du seuil de qualité λ et pour trois valeurs de $\alpha \in \{0.5, 0.8, 1\}$. Nous observons que

(1) pour $\alpha = 0.5$, le nombre de motifs extraits est très grand à cause de la multiplicité des combinaisons possibles entre les ensembles de nœuds $\mathcal{V}$ et de mots $\mathcal{K}$

(2) les temps d'exécution sont proportionnels aux nombres de motifs extraits

(3) pour $\alpha \in \{0.8, 1\}$, les nombres de motifs trouvés sont similaires pour les plus petites valeurs de λ. Au-delà d'une certaine valeur (0.72 pour $\mathcal{G}_{U_1}$ et 0.56 pour $\mathcal{G}_{U_2}$), le nombre de motifs décroît plus fortement pour $\alpha = 0.8$

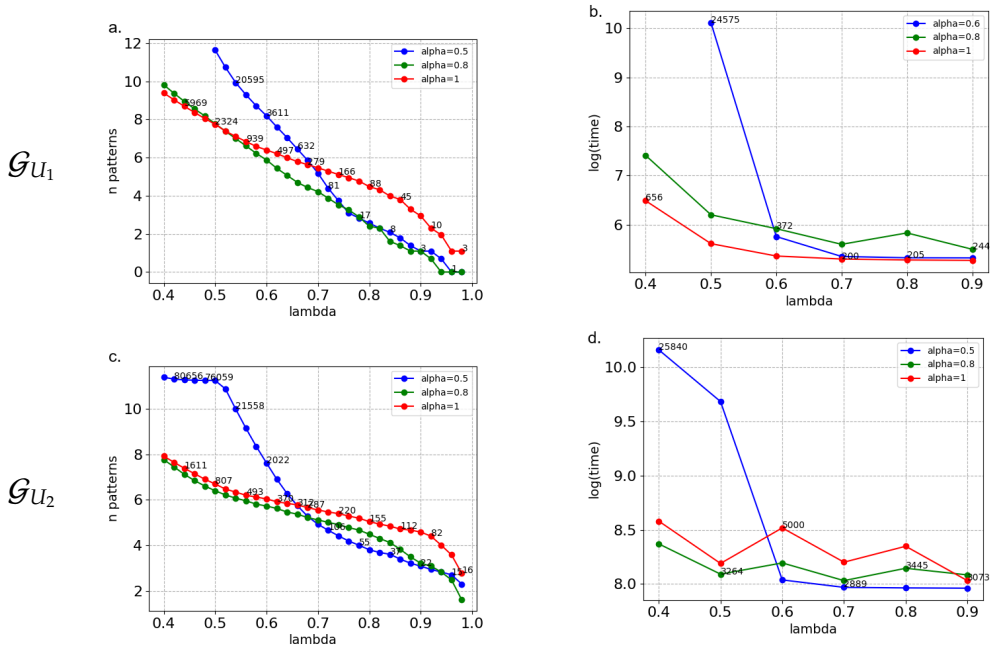


Figure 4.2 – Logarithme du nombre de communautés locales retourné par Exhaustif (gauche), et temps CPU en log(s) (droite) en fonction de α. Première ligne pour $\mathcal{G}_{U_1}$, 'la scène politique française', deuxième ligne $\mathcal{G}_{U_2}$ pour 'l'intelligence artificielle'.

La valeur optimale λ^* pour le seuil de qualité λ est celle qui permet aux communautés de nœuds issues des motifs $\mathcal{M}_{\text{Exhaustif}}$ de recouvrir un maximum de nœuds dans le sous-graphe $\mathcal{G}_U$ tout en limitant la redondance. Pour trouver cette valeur, nous calculons, pour différentes valeurs de λ, le score $sc(\lambda) = \frac{co(\lambda)}{q_3(\lambda)}$. La figure 4.3 montre la mesure de recouvrement co et la mesure de redondance q_3 en fonction de λ. Le nombre de nœuds couverts par les motifs diminue quand λ augmente. La mesure de similarité q_3 entre concepts de $\mathcal{M}_{\text{Exhaustif}}$ décroît aussi quand λ augmente pour les deux exemples et dans le cas où $\alpha \in \{0.5, 0.8\}$. Pour $\alpha = 1$, la similarité augmente avec λ pour le graphe $\mathcal{G}_{U_1}$ et elle a un maximum avec $\lambda = 0.66$ pour le graphe $\mathcal{G}_{U_2}$.

Dans chaque exemple de graphe, la valeur optimale λ^* est, parmi les valeurs évaluées de λ, celle qui maximise $sc(\lambda) : \lambda^* = \arg\max_{\lambda \in I} sc(\lambda)$. Dans la figure 4.4, cette fonction $sc(\lambda)$ est représentée pour $\alpha \in \{0.5, 0.8, 1\}$ et pour les deux exemples de sous-graphes d'intérêt. Pour

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

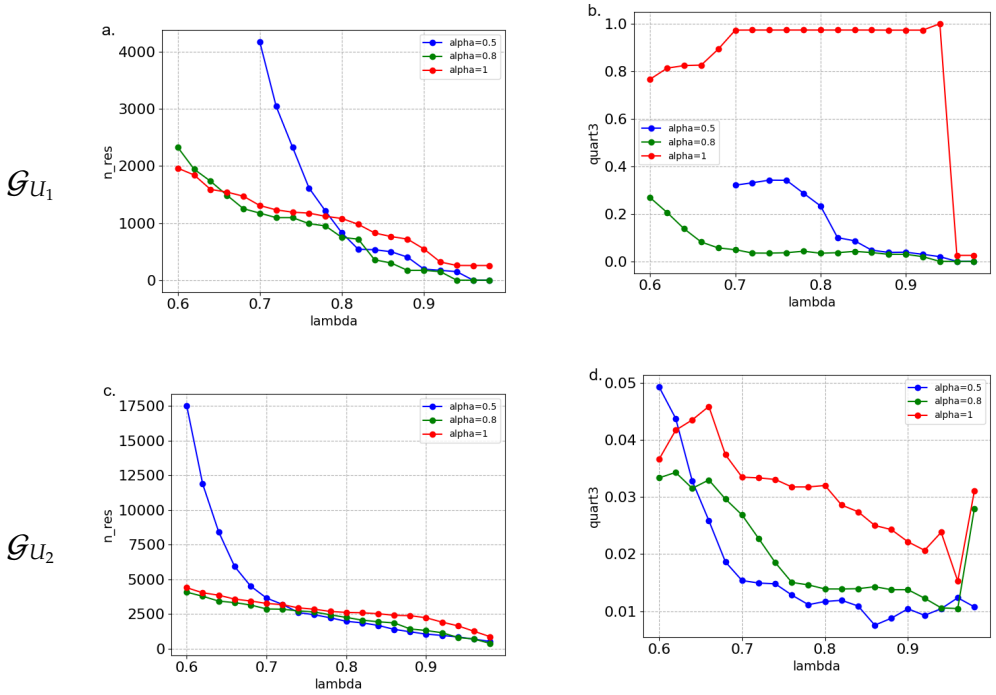


FIGURE 4.3 – $co(\lambda)$ (gauche), et le troisième quartile de la distribution de la mesure de similarité de Jaccard $q_3(\lambda)$ (droite), pour les deux graphes Twitter.

$\alpha = 0.8$ (courbe verte), la courbe de score admet un maximum et nous avons $\lambda^* =$ pour $\mathcal{G}_{U_1}$ et $\lambda^* =$ pour $\mathcal{G}_{U_2}$. Ces valeurs offrent le meilleur équilibre entre recouvrement de nœuds et diversité des motifs.

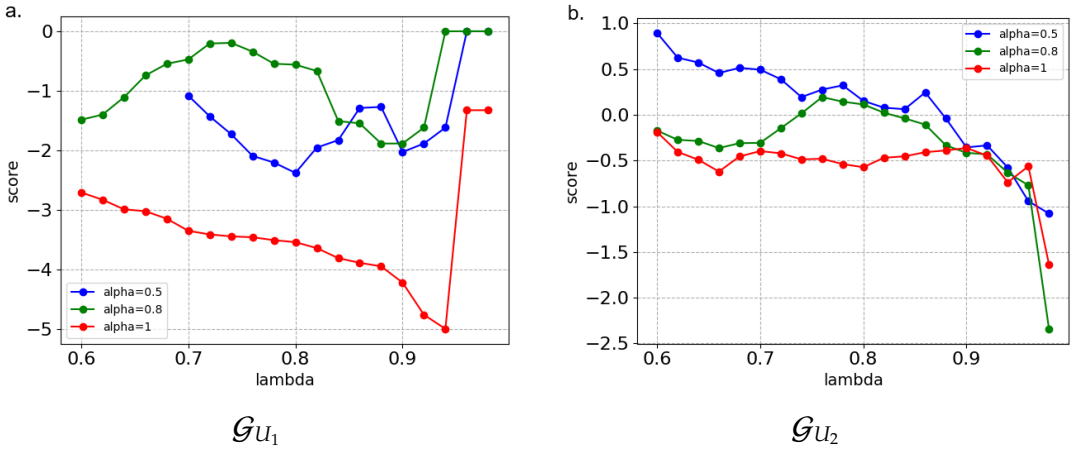


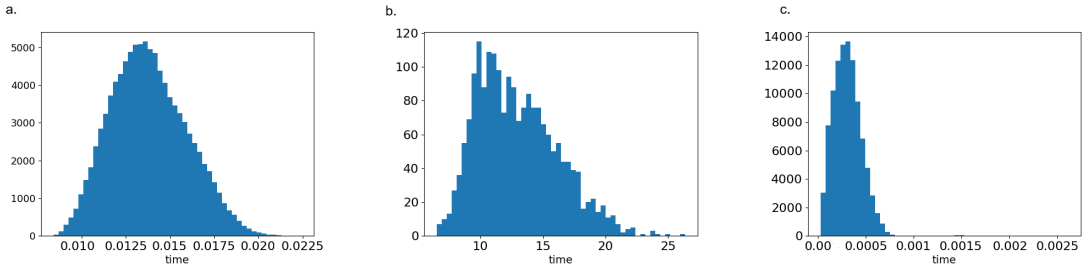
FIGURE 4.4 – Valeurs de $sc(\lambda)$ pour 'la scene politique française' et 'l'intelligence artificielle'.

Cette procédure d'optimisation de λ permet de choisir automatiquement une valeur pour le seuil de qualité de sorte à maximiser le nombre de nœuds couverts par les motifs et à minimiser leur redondance pour n'importe quel multigraphe de co-appartenance. Néan-

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

moins, cette procédure a l'inconvénient de nécessiter beaucoup de calculs. En effet, pour nos exemples, quand λ prend des valeurs inférieures à 0.6, les temps d'exécution de Exhaustif sont longs et les nombres de concepts sont grands. Ceci a pour effet de rendre le calcul de la mesure de similarité entre paires de concepts q_3 coûteux voire impossible dans certains cas. Une alternative est alors d'utiliser l'algorithme Top-K (algorithmes 4 et 5) pour obtenir les k motifs pour lesquels la somme des mesures de qualité est maximale et qui vérifient la condition de diversité : les similarités de Jaccard entre toutes les paires de motifs sont inférieures au paramètre $\delta \in [0, 1]$ choisi par l'utilisateur.

Étape (2) : Après avoir fixé le paramètre λ pour la méthode Exhaustif, nous utilisons la même valeur pour comparer les temps d'exécution des algorithmes d'échantillonnage. La première observation est que ConceptSampling a un temps d'échantillonnage très long. En effet, 500 itérations de la chaîne de Markov nécessitent en moyenne 6482 secondes ce qui est 20 fois plus long que le temps nécessaire à Exhaustif pour trouver les concepts du motif (308 secondes pour $\alpha = 0.8$ et $\lambda = 0.74$). De plus, le nombre d'itérations (500) est plus petit que le nombre d'états dans la chaîne de Markov (le nombre de concepts dans le contexte). Dans ce cas, nous ne pouvons pas être assuré de la convergence de la chaîne de Markov vers sa distribution stationnaire proportionnelle à la mesure de qualité Q. En résultat, les motifs échantillonnés par ConceptSampling sont de mauvaise qualité et ne feront pas partie de la comparaison qui suit. La figure 4.5 montre les temps d'échantillonnage pour les méthodes DirectSampling, MaximalSampling, et ClosedMaximalSampling sur le sous-graphe $\mathcal{G}_{U_1}$. Nous observons que le temps d'échantillonnage de DirectSampling est court et stable. Pour MaximalSampling, ce temps est en moyenne 30 fois plus long et a une plus grande variabilité. ClosedMaximalSampling est l'algorithme dont l'exécution est la plus lente à cause des multiples fermetures qui doivent être calculées pour tirer un seul échantillon. Les mêmes observations peuvent être faites pour le deuxième exemple de sous-graphe d'appartenance $\mathcal{G}_{U_2}$.



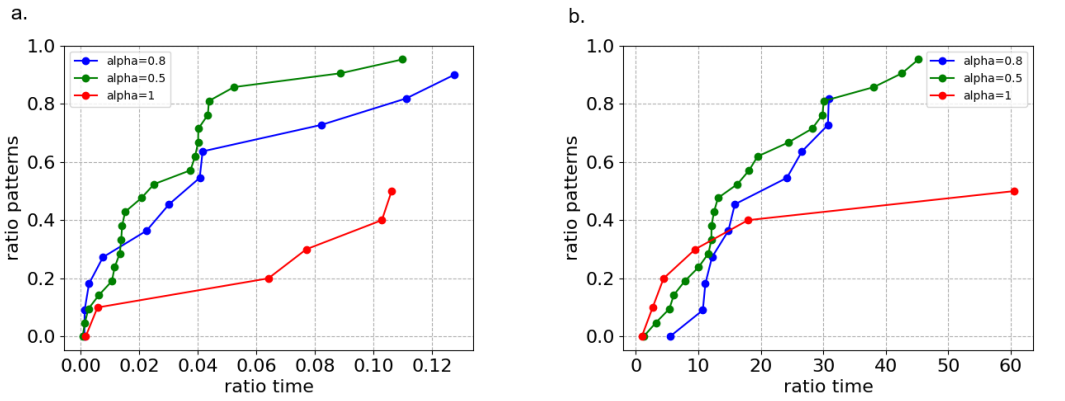
MaximalSampling ClosedMaximalSampling DirectSampling

Figure 4.5 – Distribution du temps moyen d'échantillonnage d'une communauté par MaximalSampling (10^5 échantillons), ClosedMaximalSampling (2500 échantillons) et DirectSampling (10^5 échantillons) pour $\mathcal{G}_{U_1}$.

Le temps d'échantillonnage n'est pas un critère suffisant pour comparer les méthodes d'échantillonnage entre elles. Il est important d'évaluer la proportion de motifs trouvés par chacune de ces méthodes relativement à l'ensemble total de motifs $\mathcal{M}_{\text{Exhaustif}}$, tout en s'assurant que les temps cumulés d'échantillonnage ne dépassent pas le temps d'exécution de Exhaustif. Pour cela, nous représentons dans la figure 4.6 l'évolution du ratio de motifs uniques trouvés par les méthodes d'échantillonnage relatif à $|\mathcal{M}_{\text{Exhaustif}}|$ en fonction du ratio

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

de temps d'échantillonnage cumulé relativement au temps d'exécution de Exhaustif. Nous ne rapportons pas, dans le figure, les résultats pour Exhaustif car cette méthode échoue à extraire un sous-ensemble significatif de motifs de $\mathcal{M}_{\text{Exhaustif}}$. Ceci est dû au fait que les motifs échantillonnés avec cet algorithme ne vérifient pas forcément la contrainte de qualité minimale et n'appartiennent pas obligatoirement à $\mathcal{M}_{\text{Exhaustif}}$. La figure 4.6 montre que MaximalSampling s'exécute bien plus rapidement que ClosedMaximalSampling tout en échantillonnant une grande proportion de l'ensemble exhaustif de motifs. En effet, les opérateurs de fermeture utilisés dans ClosedMaximalSampling ralentissent considérablement son exécution : pour $\alpha = 1$, et avec un temps d'échantillonnage cumulé 60 fois plus grand que le temps d'exécution de Exhaustif, ClosedMaximalSampling ne trouve que 50% des motifs de $\mathcal{M}_{\text{Exhaustif}}$. La méthode MaximalSampling semble la plus appropriée pour l'échantillonnage de motifs depuis nos exemples de contextes issus de graphes de co-appartenance. C'est la méthode d'échantillonnage que nous considérerons dans la suite pour les comparaisons aux méthodes compétitives.



MaximalSampling/ Exhaustif ClosedMaximalSampling/ Exhaustif

Figure 4.6 – Proportion de motifs distincts échantillonnés versus ratio de temps pour MaximalSampling (gauche) et ClosedMaximalSampling (droite) pour $\mathcal{G}_{U_1}$.

Éétape (3) : Les graphes de co-appartenance aux listes Twitter ont des informations textuelles associées aux nœuds et aux couches du graphe. Ce sont respectivement les biographies des utilisateurs Twitter et les descriptions des listes Twitter. Ces informations textuelles peuvent être utilisées pour caractériser de manière sémantique les motifs et les communautés de nœuds qui y correspondent. Ceci aide à comprendre le sens des communautés en cherchant l'information sémantique commune aux nœuds d'une même communauté. Nous proposons alors de caractériser chaque motif $(L_i, (V_i, K_i))$ et sa communauté de nœuds correspondante V_i avec l'ensemble des mots les plus fréquents qui apparaissent dans les attributs textuels des couches L_i et dans ceux des nœuds V_i. Pour donner une caractérisation visuelle de chaque motif et communauté de nœud correspondante trouvés par notre approche, nous utilisons un diagramme représentant les 12 mots les plus fréquents de l'ensemble $K_i \cup K'_i$ et un nuage de mots de cet ensemble dans lequel la taille de chaque mot est proportionnelle à sa fréquence. La figure 4.7 comporte deux exemples de caractérisation visuelle de communautés : un pour le sous-graphe de la 'scène politique française' et un autre pour le sous-graphe de 'l'intelligence artificielle'.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

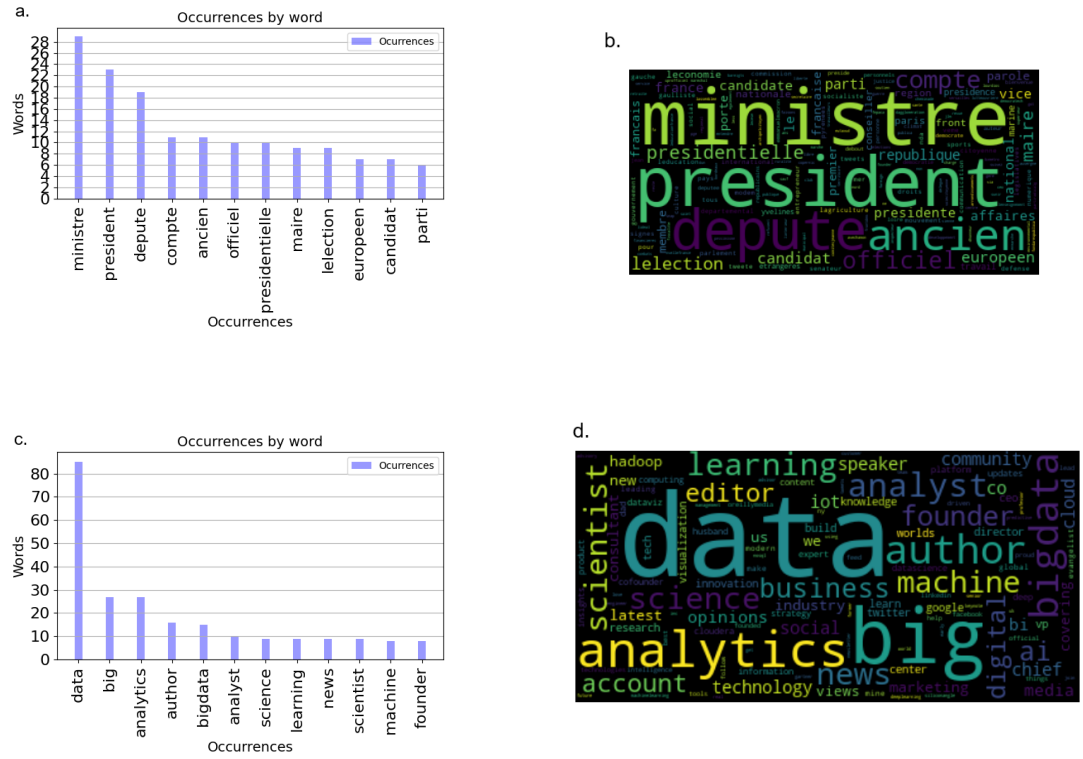


FIGURE 4.7 – Caractérisation sémantique des communautés extraites du graphe 'scène politique française' (haut) et du graphe 'intelligence artificielle' (bas) avec TOP-K.

On compare aussi les trois algorithmes EXHAUSTIF, TOP-K et MAXIMALSAMPLING en termes de statistiques de l'ensemble de motifs qu'ils retournent et de temps d'exécution (Table 4.3). Nous observons que le temps nécessaire à l'échantillonnage d'un motif avec MAXIMALSAMPLING est beaucoup plus petit que le temps d'exécution des algorithmes d'énumération EXHAUSTIF et TOP-K. Les communautés de nœuds issus des motifs retournés par MAXIMALSAMPLING recouvrent aussi une plus grande partie du graphe de co-appartenance. La raison est que MAXIMALSAMPLING échantillonne certains motifs non fermés qui ne sont pas retournés par les algorithmes d'énumération. Mais, MAXIMALSAMPLING est limité par le fait de ne pas pouvoir échantillonner tous les motifs de $\mathcal{M}_{\text{EXHAUSTIF}}$ et retourne plusieurs motifs redondants. Concernant les deux autres méthodes, l'approche TOP-K a l'avantage de retourner un ensemble de motifs dont la mesure de qualité est grande et qui sont en nombre limité ce qui facilite la tache à l'analyste pour comprendre les résultats.

Étape (4) : Commençons par certaines observations sur les communautés trouvées par les cinq méthodes compétitives sur les exemples de sous-graphes de co-appartenance Twitter en utilisant les statistiques de la Table 4.4 :

— En comparaison à nos trois méthodes, le partitionnement de nœuds retourné par l'algorithme Louvain recouvre tous les nœuds du graphe. Il contient moins de commu-

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Graphe	Méthodes	Nb clusters	Couverture (%)	Nb nœuds par motif	Qualité moyenne	Temps d'exécution (sec)
$\mathcal{G}_{U_1}$	Exhaustif	40	2.86	84.6 ± 51.6	0.85 ± 0.065	334
	Top_10	10	1.51	67.9 ± 51.6	0.98 ± 0.01	210
	MaximalSampling	558	6.98	75.8 ± 37	0.69 ± 0.11	690
$\mathcal{G}_{U_2}$	Exhaustif	121	1.82	79.5 ± 40.3	0.79 ± 0.05	3864
	Top_10	10	0.39	61.3 ± 29.4	0.87 ± 0.05	3748
	MaximalSampling	185	2.73	74.4 ± 41.5	0.63 ± 0.05	1640

Table 4.3 – Statistiques des ensembles de motifs extraits par les trois algorithmes depuis les contextes correspondants aux deux exemples de graphes Twitter. Pour MaximalSampling, le temps d'exécution est calculé pour 10^5 échantillons tirés.

	Méthodes	Nb clusters	Couverture (%)	Nb nœuds par motif	Temps d'exécution (sec)
$\mathcal{G}_{U_1}$	Louvain	25	100	1532 ± 1880	8.9
	ML-LCD	10	1.32	58 ± 44	46100
	LPV	26	5.1	216 ± 74	12870
	Mucha	50	15.9	122 ± 401	49937
	Infomap	2061	100	86 ± 39	56
$\mathcal{G}_{U_2}$	Louvain	20	100	7265 ± 11082	48.5
	ML-LCD	5	0.18	62 ± 28	__
	LPV	63	5.65	273 ± 382	28580
	Infomap	4720	100	115 ± 45	177

Table 4.4 – Statistiques des communautés retournées par les méthodes compétitives pour les deux exemples de graphes de co-appartenance Twitter.

nautés en comparaison à l'ensemble retourné par Exhaustif et ces communautés sont de plus grande taille. L'algorithme Louvain retourne une décomposition du graphe à une échelle macroscopique tandis que les communautés obtenues par calcul de motifs sont à une échelle plus locale. Pour l'exemple de la 'scène politique française', nous remarquons que les communautés de nœuds trouvées par Exhaustif et MaximalSampling sont des sous-graphes dans les trois communautés majeures du partitionnement Louvain qui correspondent aux trois plus grandes tendances de la scène politique française. Pour le deuxième exemple, les trois nœuds centraux sont regroupés dans la même communauté Louvain caractérisée par des mots fréquents comme 'ai' et 'research', alors que les communautés de nœuds issues des motifs sont plus locales et distribuées dans plusieurs régions du graphe. Les communautés de nœuds trou-

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

vées avec une des méthodes d'extraction de motifs sont des sous-communautés plus spécifiques en comparaison aux communautés Louvain. Elles contiennent moins de nœuds et ont des mesures d'exceptionnalité sémantique plus grandes. L'algorithme Louvain a l'avantage d'avoir un temps d'exécution plus court : il s'exécute en moins d'une minute pour les deux exemples de sous-graphes Twitter. Et aussi, les partitionnements retournés par l'algorithme Louvain recouvrent tous les nœuds du graphe. Mais, cet algorithme n'utilise qu'une représentation sur une couche du multigraphe et écarte l'information présente dans la multiplicité des couches du multigraphe de co-appartenance.

— L'algorithme ML-LCD permet de trouver une communauté locale dans un multigraphe à partir d'un nœud graine. Pour les deux exemples, nous avons lancé 10 exécutions de ML-LCD avec les 3 nœuds centraux de chaque exemple et 7 autres nœuds tirés aléatoirement du graphe. Il en résulte un ensemble de communautés dont le nombre est plus petit ou égal au nombre d'exécutions et dont la taille est similaire à celle des communautés trouvées avec les motifs. Dans les deux exemples Twitter, seule une petite proportion des nœuds du graphe est recouverte par l'ensemble de communautés issu des 10 exécutions de ML-LCD. L'algorithme a aussi l'inconvénient de nécessiter plusieurs heures pour une seule exécution qui ne retourne qu'une seule communauté.

— L'algorithme LPV a été spécifiquement implémenté pour trouver des communautés thématiques d'utilisateurs Twitter en utilisant leur appartenance aux listes Twitter. Pour nos expériences, nous avons exécuté LPV avec les valeurs de paramètres : $\rho = 0.1$, $\tau = 0.2$ et $\mu = 0.05$. L'algorithme est coûteux en calcul car il nécessite de multiples exécutions d'un autre algorithme de détection de communautés OSLOM sur un graphe de listes dérivé du graphe de co-appartenance initial des utilisateurs. Pour les deux exemples considérés, les communautés retournées par LPV ne recouvrent pas la totalité des nœuds du graphe de co-appartenance.

— L'algorithme Infomap s'exécute très rapidement pour les deux exemples de sous-graphes de co-appartenance Twitter et retourne un partitionnement des nœuds avec un recouvrement de 100%. Mais, le nombre de communautés trouvées par Infomap est très grand en comparaison aux autres méthodes et ses communautés sont recouvrantes : un même nœud peut appartenir à de multiples communautés. La raison de cela est que Infomap considère différentes occurrences d'un même nœud dans différentes couches comme des nœuds distincts qui peuvent appartenir à des communautés distinctes.

— L'algorithme Mucha a l'inconvénient de nécessiter un long temps d'exécution qui semble proportionnel au nombre de couches dans le multigraphe qui est grand dans les exemples de graphes de co-appartenance Twitter. Ceci empêche sont utilisation pour l'exemple de sous-graphe de 'l'intelligence artificielle' et son exécution dure 14 heures pour le sous-graphe de la 'scène politique française'.

Pour chacune de ces méthodes nous calculons la moyenne des TF-IDF des mots de leurs attributs textuels et la divergence de Kullback-Leibler entre les distributions de probabilité correspondantes. Nous utilisons les attributs textuels associés aux nœuds des multigraphes de co-appartenance Twitter : ce sont les biographies associées aux utilisateurs du réseau social. Cette information n'a été utilisée dans aucun des algorithmes de détection de communautés. La figure 4.8 représente la distribution des moyennes de TF-IDF pour chaque

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

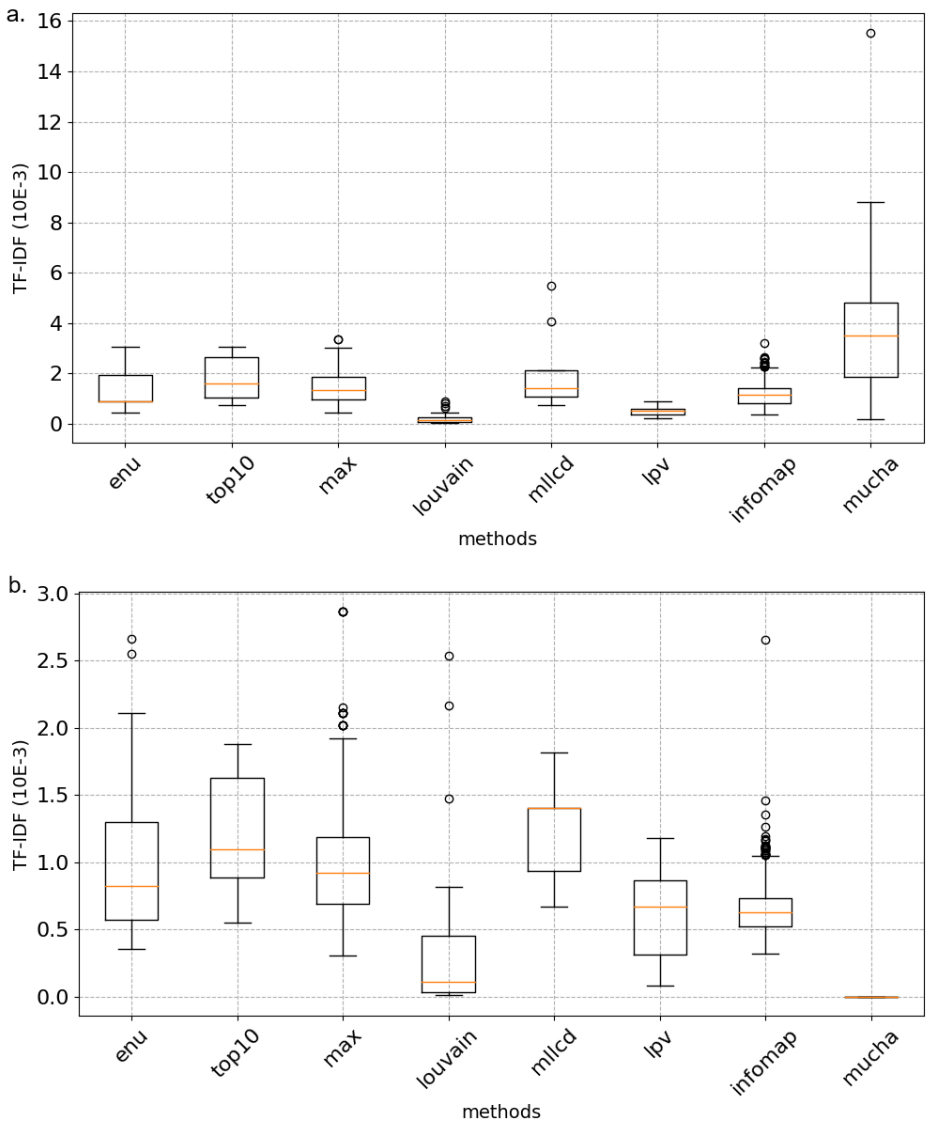


FIGURE 4.8 – Distributions des moyennes de TF-IDF des mots des biographies associées aux membres des communautés trouvées par les différentes méthodes. En haut : la 'scène politique française' et en bas 'l'intelligence artificielle'.

méthode et pour les deux exemples de sous-graphes de co-appartenance Twitter. Pour le premier exemple de la 'scène politique française', la méthode TOP-K (Top-10) a la deuxième plus grande valeur médiane après Mucha. Par contre, la distribution du TF-IDF moyen pour la méthode Mucha a une plus grande variabilité (plus grand écart-type). Nous rappelons aussi que l'algorithme Mucha nécessite plus de ressources de calcul en comparaison aux méthodes d'extraction de motifs présentées. Pour l'exemple de 'l'intelligence artificielle', c'est ML-LCD qui a une plus grande valeur médiane pour la distribution des TF-IDF moyens plus grande que celle de TOP-K. Mais, ML-LCD s'exécute moins rapidement que TOP-K et ses communautés ne recouvrent qu'un sous-ensemble des nœuds du graphe.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

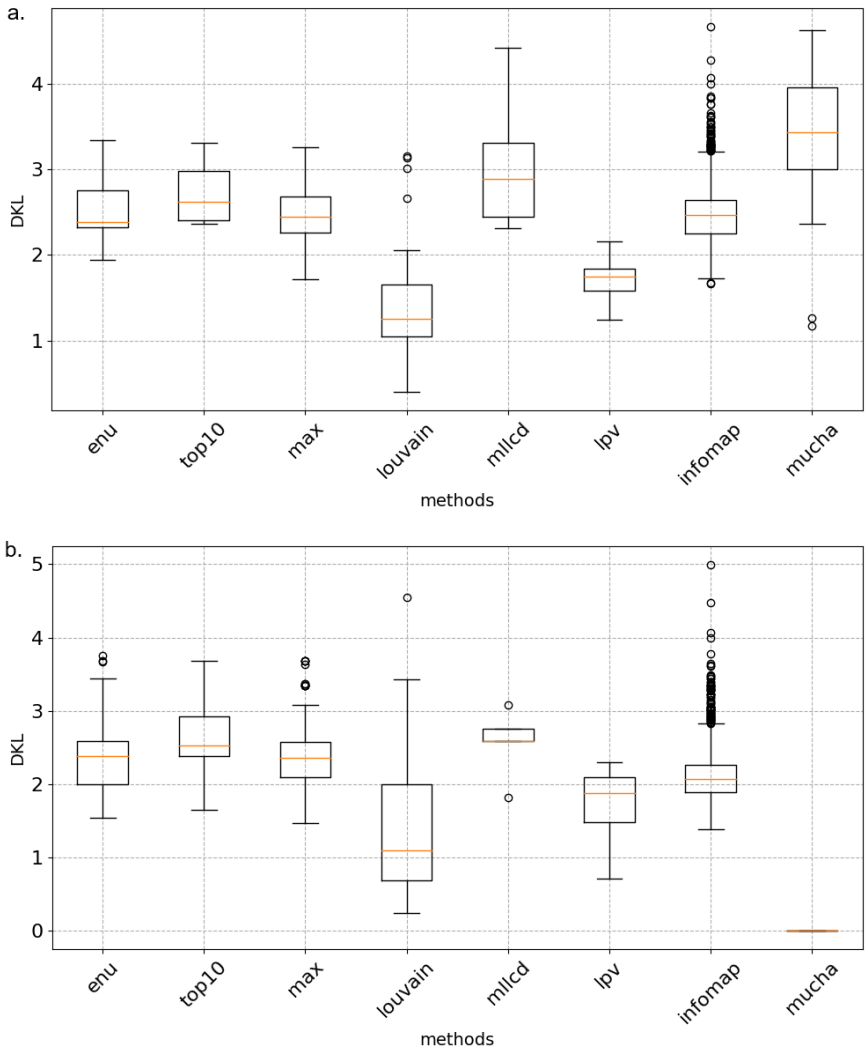


FIGURE 4.9 – Distributions des divergences de Kullback-Leibler des biographies associées aux membres des communautés trouvées par les différentes méthodes. En haut : la 'scène politique française' et en bas 'l'intelligence artificielle'.

De manière similaire, la figure 4.9 représente les distributions des divergences de Kullback Leibler. Elles sont calculées entre la distribution de probabilité générée par les mots des attributs associés aux nœuds d'une communauté et la distribution globale générée par les mots associés à tous les nœuds du sous-graphe de co-appartenance. Pour le premier exemple, ML-LCD et Mucha ont une valeur médiane pour la distribution des divergences de Kullback-Leibler plus grande que celle des méthodes d'extraction de motifs. Mais, ces deux algorithmes nécessitent plus de ressources de calcul et ML-LCD a un plus petit ratio de recouvrement de nœuds. Pour le deuxième graphe, seul l'algorithme ML-LCD a une valeur médiane de la distribution des divergences de Kullback-Leibler plus grande que nos méthodes mais cela au prix d'un temps d'exécution plus long et un plus petit ratio de recouvrement des nœuds du sous-graphe de co-appartenance.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

En conclusion, pour les exemples de sous-graphes de co-appartenance Twitter, la méthode Top-K retourne un ensemble de communautés dont les mesures d'exceptionnalité sémantique sont plus grandes en comparaison aux méthodes compétitives. Ces mesures pour Mucha et ML-LCD sont similaires mais ces méthodes ont une exécution plus lente. De plus, ML-LCD est une méthode locale qui nécessite de choisir un nœud graine et le nombre de nœuds du graphe couverts par ses communautés reste limité même avec 10 exécutions en utilisant des nœuds graines différents.

Étude qualitative d'un exemple de communauté

On représente dans la figure 4.10 une communauté de nœuds issue d'un concept extrait avec Exhaustif depuis le contexte d'itemsets du multigraphe de co-appartenance $\mathcal{G}_{U_1}$ de la 'scène politique française'. La communauté fait partie du cluster 5 dans le partitionnement Louvain (Figure 3.2). Le concept a une mesure de qualité $Q = 0.827$ et la communauté de nœuds contient 21 utilisateurs Twitter qui apparaissent dans trois couches du multigraphe. Ce qui est intéressant par rapport à cette communauté c'est qu'elle contient exclusivement des membres et collaborateurs du parti politique de la majorité gouvernementale en France : LREM. Ceci montre l'aptitude de nos méthodes à extraire des communautés ciblées et spécifiques à la différence de l'algorithme Louvain qui retourne un partitionnement comportant des communautés plus générales de plus grande taille. L'algorithme ML-LCD trouve aussi des communautés de nœuds ciblées dans le multigraphe de co-appartenance $\mathcal{G}_{U_1}$. En effet, pour ce sous-graphe de la 'scène politique française', les trois communautés trouvées par ML-LCD en utilisant comme nœuds graines les nœuds centraux du domaine d'intérêt font partie de l'ensemble de communautés extraits avec Exhaustif. Mais, ce dernier s'exécute bien plus rapidement et retourne les résultats en 1000 fois moins de temps environ.

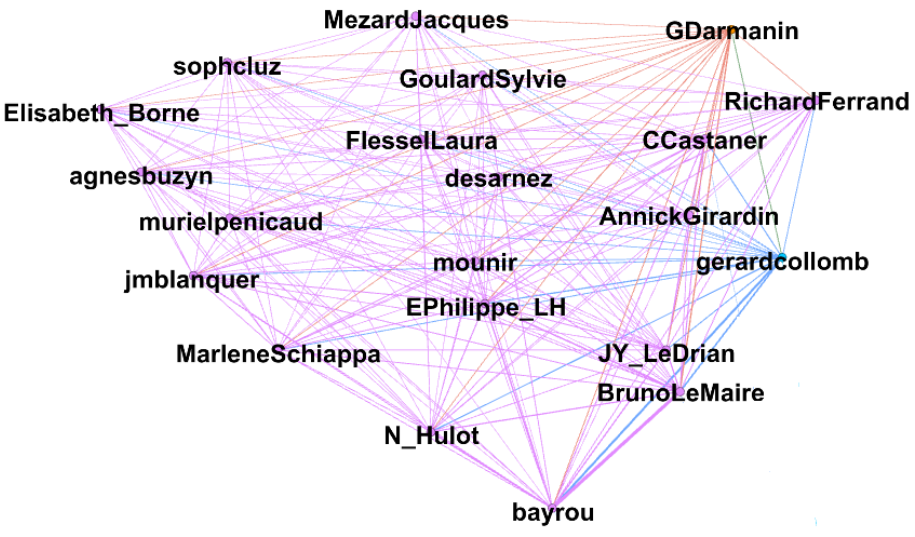


Figure 4.10 – Exemple d'une communauté extraite de $\mathcal{G}_{U_1}$ avec Exhaustif.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

4.4.2 Étude de l'encyclopédie collaborative Wikipedia

Afin d'approfondir la validation des résultats de nos méthodes, nous les évaluons sur un deuxième exemple de graphe de co-appartenance : le graphe de co-citations des pages Wikipedia. Le jeu de données contient 1.791.489 pages propagées dans 17.801 catégories et 28.511.807 citations [2]. Dans le graphe de co-appartenance correspondant, les citations correspondent aux nœuds et les pages aux couches. Chaque couche est annotée avec le titre de la page correspondante. Les catégories peuvent être utilisées comme une 'vérité terrain' pour l'évaluation des performances des méthodes de détection communautés et la comparaison aux méthodes compétitives. Nous suivons les mêmes étapes de la méthodologie présentée au début de la section 4.4.

Statistiques d'exécution

Tout d'abord, nous appliquons la procédure de construction de sous-graphes d'intérêt *Marche aléatoire* (algorithme 1 vu au chapitre 3) Wikipedia pour le domaine d'intérêt 'Tennis' en utilisant les trois nœuds centraux : 'Wimbledon Championships', 'Rafael Nadal' et 'ATP'. Il en résulte un sous-graphe $\mathcal{G}_{U_3}$ qui est lui-même un multigraphe de co-appartenance de plus petite taille que le graphe original et constitué de 29.931 nœuds, 12.407.483 liens et 1.344 couches.

Ensuite, nous exécutons l'algorithme Exhaustif sur $\mathcal{G}_{U_3}$ avec plusieurs valeurs pour les paramètres α et λ ce qui produit les nombres de motifs extraits et les temps d'exécution représentés en figure 4.11.

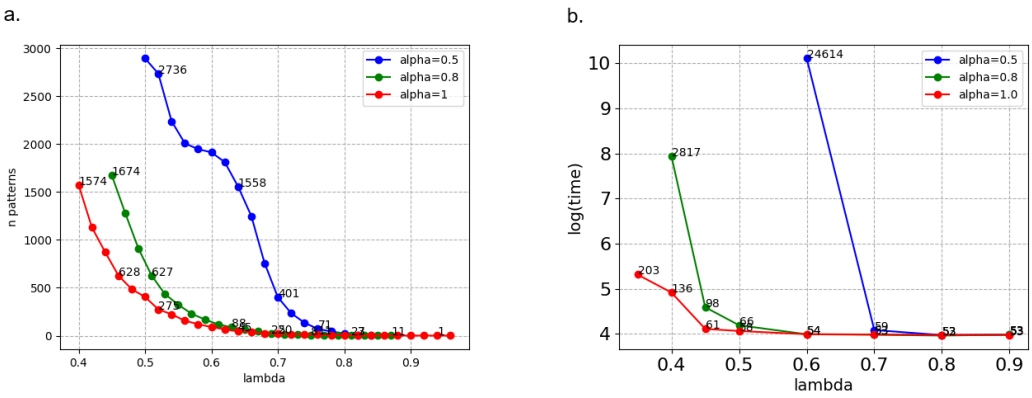


Figure 4.11 – Nombre de communautés distinctes retournées par Exhaustif (à gauche) et les temps d'exécution avec une échelle logarithmique (à droite), en fonction de λ avec différentes valeurs pour α (voir légende).

Optimisation λ

Pareillement à l'exemple Twitter, nous cherchons la valeur optimale λ^* du seuil de qualité minimal λ pour Exhaustif pour des valeurs de λ dans l'intervalle I.

2. https://snap.stanford.edu/data/wiki-topcats.html

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

La recherche de λ^* se fait en utilisant le score $sc(\lambda)$, défini précédemment à l'étape (1) de la section 4.4.1, qui favorise les ensembles de motifs tels que le recouvrement de nœuds du graphe est maximal et la redondance entre motifs de l'ensemble est minimale. La figure 4.12 représente les mesures de recouvrement, de redondance et le score résultant pour les ensembles de motifs obtenus par EXHAUSTIF avec différentes valeurs pour les paramètres α et λ.En utilisant les courbes à droite de la figure 4.12, nous montrons que $\lambda^* = \arg\max_\lambda(sc(\lambda)) = 0.5$ pour $\alpha = 0.8$. Pour cet exemple de graphe, les trois courbes de score, correspondant aux trois valeurs du paramètre α, sont décroissantes ce qui veut dire que λ^* est la plus petite valeur dans l'intervalle des valeurs considérés I.

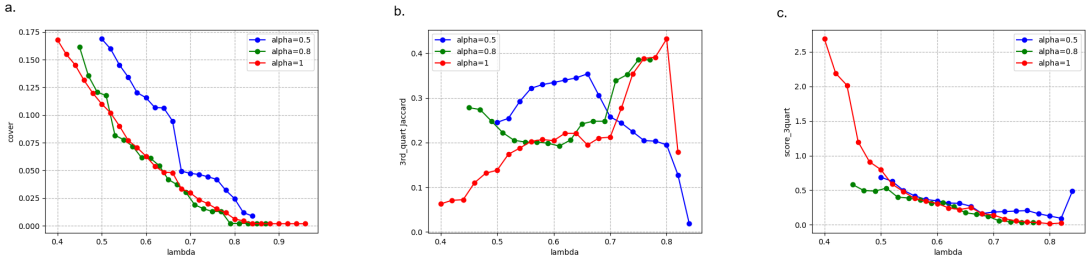


FIGURE 4.12 – $co(\lambda)$, $q_3(\lambda)$ et $sc(\lambda)$ pour le graphe de co-appartenance Wikipedia.

TOP-K

On applique aussi l'algorithme TOP-K sur le contexte issu du graphe de co-appartenance $\mathcal{G}_{U_3}$ pour trouver le sous-ensemble de motifs de cardinalité k dont la somme des mesures de qualité est maximale et la redondance est minimale. La figure 4.13 représente le temps d'exécution de TOP-K en fonction des paramètres k et δ, et la mesure de qualité moyenne des motifs pour différentes valeurs de k. Nous observons que la qualité moyenne de l'ensemble des top-k motifs diversifiés décroît avec le paramètre k, tandis que le temps d'exécution augmente ce qui montre l'efficacité de la stratégie d'élagage sur le paramètre $\min_Q$. De plus, à partir d'une certaine valeur pour k (6 pour $\delta = 0.4$ et 7 pour $\delta \in \{0.6, 0.8\}$), l'augmentation du temps d'exécution devient exponentielle.

Échantillonnage

Par la suite, nous testons l'algorithme MAXIMALSAMPLING et sa version avec fermetures CLOSEDMAXIMALSAMPLING sur le contexte issu du graphe de co-citations Wikipedia $\mathcal{G}_{U_3}$. Les histogrammes des temps d'échantillonnage pour les deux algorithmes sont représentés en figure 4.14. Il apparaît que CLOSEDMAXIMALSAMPLING nécessite 200 fois plus de temps de calcul que MAXIMALSAMPLING pour tirer aléatoirement un motif. Nous représentons aussi, dans la figure 4.14, le ratio de motifs échantillonnés en fonction du ratio de temps de calcul pour les deux algorithmes d'échantillonnage par rapport à l'algorithme exhaustif EXHAUSTIF. Nous observons que : avec un budget-temps plus petit que celui de l'énumération exhaustive, MAXIMALSAMPLING n'échantillonne que 40% des motifs énumérés avec $\alpha = 1$. Ceci limite l'utilité de l'échantillonnage de motifs pour cet exemple de graphe de co-appartenance $\mathcal{G}_{U_3}$ car sa taille est relativement petite ce qui rend l'énumération rapide.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

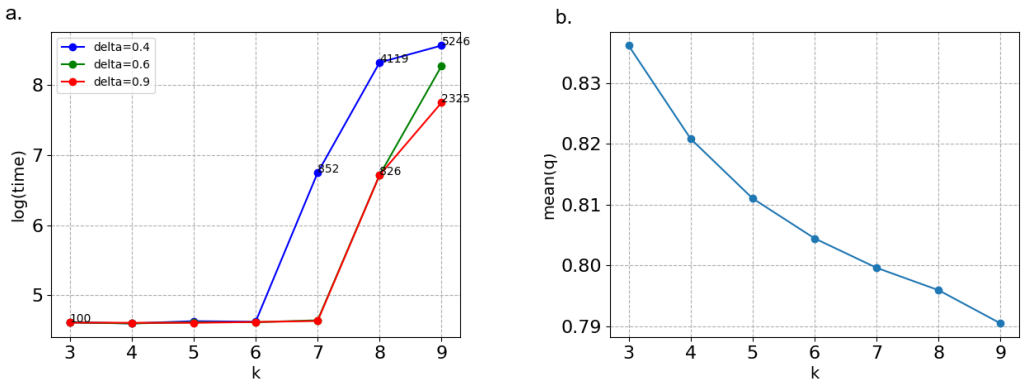


FIGURE 4.13 – Logarithme des temps d'exécution de l'algorithme TOP-K en fonction de k et de δ (à gauche). Qualité moyenne des motifs extraits pour $\delta = 0.8$ en fonction de k (à droite).

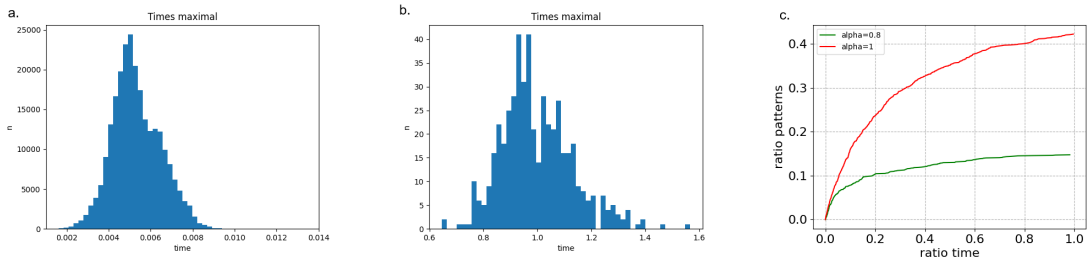


FIGURE 4.14 – Distributions des temps d'échantillonnage pour MAXIMALSAMPLING (à gauche) et CLOSEDMAXIMALSAMPLING (au milieu) avec 2.500 échantillons. Ratio de motifs distincts échantillonnés vs. ratio de temps d'exécution pour MAXIMALSAMPLING par rapport à EXHAUSTIF (à droite). Résultats obtenus pour le graphe Wikipedia.

Communautés et caractérisation sémantique

De manière similaire aux exemples Twitter, nous pouvons caractériser les communautés de nœuds trouvées par nos méthodes à l'aide des mots les plus fréquents dans les attributs textuels associés aux nœuds de la communautés et ceux des couches dans lesquelles ils apparaissent. Ici, les attributs textuels des nœuds sont les introductions des pages Wikipedia. Un exemple de caractérisation d'une communauté obtenue avec TOP-K est présenté à la Figure 4.15.

Évaluation et comparaison

Dans la table 4.6, nous synthétisons les statistiques des ensembles de communautés retournés par chacune des méthodes compétitives. L'algorithme Louvain partitionne le sous-graphe de co-appartenance $\mathcal{G}_{U_3}$ en un petit nombre de grandes communautés. Les algorithmes ML-LCD et Mucha s'exécutent lentement en comparaison aux méthodes présentées (Table 4.5). L'algorithme Infomap trouve un très grand ensemble de communautés chevauchantes avec peu de temps de calcul.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

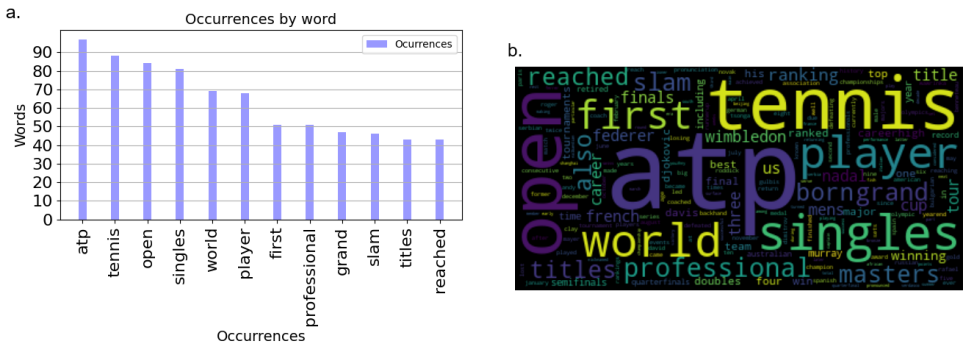


FIGURE 4.15 – Exemple de caractérisation sémantique d'une communauté extraite avec TOP-K depuis le graphe de co-appartenance Wikipedia.

Méthodes	Nb clusters	Couverture (%)	Nb nœuds par motif	Qualité moyenne	Temps d'exécution (sec)
EXHAUSTIF	102	5.77	108 ± 40	0.674 ± 0.048	60
Top_10	10	1.4	110 ± 50	0.786 ± 0.04	1624
MAXIMALSAMPLING	125	6.14	99 ± 51	0.648 ± 0.043	500

TABLE 4.5 – Statistiques des ensembles de motifs retournés par les trois algorithmes pour le graphe de co-appartenance Wikipedia. Pour MAXIMALSAMPLING, la moyenne et l'écart-type des temps d'échantillonnage sont donnés pour 10^5 tirages.

Méthodes	Nb clusters	Couverture (%)	Nb nœuds par motif	Temps d'exécution (sec)
Louvain	22	100	1360 ± 1669	8.1
ML-LCD	8	4.95	192 ± 261	24532
Mucha	25	14.6	175 ± 369	8880
Infomap	1237	100	84 ± 149	36

TABLE 4.6 – Statistiques des ensembles de motifs retournés par les méthodes compétitives pour le graphe de co-appartenance Wikipedia.

On compare les communautés trouvées grâce à l'extraction de motifs aux communautés retournées par les méthodes compétitives. Nous utilisons les paragraphes d'introduction des pages Wikipedia comme source de données externe et indépendante pour estimer l'exceptionnalité sémantique des communautés dans le but d'évaluer leur qualité et comparer les méthodes. A cet effet, nous employons des métriques comme le TF-IDF et la divergence de Kullback-Leibler pour comparer les mots utilisés dans les introductions des pages d'une communauté au vocabulaire de mots utilisés dans les pages du graphe de co-appartenance $\mathcal{G}_{U_3}$. De plus, pour cet exemple, une vérité terrain du partitionnement en communautés est disponible : c'est la classification des pages Wikipedia en catégories. Elle permet de se

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

comparer objectivement aux méthodes en calculant le F1-score entre leurs communautés et la vérité terrain.

Dans la Figure 4.16, nous pouvons voir que nos trois algorithmes trouvent des communautés telles que la distribution des moyennes de TF-IDF et celle des divergences de Kullback-Leibler ont les plus grande médianes en comparaison aux méthodes compétitives. L'optimisation de modularité avec l'algorithme Mucha et la méthode locale ML-LCD ont des médianes comparables mais leur temps d'exécution est plus long. Pour cet exemple, l'algorithme Mucha s'est exécuté en 6 heures et ML-LCD en un peu plus de 2 heures c'est qui est bien supérieur aux temps d'exécution de nos algorithmes qui ne dépassent pas 30 minutes (Tables 4.5 et 4.6).

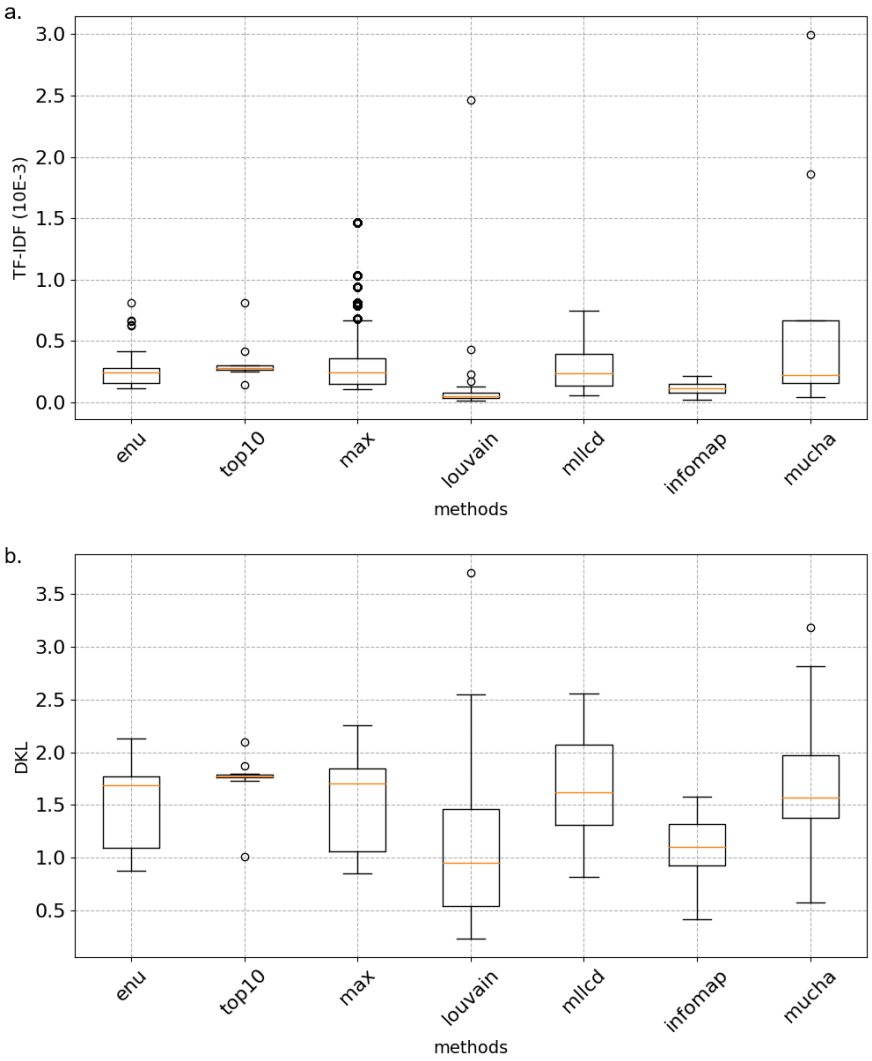


FIGURE 4.16 – Distributions des moyennes de TF-IDF (en haut) et des divergences de Kullback-Leibler (en bas) pour les mots des biographies des utilisateurs dans les communautés extraites par nos algorithmes et par les méthodes compétitives depuis le graphe de co-appartenance Wikipedia.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

La figure 4.17 montre le F1-score calculé, pour chaque méthode, entre les communautés extraites et les catégories Wikipedia considérées comme vérité terrain du partitionnement de $\mathcal{G}_{U_3}$. Les algorithmes Exhaustifλ^* et Top-K trouvent des ensembles de communautés avec la plus grande valeur médiane pour la distribution des F1-scores. En conclusion, pour l'exemple du multigraphe de co-appartenance Wikipedia, les communautés trouvées par les méthodes d'extraction de motifs d'itemsets sont celles qui ont la plus grande exceptionnalité sémantique, la plus grande correspondance à la réalité terrain du partitionnement et un temps d'exécution parmi les plus courts.

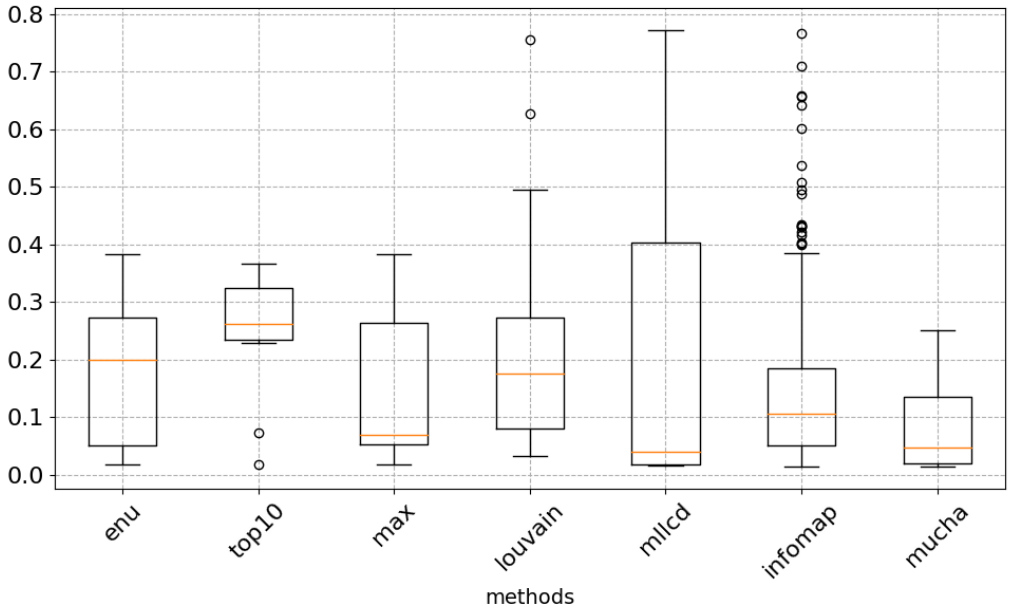


Figure 4.17 – F1-score entre les communautés trouvées par les différentes méthodes et la réalité-terrain des communautés issue des catégories pour le graphe de co-appartenance Wikipedia.

4.5 Conclusion

Dans ce chapitre, nous avons introduit l'algorithme d'énumération exhaustive Exhaustif et son variant Top-K qui permettent d'extraire des communautés ciblées depuis les multigraphes de co-appartenance. Nous avons aussi étudié plusieurs méthodes d'échantillonnage qui permettent de tirer aléatoirement ces communautés avec un coût de calcul très limité en comparaison aux algorithmes d'énumération. Ces méthodes sont utiles quand la taille du graphe à analyser est grande et que l'énumération exhaustive est infaisable. Nous avons observé que l'algorithme MaximalSampling est celui qui a la meilleure performance en termes de temps d'échantillonnage et de ratio de motifs échantillonnés par rapport aux motifs énumérés. Toutes ces méthodes utilisent la fonction de qualité Q et la contrainte qui lui est associée $P : Q > \lambda$ pour déterminer si un motif symbolique et le sous-graphe qui lui correspond constituent une communauté dans le multigraphe. La fonction de qualité Q

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

est une somme convexe des similarités structurelles et sémantiques des sous-graphes du multigraphe de co-appartenance.

Les communautés trouvées par notre approche sont ciblées et à une échelle plus petite quand nous les comparons aux communautés Louvain : dans un même cluster Louvain, nous trouvons plusieurs communautés obtenues avec notre approche. De plus, elles sont sémantiquement homogènes et exceptionnelles par rapport au reste du graphe. Ceci est montré par leurs distributions de TF-IDF moyen et de divergence de Kullback-Leibler dont les médianes sont supérieures à celles des méthodes compétitives Louvain, LPV et Infomap. Les algorithmes ML-LCD et Mucha trouvent des communautés dont les mesures d'exceptionnalité sont similaires à celles des communautés trouvées par notre approche mais les deux méthodes nécessitent bien plus de ressources de calculs.

Grâce à la prise en compte des attributs sémantiques du multigraphe de co-appartenance, les méthodes d'extraction de motifs symboliques sous contraintes permettent d'obtenir de meilleurs résultats que les méthodes de la littérature pour la détection de communautés dans les multigraphes. C'est la conclusion de la partie expérimentale dans laquelle nous avons utilisé deux exemples de multigraphes de co-appartenance et des métriques d'évaluation basées sur l'exceptionnalité sémantique des attributs textuels de la communauté par rapport à l'ensemble des attributs textuels du multigraphe.

Jusque là, l'analyse de l'information sémantique contenue dans les attributs textuels est limitée à des comparaisons entre des distributions de mots. La similarité entre deux attributs textuels dépend directement du nombre de mots qu'ils ont en commun. Dans le prochain chapitre, nous verrons comment analyser l'information sémantique du multigraphe de co-appartenance d'une meilleure façon. Pour cela, nous utiliserons un modèle de réseau de neurones pour le traitement du langage naturel qui permet de représenter les phrases (ou séquences de mots) dans un espace vectoriel. Les représentations obtenues seront utilisées pour calculer les similarités sémantiques entre les attributs textuels du multigraphe de co-appartenance. Ce dernier pourra alors être représenté par un jeu de données numérique duquel nous extrairons les motifs numériques qui correspondent à des communautés.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Chapitre 5

Détection de communautés par extraction de motifs numériques

Nous avons vu dans le chapitre précédent l'utilisation des motifs symboliques d'un contexte formel pour la détection de communautés dans les multigraphes de co-appartenance. Mais cette approche a certaines limites qui lui sont propres. Tout d'abord, le nombre d'attributs binaires nécessaires est grand et augmente rapidement avec la taille du multigraphe. En effet, dans le contexte formel, le nombre d'attributs est égal à la somme du nombre de nœuds et du nombre de mots dans le vocabulaire. L'augmentation du nombre d'attributs pénalise considérablement les méthodes d'extraction de motifs. Un autre inconvénient de l'approche basée sur des attributs binaires est qu'elle ne prend pas en compte la totalité de l'information contenue dans les descriptions textuelles des groupes. Les mots y sont considérés séparément et l'ordre dans lequel ils apparaissent dans les phrases n'est pas pris en compte dans le contexte formel. Pour répondre à ces limites, nous proposons dans ce chapitre de transformer le contexte formel en un jeu de données numériques dans lequel les motifs correspondent à des communautés dans le multigraphe de co-appartenance. Pour cela, nous utilisons les dernières avancées faites dans le domaine du traitement du langage naturel pour construire le jeu de données numériques qui regroupe la relation d'appartenance des entités aux groupes et les similarités sémantiques entre les descriptions de groupes. Par la suite, nous utilisons le formalisme des structures de motifs (Ganter and Kuznetsov, 2001) et plus particulièrement les motifs d'intervalles (Kaytoue et al., 2011) pour identifier les ensembles de nœuds qui forment des communautés dans le multigraphe de co-appartenance.

Nous rappelons que les données de départ sont les ensembles de groupes $\mathcal{L}$, d'entités $\mathcal{U}$ et des mots du vocabulaire $\mathcal{K}$, et les applications v et k qui retournent respectivement les entités appartenant à un groupe, et les mots utilisés dans sa description. Dans le chapitre 3, nous avons construit un contexte $\mathcal{C} = (\mathcal{L}, (\mathcal{U}, \mathcal{K}), \mathcal{D})$ qui est un jeu de données binaire regroupant les associations entre les groupes $\mathcal{L}$ d'un cote, et les entités $\mathcal{U}$ et les mots du vocabulaire $\mathcal{K}$ de l'autre. Dans la suite, la relation binaire d'appartenance des entités aux groupes sera étendue pour qu'elle prenne en compte la similarité sémantique entre les descriptions de groupes de sorte à construire un jeu de données à valeurs numériques réelles recoupant les entités et les groupes. La recherche de motifs dans ce jeu de données permet d'obtenir des sous-ensembles d'entités ayant des valeurs d'appartenance généralisée aux groupes similaires. Ces sous-ensembles d'entités correspondent à des ensembles

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

de nœuds dans le multigraphe de co-appartenance $\mathcal{G} = (\mathcal{U}, E, \mathcal{L}, k)$. Après avoir défini une fonction de qualité et des contraintes appropriées, nous pouvons sélectionner les top-k motifs qui correspondent à des communautés de nœuds homogènes dans le multigraphe de co-appartenance. A cet effet, nous proposons une procédure de construction du jeu de données numérique à partir des données de départ en utilisant différentes similarités dont la similarité cosinus entre les représentations vectorielles des descriptions de groupes obtenues avec la méthode de traitement de langage naturel BERT (Devlin et al., 2019). Nous introduisons ensuite l'algorithme MIMETIC pour l'extraction des top-k motifs d'intervalles fermés, fréquents et diversifiés depuis le jeu de données numérique. Les extensions de ces motifs sont des ensembles d'entités et correspondent à des communautés dans le multigraphe de co-appartenance $\mathcal{G} = (\mathcal{U}, E, \mathcal{L}, k)$. Comme vérifié dans les expériences, les communautés trouvées par notre approche sont les plus importantes dans le graphe et sont obtenues avec beaucoup moins de ressources de calcul. Nous montrons aussi que l'information sémantique liée aux couches du multigraphe complète l'information purement topologique du multigraphe et permet d'obtenir des communautés plus précises par rapport à la réalité terrain et de rendre l'approche moins sensible au bruit.

Ce chapitre présente quatre contributions :

(1) la formalisation du problème de détection de communautés dans les graphes de co-appartenance répandus en pratique
(2) l'utilisation de techniques de traitement du langage naturel pour évaluer les similarités entre couches
(3) la proposition d'une méthode originale pour la détection de communautés basée sur l'extraction de k motifs d'intervalles diversifiés
(4) les expériences qui montrent la capacité de la méthode présentée à identifier des communautés dans trois jeux de données différents et avec de meilleurs résultats que les méthodes compétitives

Le reste du chapitre est organisé comme suit : la Section 5.1 présente l'extraction de motifs dans des données numériques et les principales définitions liées au formalisme des structures d'intervalles dont les motifs d'intervalles sont un cas particulier. La Section 5.2 introduit les méthodes de traitement du langage naturel (NLP) qui nous permettent de transformer un texte en vecteurs numériques. Dans la Section 5.3, nous présentons la relation entre les motifs d'intervalles et la détection de communautés dans les multigraphes de co-appartenance ainsi que les procédures et algorithmes développés. Enfin, nous décrivons dans la Section 5.4 les expériences et les résultats obtenus.

5.1 Extraction de motifs dans des données numériques

Les contextes formels sont réservés aux jeux de données contenant des attributs symboliques (booléens). Le cadre plus général des structures de motifs (Ganter and Kuznetsov, 2001) permet de remplacer les sous-ensembles d'attributs binaires M des contextes formels par une forme plus général de descriptions $(D, \sqcap)$. Les structures de motifs sont un moyen de considérer une plus grande variété de langages de descriptions à condition que celles-ci puissent être ordonnées dans un demi-treillis inférieur. En d'autres termes, une structure de motifs doit vérifier la condition statuant que tout ensemble de descriptions admet une borne inférieure.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Dans la suite, nous commencerons par présenter plus en détails le cadre théorique des structures de motifs puis nous définirons les motifs d'intervalles dans un jeu de données numériques et l'opérateur de fermeture qui leur est associé. Et pour finir, nous présenterons un état de l'art des méthodes d'extraction des motifs d'intervalles fermés.

5.1.1 Structures de motifs

Même si le cadre de l'Analyse Formelle des Concepts permet la généralisation des attributs binaires à des attributs plus complexes, il nécessite pour cela un redimensionnement de ces attributs complexes en les remplaçant par plusieurs attributs binaires. En particulier, le redimensionnement interordinal d'une variable numérique nécessite deux fois plus de variables binaires que la cardinalité de son domaine. Pour les jeux de données contenant de nombreux attributs numériques dont le domaine est grand, le nombre d'attributs binaires dans le contexte formel après redimensionnement explose et peut rendre l'extraction des motifs infaisable. Pour éviter ce problème, un cadre formel qui permet de prendre en charge les attributs complexes a été introduit dans (Ganter and Kuznetsov, 2001) sous le nom des structures de motifs.

Définition 20. ***Une structure de motifs*** $\mathcal{S} = (G, (D, \sqcap), \delta)$ *est donnée par :*

- *un ensemble d'objets* G
- *un demi-treillis inférieur des descriptions des objets* $(D, \sqcap)$ *avec* $\sqcap$ *est l'opérateur de disjonction (meet) qui retourne le plus grand élément des minorants d'un sous-ensemble* $S \subseteq D$
- δ *l'application associant les descriptions aux objets*

Les motifs *sont des éléments de* D *et sont ordonnés à l'aide de l'opérateur* $\sqsubseteq$ *défini comme suit :* $c \sqcap d = c \iff c \sqsubseteq d$. *Dans ce cas, nous disons que le motif* c *est* ***moins restrictif*** *que le motif* d.

Intuitivement, une structure de motifs est un ensemble d'objets avec des descriptions D et un opérateur de similarité $\sqcap$. Étant donné un ensemble arbitraire d'objets, l'opérateur de similarité retourne la description commune aux objets de l'ensemble. L'opérateur $\sqcap$ est idempotent, commutatif et associatif.

Exemple 5.1. Les structures de motifs permettent de généraliser l'Analyse Formelle des Concepts à des jeux de données dans lesquels les attributs ne sont pas nécessairement binaires. La conséquence est que tout contexte formel $C = (G, M, I)$ peut être représenté par une structure de motifs $S = (G, (D, \sqcap), \delta)$ où :

- $D = \mathcal{P}(M)$
- $\sqcap = \cap$
- $\forall g \in G, \delta(g) = \{m \in M \mid gIm\}$

Dans ce cas, nous avons aussi $\sqsubseteq = \subseteq$

Définition 21. *En considérant les deux opérateurs* $(.)^{\square}$ *avec* $A \subseteq G$ *et* $d \in (D, \sqcap)$ *tels que :*

$$d^{\square} = \{g \in G | d \sqsubseteq \delta(g)\}$$

$$A^{\square} = \bigsqcap_{g \in A} \delta(g)$$

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Ces opérateurs forment ***une connexion de Galois*** *entre* $(\mathcal{P}(G), \subseteq)$ *et* $(D, \sqsubseteq)$. *Et,* $(.)^{\square\square}$ *est* ***un opérateur de fermeture****, c'est-à-dire que si un motif d vérifie* $d^{\square\square} = d$, *alors d est un motif fermé.*

Définition 22. *Les paires* (A, d) *telles que :*

$$A \subseteq G, d \in D, A^{\square} = d \text{ et } d^{\square} = A$$

sont appelées ***les concepts*** *de la structure de motifs* $\mathcal{S} = (G, (D, \sqcap), \delta)$. *Le sous-ensemble d'objets A est alors appelé* ***l'extension*** *du concept, et la description commune d des objets est son* ***intention****.*

5.1.2 Motifs d'intervalles dans un jeu de données numériques

Nous rappelons la définition d'un jeu de données numérique avant de présenter les motifs d'intervalles.

Définition 23. ***Un jeu de données numérique*** $N = (G, M, \{\mathbf{m_i}\}_{i \in [1,|M|]})$ *est donné par :*

- *un ensemble d'objets G*
- *un ensemble d'attributs numériques M tel que chaque attribut* $m_i \in M$ *a un domaine fini* W_{m_i}.
- $\mathbf{m_i}(g) = w$ *veut dire que l'objet g prend la valeur* $w \in W_{m_i}$ *pour l'attribut* m_i.

Exemple 5.2. Nous considérons un exemple de jeu de données numérique $D_2 = (G_2, M_2)$où M_2 est un ensemble d'attributs numériques et tel que représenté dans la table 5.1. Les attributs numériques $m_1, m_2 \in M_2$ sont à valeur dans $\mathbb{R}$.

	m_1	m_2
g_1	1.2	43
g_2	7.3	21
g_3	2.3	31
g_4	4.5	28

TABLE 5.1 – D_2 : Jeu de données numériques

Dans (Kaytoue et al., 2011), les auteurs ont montré qu'une structure de motifs peut être dérivée d'un jeu de données numérique en utilisant les motifs d'intervalles. Ces motifs sont des vecteurs d'intervalles de dimension égale au nombre d'attributs numériques du jeu de données.

Définition 24. ***Un motif d'intervalles*** *d est un vecteur d'intervalles* $\langle a_i, b_i \rangle_{i \in [1,|M|]}$ *où* $a_i, b_i \in W_{m_i}$. *Les motifs d'intervalles sont ordonnés dans un demi-treillis inférieur avec l'infimum de* $c = \langle a_i, b_i \rangle$ *et* $d = \langle e_i, f_i \rangle$ *qui est donné par :*

$$c \sqcap d = \langle [min(a_i, e_i), max(b_i, f_i)] \rangle$$

La relation d'ordre *induite par cette définition est :*

$$c \sqsubseteq d \iff [e_i, f_i] \subseteq [a_i, b_i] \forall i \in [1, |M|]$$

Un objet g est dans ***l'image*** *d'un motif d'intervalles* $d = \langle a_i, b_i \rangle_{i \in [1,|M|]}$ *si* $m_i(g) \in [a_i, b_i] \forall i \in [1, |M|]$.

Exemple 5.3. Soient les exemples de motifs d'intervalles $p_1 = \langle [1, 10] \times [20, 50] \rangle$ et $p_2 = \langle [2, 5] \times [25, 35] \rangle$ pour le jeu de données numériques D_2 de l'exemple 5.2. Nous avons la relation d'ordre $p_1 \sqsubseteq p_2$ et l'image de p_2 est l'ensemble d'objets $\{g_3, g_4\}$.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Un motif d'intervalles dans un jeu de données numérique $N = (G, M, \{\mathbf{m_i}\}_{i\in[1,|M|]})$ est une conjonction de restrictions sur les valeurs des attributs numériques. Il peut être représenté par un hyperrectangle de dimension $|M|$ de $\mathbb{R}^{|M|}$ dans lequel chaque arête correspond à un intervalle du motif. $\forall p \in \mathbb{N}^*$, l'ensemble $\mathcal{D}_p$ est défini comme l'ensemble de tous les hyperrectangles de dimension p et parallèles aux axes de $\mathbb{R}^p$. Cet ensemble peut être muni de la relation d'ordre d'inclusion $\subseteq$. L'ensemble partiellement ordonné $(\mathcal{D}_p, \subseteq)$ est un treillis complet dont les plus grand et plus petit éléments sont respectivement $\mathbb{R}^p$ et $\varnothing$. De plus, la borne inférieure d'un sous-ensemble $S \subseteq \mathcal{D}_p$ est donnée par $\cap S$, et la borne supérieure est donnée par le plus petit hyperrectangle de dimension p et parallèle aux axes de $\mathbb{R}^p$ qui inclut les hyperrectangles de S.

La caractérisation géométrique des motifs d'intervalles permet de mieux les visualiser et de montrer qu'ils peuvent bien être ordonnés dans un treillis complet.

5.1.3 Langage des motifs d'intervalles

Définition 25. *Soit $p \in \mathbb{N}^*$, le langage des motifs d'intervalles est le treillis complet $(\mathcal{D}_p, \sqsubseteq)$ où l'ordre $\sqsubseteq$ est donné par l'inclusion duale $\supseteq$. Soient $p_1 = \langle [a_i, b_i] \rangle_{i\in[1:p]}$ et $p_2 = \langle [e_i, f_i] \rangle_{i\in[1:p]}$, alors :*

$$p_1 \sqsubseteq p_2 \iff [a_i, b_i] \supseteq [e_i, f_i] \iff [e_i, f_i] \subseteq [a_i, b_i], \quad \forall i \in [1:p]$$

Le langage des motifs d'intervalles $(\mathcal{D}_p, \sqsubseteq)$ est infini, nous considérons dans la suite sa restriction $(\mathcal{D}_\mathcal{G}, \sqsubseteq)$ obtenue en se restreignant aux descriptions des objets de l'ensemble G du jeu de données numérique N. Dans $(\mathcal{D}_G, \sqsubseteq)$, les descriptions et les hyperrectangles correspondants sont construits en utilisant uniquement les valeurs d'attributs qui apparaissent dans le jeu de données.

Définition 26. *L'opérateur de fermeture $(.)^{\square\square}$ des structures de motifs peut être appliqué aux motifs d'intervalles.*
Soit I un motif d'intervalles, nous définissons

$$\phi(I) = \{g \in G | \mathbf{m_i}(g) \in [a_i, b_i] \ \ \forall m_i \in M\}$$

et, pour $A \subseteq G$,

$$\psi(A) = \langle [a_i, b_i] \rangle_{i\in[1,|M|]} \quad \text{avec} \quad a_i = \min_{g\in A} \mathbf{m_i}(g) \ \text{ et } \ b_i = \max_{g\in A} \mathbf{m_i}(g)$$

Nous avons alors, $(.)^{\square\square} = \psi \circ \phi(.)$ et $(.)^{\square\square} = \phi \circ \psi(.)$ des opérateurs de fermeture pour les motifs d'intervalles et les ensembles d'objets respectivement.

Exemple 5.4. En reprenant le jeu de données numériques de l'exemple 5.2 et la structure de motifs associée $(G_2, (\mathcal{D}_{G_2}, \sqsubseteq), \delta)$, nous avons $(\{g_2, g_3\})^{\square\square} = \{g_2, g_3, g_4\}$:

$$(\{g_2, g_3\})^{\square\square} = \phi \circ \psi(\{g_2, g_3\}) = \phi(\langle [2.3, 7.3] \times [21, 31] \rangle) = \{g_2, g_3, g_4\}$$

Définition 27. *Pour les motifs d'intervalles, **une classe d'équivalence** est un ensemble de motifs d'intervalles ayant la même image. L'unique motif d'intervalles fermé d'une classe d'équivalence est le motif maximal par rapport à la relation d'ordre $\sqsubseteq$.*
***Un treillis de concepts** peut être construit avec les motifs d'intervalles fermés et leurs images.*

Les motifs d'intervalles fermés et leurs images forment des concepts qui peuvent être ordonnés dans un treillis complet muni de la relation d'ordre partielle $\leqslant$. Soient I_1 et I_2 deux

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

motifs d'intervalles fermés et $A_1, A_2 \subseteq G$ leurs images respectives. (A_1, I_1) et (A_2, I_2) sont des concepts tels que :

$$(A_1, I_1) \leqslant (A_2, I_2) \iff I_1 \sqsubseteq I_2 \iff A_2 \subseteq A_1$$

Le treillis de concepts est le langage de motifs qui sera utilisé dans la suite du chapitre. Seuls les motifs d'intervalles fermés seront considérés ce qui permet une énumération efficace de tous les sous-groupes $A \subseteq G$ de la structure de motifs d'intervalles associée au jeu de données numérique.

5.2 Traitement du langage naturel

Nous nous intéressons, dans cette partie, aux méthodes de traitement du langage naturel qui permettent de mesurer la similarité entre deux documents textuels. Nous présentons, en particulier, les méthodes qui permettent de représenter un document textuel dans un espace vectoriel. Le calcul de similarité entre les vecteurs résultants peut alors se faire avec la similarité cosinus par exemple. La méthode Latent Semantic Analysis (LSA) (Deerwester et al., 1990) fait partie des approches les plus anciennes pour représenter les mots d'un corpus de texte dans un espace vectoriel de petite dimension. La méthode est basée sur l'application de la décomposition en valeurs singulières de la matrice W qui indique les occurrences des mots dans les phrases du corpus. En notant n la taille du vocabulaire, d le nombre de phrases et k la dimension souhaitée pour l'espace vectoriel de représentation, nous avons : $W = USV^T$ où W est une matrice $d \times n$, U et V^T sont de dimension $d \times k$ et $k \times n$ respectivement, et S est une matrice carrée diagonale $k \times k$. Une fois la décomposition calculée, les vecteurs de $\mathbb{R}^k$ qui représentent les mots du corpus sont obtenus dans les colonnes de la matrice SV^T. La principale limite de cette approche est qu'elle devient très coûteuse en calcul quand la taille du corpus de texte est grande.

Les récentes avancées en traitement du langage naturel sont la conséquence du développement des méthodes d'apprentissage profond qui permettent de traiter des données de très grande taille. Ces méthodes sont aussi appelées des modèles de langage car elles se basent sur l'estimation de la probabilité d'une séquence de mots en fonction de son voisinage dans le texte. Ce sont des modèles de langage à espace continu car elles transforment les vecteurs binaires qui représentent chaque mot par sa position dans le vocabulaire en des vecteurs dans un espace euclidien continu de plus petite dimension que la taille du vocabulaire. De nombreux modèles de langage à espace continu ont démontré leur capacité à obtenir des bons résultats pour différentes autres tâches de traitement du langage naturel : traduction automatique, reconnaissance vocale, analyse des sentiments ... (Schwenk, 2007). Les représentations générées implicitement par ces modèles d'apprentissage capturent les similarités sémantiques et syntaxiques du langage. Par exemple, certaines relations entre mots apparaissent en calculant la différence entre leurs représentations vectorielles. Ainsi, dans (Mikolov et al., 2013b), les auteurs ont vérifié des égalités approximatives entre les différences de représentations vectorielles comme : $x_{car} - x_{cars} \simeq x_{apple} - x_{apples}$ où x_i est la représentation vectorielle du mot i. Ces différences de vecteurs représentent la relation entre un mot et son pluriel.

Parmi les modèles de langage à espace continu, le modèle d'apprentissage profond NNLM (Bengio et al., 2003) utilise un réseau de neurones artificiels acyclique à deux couches

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

cachées : une couche linéaire de projection et une couche non linéaire densément connectée. L'apprentissage est basé sur la prédiction d'un mot en fonction de ceux qui le précédent (jusqu'à un horizon N) en utilisant leurs coordonnées dans l'espace continu de projection. Les représentations des mots obtenues à la fin de l'apprentissage sont celles qui performent le mieux dans la tâche de prédiction. Le modèle RNNLM (Mikolov et al., 2010) utilise un réseau de neurones récurrent pour effectuer la même tâche de prédiction d'un mot en fonction de l'information contenue dans les mots qui le précédent. Les architectures de ces deux modèles de réseaux de neurones contiennent une couche cachée non linéaire densément connectée : ce type de couche contient beaucoup de paramètres à optimiser et ralentit l'apprentissage du modèle. Pour répondre à cette limite, (Mikolov et al., 2013a) ont proposé la méthode Word2vec et ont comparé différentes techniques pour l'apprentissage des représentations vectorielles des mots à partir de documents de taille encore plus grande. La technique la plus performante, parmi celles qui ont été comparées, utilise le modèle Skip-Gram. Ce modèle cherche les représentations vectorielles des mots qui permettent d'obtenir la meilleure prédiction de leurs voisinages tels qu'ils ont été observés dans le document de texte fourni en entrée.

Toutes les méthodes présentées jusque-là font le changement de représentation pour les mots des documents. Avec ces méthodes, la représentation dans un espace continu d'une phrase ou d'un paragraphe est obtenue en agrégeant les représentations des mots qui les constituent. Mais, plus récemment, de nouveaux modèles de réseaux de neurones ont été développés : ils permettent de représenter directement une séquence de mots dans un espace euclidien. Ces modèles se basent sur un nouveau type de réseau de neurones appelé "Transformer" qui utilise une architecture encodeur-décodeur avec un mécanisme d'attention (self-attention) (Vaswani et al., 2017).

L'apprentissage de tels modèles de représentation de textes dans des espaces vectoriels est très coûteux en ressources de calcul. Néanmoins, de plus en plus de modèles pré-entraînés sont disponibles et permettent de limiter le coût de calcul. En effet, ces modèles ne commencent pas l'apprentissage des paramètres avec des valeurs initiales aléatoires mais avec les valeurs obtenues après une étape préalable d'apprentissage sur un très grand corpus générique. Le deuxième apprentissage avec les données spécifiques que nous cherchons à analyser est alors plus rapide car les paramètres sont plus proches de leur valeur optimale. Le modèle d'apprentissage profond pour le traitement du langage naturel BERT (Bidirectional Encoder Representation from Transformers) est un exemple de "Transformer" pré-entraîné. Les données qui sont utilisées pour cette étape sont le corpus de texte BooksCorpus contenant plus de 800 millions de mots et un sous-ensembles des pages Wikipedia en anglais. BERT a la particularité d'être bidirectionnel : le traitement d'un mot par le réseau de neurones dépend d'un sous-ensemble de mots parmi ceux qui le précédent et ceux qui le suivent.

5.3 Motifs numériques pour les multigraphes de co-appartenance

5.3.1 Construction du jeu de données numériques

Pour construire le jeu de données numériques contenant l'information structurelle et sémantique du multigraphe de co-appartenance $\mathcal{G} = (\mathcal{V}, E, \mathcal{L}, \mathbf{k})$, nous définissons la mesure de similarité entre couches *Sim* puis nous en déduisons une fonction d'appartenance

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

généralisée d'un nœud à une couche dont les valeurs $\forall L \in \mathcal{L}$ et $\forall V \in \mathcal{V}$ forment la matrice de dimension $|\mathcal{V}| \times |\mathcal{L}|$ qui correspond au jeu de données numérique.

Définition 28. ***Similarité sémantique entre couches*** *: Soit Sim une mesure de similarité entre deux couches :* $\forall L_i, L_j \in \mathcal{L}, Sim(\mathbf{k}(L_i), \mathbf{k}(L_j)) \in [0,1]$ *et* $Sim(\mathbf{k}(L_i), \mathbf{k}(L_i)) = 1$

Nous détaillons ci-dessous l'approche suivie pour calculer les similarités entre couches. Ce processus est basé sur une fonction d'appartenance généralisée des nœuds aux liens définie comme suit :

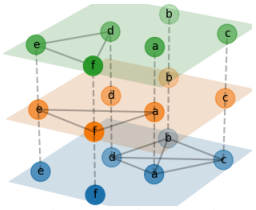
Définition 29. ***fonction d'appartenance généralisée*** *Étant donné un graphe de co-appartenance* $\mathcal{G}$*, la fonction d'appartenance généralisée* $M(v, L_i)$ *d'un nœud* $v \in \mathcal{V}$ *à une couche* L_i *est définie par* $M(v,i) = 1$*, if* $v \in \mathbf{v}(L_i)$*, et* $M(v,i) = \max_{l' | v \in \mathbf{v}(l')} Sim(\mathbf{k}(L_i), \mathbf{k}(l'))$*, autrement.*

Exemple 5.5. Considérons le multigraphe de co-appartenance de 3 couches $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \{L_1, L_2, L_3\}, \mathbf{k})$ évoqué dans l'exemple 3.1 du chapitre 3 pour lequel nous montrons ici comment en dériver un jeu de données numérique. Supposons les valeurs de similarité sémantique suivantes :

$$Sim(\mathbf{k}(L_1), \mathbf{k}(L_3)) = 0.9$$
$$Sim(\mathbf{k}(L_2), \mathbf{k}(L_3)) = 0.7$$
$$Sim(\mathbf{v}(L_1), \mathbf{v}(L_2)) = 0.2$$

Soit le nœud $v \in \mathcal{V}$ tel que $v \in \mathbf{v}(L_1)$ et $v \notin \mathbf{V}(L_3)$, nous avons alors :

$$M(v,3) = max(Sim(\mathbf{k}(L_1), \mathbf{k}(L_3)), Sim(\mathbf{k}(L_2), \mathbf{k}(L_3))) = 0.9$$



Multigraphe de co-appartenance

	Layer 1	Layer 2	Layer 3
a	1.0	1.0	0.9
b	1.0	0.0	0.9
c	1.0	0.0	0.9
d	1.0	0.7	1.0
e	0.9	1.0	1.0
f	0.9	1.0	1.0

Jeu de données numérique

Nœuds	Couches et intervalles
a b c d	(Layer 1)[1.0,1.0] (Layer 3)[0.9,1.0]
d e f	(Layer 1)[0.9,1.0] (Layer 2)[0.7,1.0] (Layer 3)[1.0,1.0]

Communautés

FIGURE 5.1 – Exemple de multigraphe de co-appartenance de 3 couches, le jeu de données numérique correspondant, et deux exemples de communautés ($Sim(\mathbf{v}(L_1), \mathbf{v}(L_2)) = 0.2$, $Sim(\mathbf{v}(L_1), \mathbf{v}(L_3)) = 0.9$ et $Sim(\mathbf{v}(L_2), \mathbf{v}(L_3)) = 0.7$).

La Figure 5.1 fournit le jeu de données numérique associé au multigraphe de co-appartenance et deux exemples de communautés qui correspondent aux images de deux motifs d'intervalles. Le jeu de données numérique synthétise les similarités structurelle et sémantique entre les couches du multigraphe de co-appartenance.

Motifs d'intervalles et multigraphe de co-appartenance

Maintenant, pour former des communautés dans un graphe de co-appartenance, nous proposons de regrouper les nœuds qui sont connectés dans les mêmes couches ou dans des

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

couches proches d'un point de vue sémantique. Pour cela, nous utilisons les structures de motifs et plus particulièrement les motifs d'intervalles.

Nous proposons d'identifier les communautés dans les graphes de co-appartenance en utilisant la fonction d'appartenance généralisée. L'objectif est d'identifier des ensembles de nœuds partageant les mêmes grandes valeurs d'appartenance à un ensemble de listes. Pour cela, nous représentons, en utilisant la fonction d'appartenance généralisée, le graphe de co-appartenance comme un jeu de données numérique. Les communautés du graphe correspondent alors aux motifs d'intervalles fermés et fréquents du jeu de données (Kaytoue et al., 2017). L'extension des motifs obtenus est un ensemble de nœuds, et l'intention est un ensemble de $|\mathcal{L}|$ intervalles tel que tous les nœuds de l'extension ont leur valeur d'appartenance généralisée pour une couche $l \in \mathcal{L}$ dans l'intervalle correspondant. Les communautés de nœuds sont alors caractérisées par les intervalles contenant de grandes valeurs et qui correspondent aux couches partagées par les nœuds de la communauté.

Définition 30. ***Jeu de données associé au multigraphe de co-appartenance*** *Le jeu de données numérique associé à un multigraphe de co-appartenance croise les nœuds $\mathcal{V}$ et les couches $\mathcal{L}$. Il comporte les entrées $M(v, l)$ avec $v \in \mathcal{V}$ et $l \in \mathcal{L}$ (définition 29). Un motif d'intervalles I est un vecteur de d intervalles $\langle [a_l, b_l] \rangle_{l \in \mathcal{L}}$. Un nœud $v \in \mathcal{V}$ appartient au support de I ssi $\forall l \in \mathcal{L}, M(v, l) \in [a_l, b_l]$.*

Il est possible que plusieurs motifs d'intervalles partagent le même support. En utilisant les opérateurs de fermeture $\phi \circ \psi$ et $\psi \circ \phi$, nous restreignons les motifs d'intervalles aux motifs d'intervalles fermés sans perte d'information.

Les communautés de nœuds correspondent ici aux ensembles de nœuds partageant des grandes valeurs d'appartenance à un même sous-ensemble de couches. Par conséquent, nous ne considérons que les motifs d'intervalles tels que $b_l = 1, \forall l \in \mathcal{L}$. Les intervalles informatifs sont ceux pour lesquels $a_i \neq 0$. Nous évaluons alors la qualité d'un motif d'intervalles fermé comme la somme des valeurs d'appartenances des nœuds de son support aux intervalles qui vérifient $a_i \neq 0$.

5.3.2 Calcul de similarités entre attributs textuels avec le modèle BERT

Pour tirer profit de l'information textuelle associée à chaque couche du multigraphe, nous proposons d'utiliser les récentes techniques en apprentissage machine et plus particulièrement en traitement du langage naturel pour représenter les attributs textuels dans un espace vectoriel euclidien. Des techniques non supervisées comme Word2Vec (Mikolov et al., 2013a) ont été proposées pour accomplir cette tâche mais elle restent limitées car elles ne prennent pas en compte le voisinage des mots dans leur contexte pour la définition de leur représentation. Des méthodes plus récentes d'apprentissage profond permettent de tenir compte de la position des mots dans les phrases de sorte qu'un mot polysémique est proche des mots qui ont un sens similaire dans le même contexte. L'apprentissage de tels modèles de représentation de texte dans des espaces vectoriels est très coûteux en ressources de calcul. Néanmoins, de plus en plus de modèles pré-entraînés sont disponibles et permettent de limiter le coût de calcul. En effet, ces modèles ne commencent pas l'apprentissage des paramètres avec des valeurs initiales aléatoires mais avec les valeurs obtenues après une première étape d'apprentissage sur un très grand corpus générique. Le deuxième apprentissage avec les données spécifiques au graphe de co-appartenance est alors plus rapide car les paramètres sont plus proches de leur valeur optimale.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Nous utilisons le modèle d'apprentissage profond pour le traitement du langage naturel BERT (Bidirectional Encoder Representation from Transformers), pré-entraîné sur le corpus de texte BooksCorpus contenant plus de 800 millions de mots et sur un sous-ensembles des pages Wikipedia en anglais. Le modèle est fine-tuné avec les attributs textuels des couches du multigraphe. Ceci rend le modèle plus spécifique à nos données et augmente la dispersion des vecteurs générés à partir des attributs textuels. Les similarités entre les vecteurs obtenus sont calculés à l'aide de la distance cosinus.

5.3.3 Extraction des top-k motifs d'intervalles

Définition 31. ***Top-k motifs d'intervalles*** *Une communauté issue d'un motif d'intervalles est composée de nœuds possédant des grandes valeurs d'appartenance sur des intervalles restreints à gauche. Nous mesurons la qualité d'une communauté avec la fonction Q définie par :*

$$Q(I) = \sum_{l \in \mathcal{L},\, a_l > 0} \; \sum_{v \in \phi(I)} M(v, l)$$

Les top-k motifs d'intervalles sont les k motifs d'intervalles fermés avec les plus grandes valeurs pour la fonction de qualité Q.

Aussi, nous nous attendons à ce que les communautés soient différentes l'une de l'autre. Nous mesurons la similarité entre deux communautés en calculant la proportion de nœuds communs à l'aide de l'indice de Jaccard :

$$Jaccard(I, J) = \frac{\phi(I) \cap \phi(J)}{\phi(I) \cup \phi(J)}$$

Deux communautés I et J sont dites diversifiées si la proportion de leurs nœuds communs est inférieur au paramètre δ : $Jaccard(I, J) \leqslant \delta$. Nous cherchons alors l'ensemble de communautés contenant au moins σ nœuds et tel que toutes les paires de communautés vérifient la condition de diversification.

Définition 32. ***Top-k motifs d'intervalles diversifiés*** *Soit S un ensemble de motifs d'intervalles fermés. Etant donnés les trois paramètres σ, δ et k, nous pouvons calculer l'ensemble $\mathcal{M}_{top}$ des top-k motifs d'intervalles diversifiés tel que :*

(1) $\mathcal{M}_{top}$ *est de taille k avec* $\mathcal{M}_{top} \subseteq S$

(2) $\forall I \in \mathcal{M}_{top}, |\phi(I)| \geqslant \sigma$

(3) $\forall I, J \in \mathcal{M}_{top}, Jaccard(I, J) < \delta$

(4) $\forall I \in \mathcal{M}_{top}$, *si* $\exists J \in S$ *tel que* $Jaccard(I, J) \geqslant \delta$, *alors* $Q(I) \geqslant Q(J)$

Une implémentation efficace pour l'extraction des motifs d'intervalles fermés est proposée dans (Kaytoue et al., 2011). L'algorithme évite de générer un même motif d'intervalles à plusieurs reprises en utilisant un test de canonicité avec l'ordre $<$ sur l'ensemble des couches $\mathcal{L}$: si un motif I a été généré par le changement de l'attribut l, il vérifie le test de canonicité ssi I et $\psi(\phi(I))$ ne diffèrent dans aucun attribut h tel que $h < l$. Le fait que $\psi(\phi(I))$ diffère de I sur au moins un attribut $h \leqslant index$ est noté $Closed \prec_{index} I$ dans l'algorithme 8, et entraîne l'arrêt de la récursion (ligne 2-3). La récursion est aussi arrêtée quand le support du motif

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

est inférieur à σ, dans ce cas aucun motif généré par les récursions suivantes ne peut avoir un support plus grand que le seuil σ.

Nous étendons l'algorithme en ajoutant une autre condition qui peut arrêter la récursion : le fait que la borne supérieure de la mesure de qualité Q, notée UBQ, est inférieure à min_Q où min_Q est un seuil dynamique sur la mesure de qualité mis à jour au cours de l'exécution de l'algorithme. Définissons cette borne supérieure sur Q. Pour cela, nous considérons, pendant le processus d'énumération de l'algorithme 8, l'étape durant laquelle un motif d'intervalles I avec $[a_l, b_l]$ son l^{eme} intervalle est spécialisé en un motif d'intervalles J avec le l^{eme} intervalle $[c_l, d_l]$ tels que $a_l < c_l$ et $b_l > d_l$. Pour de tels J, nous avons :

$$\begin{aligned} Q(J) &\leqslant \sum_{i=1,\, a_i>0}^{index} \sum_{v\in\phi(J)} M(v,i) + \sum_{i=index+1}^{d} \sum_{v\in\phi(J)} M(v,i) \\ &\leqslant \sum_{i=1,\, a_i>0}^{index} \sum_{v\in\phi(I)} M(v,i) + \sum_{i=index+1}^{d} \sum_{v\in\phi(I)} M(v,i) \\ &= UBQ(I) \end{aligned}$$

Algorithm 8: FCIP(P, σ, k, δ, index, $\min_Q$)

Input: P est un motif d'intervalle généré à l'étape précédente avec un changement minimal sur l'attribut numéroté *index*. σ, k et δ sont des seuils fixes, et $\min_Q$ est un seuil dynamique sur Q.

Output: IP est l'ensemble des top-k motifs d'intervalles fermés fréquents diversifiés.

```
1  Closed ← ψ(φ(P))
2  if (|φ(P)| < σ) or (Closed ≺_index P) or (UBQ(P) < min_Q) then
3  |  return
4  TopKDiv(IP, Closed, k, δ, min_Q)
5  for i = index to |D| do
6  |  [a, b] ← Closed[i]
7  |  if a = b then
8  |  |  Continue
9  |  c ← nextValue(i,a)
10 |  PatL ← [1 ... i − 1][c, b][i + 1 ... |D|]
11 |  FCIP(PatL, σ, k, δ, i, min_Q)
```

En effet, pour les intervalles qui n'ont pas encore été énumérés, nous pouvons sommer les valeurs de la fonction d'appartenance généralisée. De plus, comme $\phi(J) \subseteq \phi(I)$, nous pouvons aussi définir une borne supérieure pour $Q(J)$ en prenant la somme sur les nœuds du support du motif d'intervalles I.

À la ligne 4 de l'algorithme 8, *Closed* est un motif candidat et la fonction TopKDiv assure que l'ensemble de motifs retourné contienne les top-k motifs diversifiés (algorithme 9). Dans les lignes 5 à 11, l'algorithme génère récursivement les prochains motifs candidats en considérant l'intervalle suivant (ligne 6). Le nouvel intervalle est une spécialisation du précédent obtenu en augmentant la valeur de la borne à gauche de l'intervalle a (lignes 9 et 10). MIMETIC est appelé récursivement à la ligne 11.

La première exécution de MIMETIC se fait avec les paramètres MIMETIC(P, min_Q, k, δ, 1, 0) et avec $P = \bigtimes_i [0, 1]$.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Algorithm 9: TopKDiv(IP, *Closed*, k, δ, $\min_Q$)

Input: IP ordonné selon Q d'au plus k motifs d'intervalles fermés fréquents diversifiés.
Output: IP la liste triée mise à jour avec la valeur actuelle de $\min_Q$.

```
1  similarPatterns ← ∅
2  i ← 0
3  isBetter ← true
4  while i < |IP| and isBetter do
5      J ← IP[i]
6      if Jaccard(J, Closed) ⩾ δ then
7          insert(J, similarPatterns)
8          if Q(J) > Q(Closed) then
9              isBetter ← false
10     i ← i + 1
11 if (|similarPatterns| = 0) then
12     if (Q(Closed) ⩾ δ) then
13         if (|IP| = k) then
14             remove(IP[0]), IP)
15             insert(Closed, IP)
16             min_Q ← Q(IP[0])
17         else
18             insert(Closed, IP)
19 else
20     if isBetter then
21         forall J ∈ similarPatterns do
22             remove(J, IP)
23         insert(Closed, IP)
24         if (|IP| = k) then
25             min_Q ← Q(IP[0])
26
```

La fonction TopKDiv (algorithme 9) retourne l'ensemble des top-k motifs d'intervalles diversifiés. Il parcourt l'ensemble ordonné des k motifs identifiés auparavant (lignes 4 à 10) et enregistre ceux qui sont similaires au motif *Closed* dans la variable *similarPatterns*. Si un des motifs similaires a une valeur de qualité Q supérieure à celle de *Closed*, le parcours s'arrête car le motif *Closed* ne peut être inséré dans l'ensemble des top-k motifs diversifiés (lignes 8 et 9). Si *similarPatterns* est vide, alors il y a deux possibilités : (1) si la taille de l'ensemble des top-k motifs diversifiés est égale à k, alors *Closed* remplace le motif de plus petite mesure de qualité et le seuil de qualité min_Q est mis à jour (lignes 13 à 16), (2) sinon, le motif est simplement rajouté à l'ensemble (ligne 18). Autrement, si *similarPatterns* contient au moins un motif et que la qualité de *Closed* est supérieure à celle de tous les motifs de *similarPatterns*, alors tous les motifs similaires de *similarPatterns* sont retirés et *Closed* est rajouté à l'ensemble des top-k motifs diversifiés. A noter que min_Q n'est mis à jour que dans les cas où l'ensemble des top-k motifs contient déjà k motifs.

Le jeu de données considéré ici est construit à partir du multigraphe de co-appartenance. Il contient les valeurs d'appartenance généralisée des nœuds aux couches. Ces valeurs sont obtenues en prenant en compte les similarités structurelle et sémantique entre les couches du multigraphe. Les top-k motifs d'intervalles diversifiés de ce jeu de données ont pour image des ensembles de nœuds qui constituent des communautés homogènes. L'algorithme MIMETIC permet d'obtenir les k communautés les plus homogènes présentes dans des parties différentes du multigraphe de co-appartenance.

5.4 Résultats et comparaisons

Nous évaluons la performance de MIMETIC selon de multiples aspects en répondant aux questions :

(1) Comment s'exécute MIMETIC sur les graphes de co-appartenance de grande taille issus de situations réelles ?

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

(2) Quel est l'impact des paramètres sur la qualité des motifs ?

(3) Quelles sont les caractéristiques des communautés

(4) Comment MIMETIC se compare aux méthodes de détection de communautés de l'état de l'art sur un jeu de données possédant une réalité terrain pour le partitionnement ?

(5) Comment la sémantique améliore les résultats de la détection de communautés ?

(6) Est-ce que MIMETIC retourne des communautés facilement interprétables ?

5.4.1 Description des jeux de données et des méthodes compétitives

Pour l'évaluation, nous considérons trois exemples de graphes de co-appartenance issus de situations réelles et possédant des attributs textuels associés à leurs couches. Pour un des exemples considéré, la réalité terrain des communautés est connue :

— Cit-HepTh [1] est un graphe de co-citations d'articles de physique théorique dans lequel les articles correspondent aux couches du graphe et les citations aux nœuds. Les résumés des articles sont alors les attributs textuels des couches du multigraphe.

— IMDB [2] est une base de données de films que nous utilisons pour construire un graphe de co-appartenance des acteurs aux films. Les nœuds du graphe sont les acteurs et les couches correspondent aux films et possèdent un attribut textuel issu du synopsis du film.

— Wiki-Topcats [3] est un graphe de co-citation construit à partir de pages Wikipedia. Les couches représentent des pages Wikipedia et ont un attribut textuel qui est le titre de la page. Les nœuds d'une même couche sont reliés par une relation de co-référence. Ce jeu de données comprend aussi des catégories dans lesquels les pages sont classés et qui sont considérées comme la réalité terrain des communautés. La disponibilité de cette réalité terrain nous permet d'évaluer la performance des méthodes de détection de communautés en quantifiant le degré de d'accord entre les communautés obtenues et la réalité terrain.

La table ci-dessous liste les exemples de graphes utilisés et leurs propriétés :

Dataset	# nodes	# edges	# layers	average layers' density	average density with $M(v,i) > 0$
CIT-HEPTH[1]	13921	2336992	14932	1.61×10^{-6}	2.95×10^{-6}
WIKI-TOPCATS[3]	10000	1561063	9891	3.16×10^{-6}	6.48×10^{-5}
IMDB[2]	18125	65033	9975	3.96×10^{-8}	2.76×10^{-3}

Nous considérons cinq méthodes compétitives auxquelles nous nous comparons : MiMAG (Boden et al., 2017), MUCHA (Mucha et al., 2010), EMCD (Tagarelli et al., 2017), MNE (Zhang et al., 2018) et Infomap (Edler et al., 2017) comme discuté dans la section 2.3 et suivant la classification dans (Hanteer et al., 2019).

1. https://snap.stanford.edu/data/cit-HepTh.html
2. https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset
3. https://snap.stanford.edu/data/wiki-topcats.html

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

5.4.2 Évaluation de l'algorithme

Tout d'abord, nous considérons le comportement de MIMETIC sur les jeux de données. La figure 5.2 représente le temps d'exécution de l'algorithme, le nombre de motifs énumérés et le nombre de motifs élagués avec la condition sur la borne supérieure de la mesure de qualité *UBQ* en fonction de σ. Nous observons que MIMETIC réussit à extraire des communautés pour des valeurs de seuils très petites sur le nombre de nœuds par communauté (entre 0.5% et 5% en fonction du jeu de données). Nous remarquons aussi que la condition sur la borne supérieure de la qualité Q permet d'élaguer un grand nombre de motifs, spécialement quand σ est petit. Enfin, le temps d'exécution de l'algorithme augmente quand le seuil sur le support σ diminue car le nombre des très petites communautés est grand.

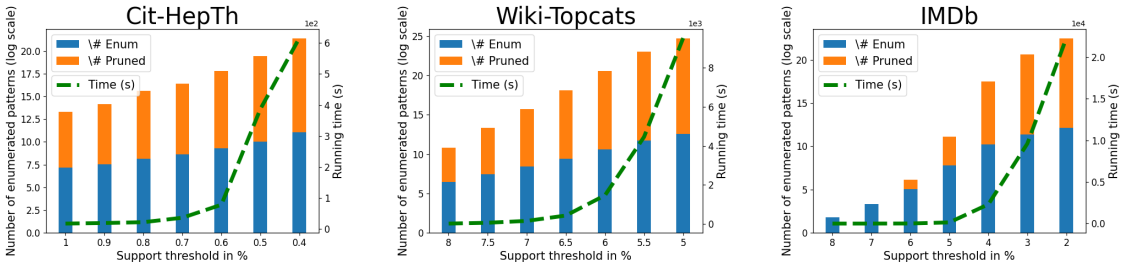


FIGURE 5.2 – Nombre de motifs énumérés et élagués et temps d'exécution ($\delta = 1$ et $k = 15$)

Effets des paramètres

La figure 5.3 montre le comportement de MIMETIC en fonction des paramètres δ et k. Rappelons que la contrainte sur la diversité requiert que la proportion de nœuds communs entre deux communautés soit plus petit que δ. Alors, quand δ est petit, le nombre de communautés obtenues peut être plus petit que k. De plus, cette contrainte a une influence sur la qualité des communautés et, ici aussi, des petites valeurs pour δ induisent de plus petites valeurs pour Q. D'après les graphiques, il apparaît que $\delta = 0.8$ est un bon compromis entre qualité et diversité des communautés. Concernant l'évolution de la qualité des communautés en fonction de la taille de l'ensemble des top-k motifs d'intervalles, nous observons que la moyenne des mesures de qualité diminue quand k augmente. Nous observons aussi que le nombre de nœuds couverts par les communautés augmente avec l'augmentation de k.

Description des communautés

La figure 5.4 montre les communautés obtenues pour chaque exemple de graphe et pour différentes valeurs de k. Le nombre moyen de nœuds par communauté est représenté dans la première ligne de la figure. Nous observons que cette moyenne est au-dessus du seuil σ même avec de grandes valeurs de k. Le nombre moyen de couches par communauté est montré dans la deuxième ligne et nous remarquons qu'il diminue quand k augmente. Les valeurs sont petites et proches de 1, ce qui indique que peu de couches sont impliquées dans chaque communauté. Cela est dû à la fonction d'appartenance généralisée. En effet, dans les cas où un nœud v n'appartient pas à la liste l, la fonction $M(v, l)$ ne prend des grandes valeurs qu'à condition que la liste l soit fortement sémantiquement similaire à au moins une

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

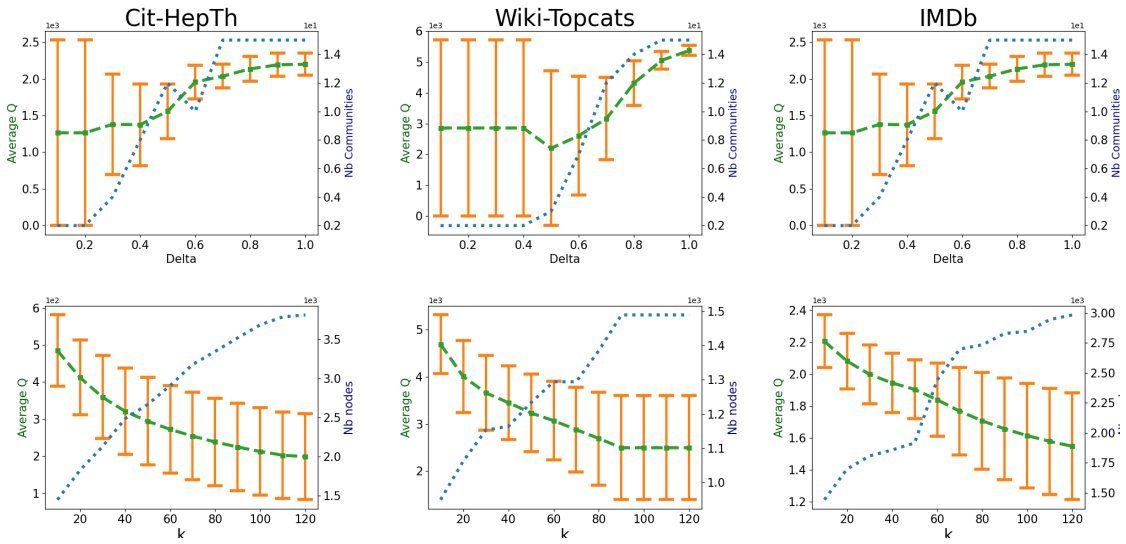


FIGURE 5.3 – Moyenne et écart-type de la mesure de qualité Q et le nombre de communautés en fonction de δ (en haut) et le nombre de nœuds couverts par l'ensemble des communautés en fonction de k (en bas) pour CIT-HEPTH ($\sigma = 100$), WIKI-TOPCATS ($\sigma = 600$) et IMDB ($\sigma = 600$) (par défaut $\delta = 0.8$, $k = 15$).

des listes auxquelles appartient le nœud v. Autrement, nous rassemblerions des couches avec des registres sémantiques très éloignés qui ne correspondent pas à la même réalité. La troisième ligne contient la moyenne et l'écart-type de l'indice de Jaccard calculé entre paires de communautés. Les communautés du graphe CitHepth sont très différentes, avec une similarité moyenne autour de 0.1. Pour les deux autres exemples de graphe, les communautés sont plus similaires mais les valeurs de similarité diminuent quand k augmente.

Comparaison aux compétiteurs

Pour estimer la capacité de MIMETIC à identifier des communautés pertinentes, nous considérons l'exemple de graphe WikiTopcats pour lequel une réalité-terrain sur les communautés est connue. Nous comparons alors l'ensemble de communautés M trouvées par notre algorithme ou par les méthodes compétitives à l'ensemble des communautés réelles M^* avec la formule :

$$\frac{1}{2|M^\star|}\sum_{m_i\in M^\star}\max_{m_j\in M}\Delta(m_i,m_j)+\frac{1}{2|M^\star|}\sum_{m_j\in M}\max_{m_i\in M^\star}\Delta(m_i,m_j).$$

Chaque communauté d'un ensemble est appariée à sa communauté la plus similaire dans l'autre ensemble, avec Δ une mesure de similarité entre deux communautés de nœuds. Nous considérons deux métriques standards : le F1-score et l'indice de Jaccard. Nous avons alors, pour chaque méthode, un score entre 0 et 1 : 0 correspondant à un résultat complètement différent de la réalité terrain et 1 à une parfaite restitution des communautés réelles par la méthode. Comme l'algorithme MIMAG renvoie une erreur de mémoire quand le nombre de couches dans le multigraphe dépasse 2500, nous réduisons le jeu de données WikiTopcats

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

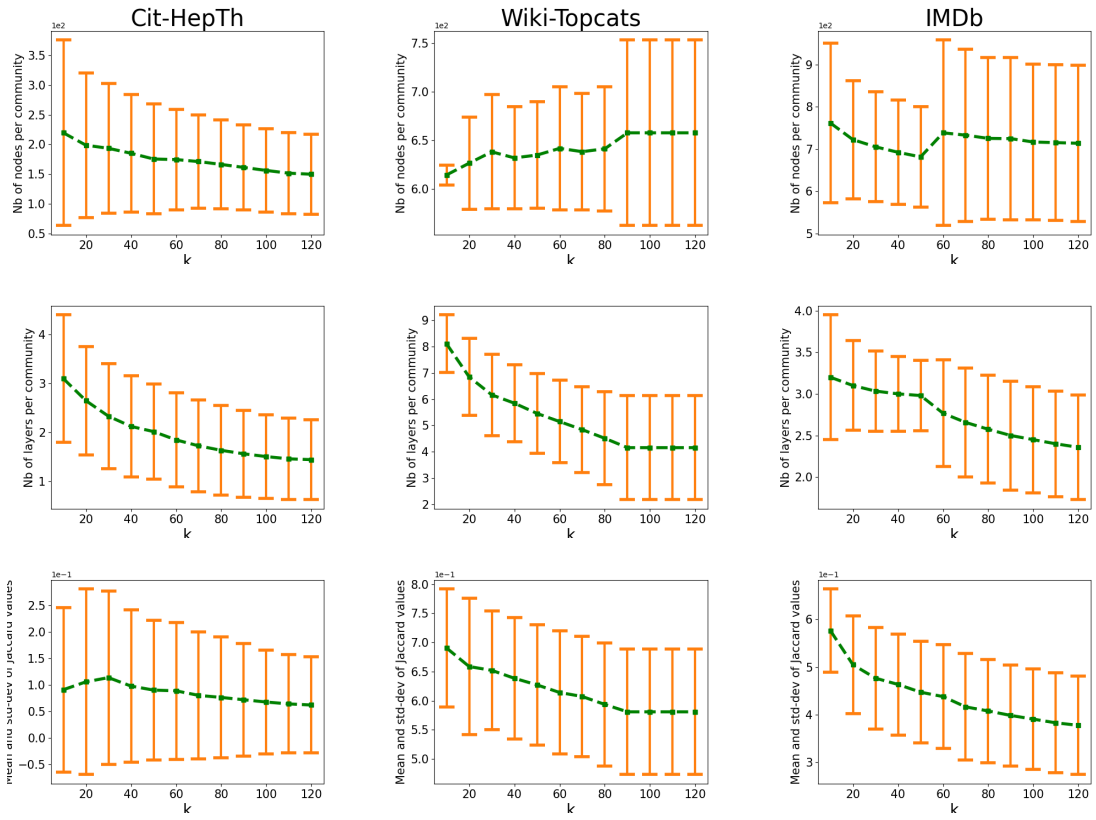


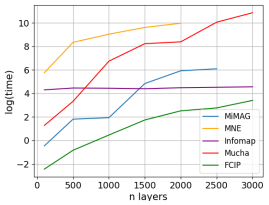
FIGURE 5.4 – Nombre de nœuds (en haut) et de couches (au milieu) par communauté. Moyenne et écart-type des valeurs des similarités de Jaccard entre paires de communautés (en bas) en fonction de k pour CIT-HEPTH ($\sigma = 100$), WIKI-TOPCATS et IMDB ($\sigma = 600, \delta = 0.8$).

et nous ne considérons que les pages appartenant à un ensemble défini des 50 catégories contenant le plus de pages. Pour exécuter MIMETIC, nous fixons $k = 50$ et nous choisissons $\sigma = 10$ pour obtenir 50 communautés avec différentes valeurs pour δ entre 0.2 et 0.8. Les résultats sont présentés à figure 5.5 à gauche. Nous observons que MIMETIC a de bonnes valeurs pour les deux métriques considérées et réussit à identifier les 50 communautés. L'algorithme MIMAG ne retourne que 4 communautés ne recouvrant que 2% des nœuds du graphe. L'algorithme Mucha produit un ensemble de 371 communautés : une contenant 1192 nœuds, 7 avec environ 40 nœuds et le reste avec très peu de nœuds par communauté. Les communautés Mucha ne sont pas homogènes en terme de catégories. Les résultats de la méthode Infomap sont aussi moins biens ceux de MIMETIC. L'algorithme MNE permet de représenter les nœuds du multigraphe dans un espace vectoriel en préservant la topologie du graphe. Il nécessite un post-traitement avec un algorithme de clustering des valeurs numériques pour obtenir les communautés de nœuds. Nous utilisons, ici, l'algorithme k-means avec le paramètre k fixé à 50. Les résultats obtenus sont là aussi moins bons que ceux de la méthode MIMETIC. Enfin, nous ne signalons pas les résultats de l'algorithme EMCD car celui-ci ne fonctionne pas quand le nombre de couches du mutigraphe est plus grand que 30 et retourne des communautés contenant un seul nœud. Concernant les temps d'exécution (Figure 5.5 au milieu), toutes les méthodes compétitives ont un temps d'exécution plus long

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

que MIMETIC.

Method		Jaccard	F1 score
FCIP	$\delta = 0.8$	0.283	0.381
	$\delta = 0.6$	0.307	0.422
	$\delta = 0.4$	0.311	0.431
	$\delta = 0.2$	0.313	0.426
MUCHA		0.137	0.187
MIMAG		0.071	0.124
INFOMAP		0.197	0.285
MNE		0.036	0.070



	FCIP-Bool		FCIP/FCIP-Bool	
δ	Jaccard	F1 score	Jaccard	F1 score
0.8	0.240	0.332	+17.9%	+14.7%
0.6	0.224	0.315	+37.0%	+33.9%
0.4	0.261	0.364	+19.1%	+18.4%
0.2	0.265	0.374	+18.1%	+13.9%

FIGURE 5.5 – WIKI-TOPCATS : Adéquation à la réalité-terrain (à gauche) ; Temps d'exécution en $\log(s)$ par rapport à # couches ($\sigma = 10, k = 50$ and $\delta = 0.4$) (au milieu). Avantage de FCIP sur FCIP-Bool (à droite).

Exemples IMDb

Nous avons exécuté MIMETIC sur le graphe de co-participation aux films IMDB. Nous avons restreint le jeu de données à un sous-ensemble de 2000 films comportant 6847 acteurs. Quand MIMAG et Mucha retournent des petites communautés avec au maximum 2 couches (films) par communauté, MIMETIC ($\sigma = 15$ et $\delta = 0.15$) trouve de plus grandes communautés recouvrant entre 3 et 5 couches. Les films correspondants aux couches d'une communauté n'ont pas que des acteurs en commun mais ils partagent aussi la même catégorie de films et ont des synopsis qui se ressemblent. Par exemple, nous trouvons une communauté de 12 acteurs et 3 films qui appartiennent tous à la catégorie 'science-fiction' et ont des mots communs dans leurs synopsis comme 'alien', 'spaceship','astronauts' ou encore 'planet'. Une autre communauté correspond à la catégorie 'drame' avec les mots-clés communs 'husband', 'affair' et 'love'.

5.4.3 La sémantique améliore-t-elle la qualité des communautés ?

Pour évaluer l'influence de la sémantique sur la qualité des communautés obtenues, nous modifions la fonction d'appartenance généralisée de la définition 29 pour que $M(v, l) = 0$ si $v \notin \mathbf{v}(l)$. Nous appelons cette version de l'algorithme MIMETIC-Bool. A noter que la méthode est alors équivalente à l'algorithme ABACUS (Berlingerio et al., 2013). Nous choisissons $\sigma = 5$ et $k = 50$ pour obtenir 50 communautés et nous faisons varier δ entre 0.2 et 0.8. La figure 5 à droite montre les résultats et nous remarquons que la prise en compte de la sémantique améliore la qualité des communautés avec un pourcentage entre 14% et 37%.

5.5 Conclusion

Nous avons introduit dans ce chapitre une méthode d'extraction de motifs numériques pour la détection de communautés dans les multigraphes de co-appartenance. Pour cela, nous avons défini une fonction d'appartenance généralisée pour estimer la similarité sémantique entre nœuds à travers les couches. Nous avons basé le calcul de ces similarités sur les représentations vectorielles des attributs textuels obtenues avec le modèle de traitement du langage naturel BERT. Nous avons proposé l'algorithme MIMETIC qui permet d'extraire

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

l'ensemble des top-k motifs d'intervalles fermés et diversifiés depuis le jeu de données numérique contenant les valeurs d'appartenance généralisée des nœuds aux couches. Nous avons comparé les résultats de notre approche aux résultats de plusieurs méthodes compétitives de l'état de l'art en utilisant l'exemple de graphe issu des co-citations des pages Wikipedia pour lequel la réalité terrain du partitionnement est connue. Nous avons trouvé que les communautés de notre approche ont une meilleure correspondance aux communautés réelles et nécessitent moins ressources de calcul. Nous avons aussi montré que la prise en compte de la similarité améliore la qualité des communautés obtenues par l'algorithme Mimetic.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Chapitre 6

Conclusion et perspectives

Dans cette thèse, nous nous sommes intéressés à la détection de communautés dans les multigraphes de co-appartenance. Ces graphes qui représentent des relations de co-appartenance d'un ensemble d'entités à un ensemble de groupes sont courants en pratique. Au cours des précédents chapitres, nous nous sommes principalement basés sur deux exemples de relations de co-appartenance : celle des utilisateurs Twitter aux listes du réseau social et la relation de co-citation des pages Wikipedia. Nous avons présenté des procédures pour construire un multigraphe à partir des données de co-appartenance et en extraire des sous-graphes qui correspondent à des domaines d'intérêt. Et, nous avons proposé des algorithmes d'extraction de motifs pour détecter les communautés dans les sous-graphes d'intérêt.

Nous avons vu que les couches dans un multigraphe de co-appartenance forment des cliques dans lesquelles tous les nœuds sont reliés. Elles possèdent des descriptions textuelles qui constituent une source d'information complémentaire pour effectuer la détection de communautés. En plus de ces caractéristiques, nous avons souligné les principales propriétés qui différencient les multigraphes de co-appartenance des multigraphes classiques étudiés dans la littérature : ils possèdent un très grand nombre de nœuds et de couches et ont un ratio de nombre de couches par nœud très supérieur à celui des multigraphes classiques.

En prenant en compte les propriétés des multigraphes de co-appartenance, nous avons proposé une méthodologie en deux étapes pour identifier leurs communautés. La première étape permet de délimiter un sous-graphe, qui correspond à un domaine d'intérêt pour l'analyste, à partir d'un ensemble de nœuds graines et cela en appliquant une des deux procédures du chapitre 3 : la procédure *Double couronnes* qui utilise un score d'homogénéité pour ne garder que des couches similaires aux nœuds graines et la procédure *Marche aléatoire* dans laquelle des marches aléatoires, dont le point de départ est un des nœuds graines, sont simulées pour déterminer les nœuds qui appartiennent au sous-graphe d'intérêt. Cette première étape permet d'obtenir un sous-graphe d'intérêt de plus petite taille que le graphe originel ce qui permet la détection de ses communautés à un moindre coût de calcul. La deuxième étape est celle de la détection effective des communautés dans le sous-graphe d'intérêt en utilisant des méthodes de fouille de données. Ainsi, dans le chapitre 4, nous avons construit un contexte formel qui contient l'information structurelle et sémantique d'un multigraphe de co-appartenance et nous avons montré que les images des motifs symboliques du contexte formel correspondaient à des ensembles de nœuds dans le multigraphe

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

de co-appartenance. Par la suite, nous avons proposé une fonction de qualité pour estimer l'homogénéité des ensembles de nœuds images des motifs symboliques et sélectionner ceux qui constituent des communautés. A cet effet, nous avons présenté les algorithmes Exhaustif et MaximalSampling qui permettent respectivement d'extraire tous les motifs symboliques fermés dont la mesure de qualité est supérieure à un seuil fixé et d'échantillonner les motifs les plus intéressants par rapport à la fonction de qualité. Au chapitre 5, nous avons défini une fonction d'appartenance généralisée des nœuds aux couches qui prend en compte leurs similarités structurelle et sémantique et qui permet de construire le jeu de données numérique qui représente le multigraphe de co-appartenance. Dans cette partie, nous avons calculé les similarités sémantiques à partir des représentations vectorielles des attributs textuels obtenus à l'aide du modèle de traitement du langage naturel BERT. Ensuite, nous avons montré comment les motifs d'intervalles fermés évalués à l'aide d'une fonction de qualité appropriée permettent la détection des communautés du multigraphe de co-appartenance. L'algorithme Mimetic permet d'extraire tous les top-k motifs d'intervalles diversifiés du jeu de données numérique en utilisant la fonction de qualité et une mesure de diversification des motifs. Les images de ces motifs dans le jeu de données numérique sont des ensembles de nœuds qui appartiennent à des couches similaires en terme de structure et de sémantique et qui constituent des communautés dans le multigraphe de co-appartenance. Nous proposons une synthèse de la méthodologie suivie dans cette thèse pour détecter les communautés dans les multigraphes de co-appartenance à la figure C.1 de l'annexe C.

Nous avons évalué la qualité des résultats obtenus par les algorithmes proposés et les avons comparés à ceux des méthodes de l'état de l'art. Pour cela, nous avons utilisé des métriques basées sur l'exceptionnalité sémantique comme le TF-IDF et la divergence de Kullback-Leibler que nous avons appliquées aux attributs textuels des nœuds (les biographies des utilisateurs pour l'exemple Twitter) appartenant à la communauté par rapport aux attributs des nœuds en dehors de la communauté. Cette mesure d'évaluation a l'avantage d'utiliser des données indépendantes de celles qui ont permis la détection des communautés. En effet, pour l'exemple Twitter, ce sont les données relatives aux listes qui sont utilisées pour détecter les communautés tandis que les biographies des utilisateurs sont utilisées uniquement pour l'évaluation. Par ailleurs, nous avons aussi évalué les résultats en calculant le degré de correspondance entre les résultats obtenus et la vérité terrain des communautés avec l'exemple du multigraphe de co-appartenance issu de Wikipedia pour lequel cette information est disponible. Nous avons observé que les algorithmes que nous avons développés spécifiquement pour les multigraphes de co-appartenance obtiennent de meilleurs résultats que les algorithmes de l'état de l'art qui sont eux destinés aux multigraphes classiques.

La méthodologie en deux étapes, présentée dans cette thèse, permet de traiter des multigraphes de co-appartenance de taille indéfini à condition de choisir un domaine d'intérêt et d'identifier les nœuds graines pertinents qui le représentent. La prise en compte de la sémantique et notamment à l'aide des modèles de traitement du langage naturel (comme le modèle BERT) permet de détecter des communautés de meilleure qualité dans le multigraphe de co-appartenance.

Le travail présenté dans cette thèse ouvre la porte à de nouvelles directions de recherche qui concernent le calcul, à partir des communautés retournées par les méthodes de fouille de données, de partitionnements de tous les nœuds du graphe ; et la généralisation de ces méthodes la détection de communautés pour qu'elles prennent en compte l'évolution temporelle du multigraphe de co-appartenance dynamique.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Tout d'abord, nous avons remarqué que les communautés retournées par nos algorithmes ne recouvrent pas tous les nœuds du multigraphe de co-appartenance. Il serait intéressant de développer une méthode qui calcule un partitionnement de tous les nœuds du graphe à partir des résultats des méthodes présentées ici. Il est possible, par exemple, d'utiliser un algorithme de propagation de labels en l'initialisant avec l'ensemble des communautés calculées à l'aide d'un de nos algorithmes. La propagation de labels utilise une procédure de vote pour assigner une communauté à un nœud en fonction des communautés de ses nœuds voisins. Notons aussi la possibilité de pondérer les votes avec les similarités entre les attributs sémantiques associés aux nœuds et aux couches du multigraphe. Par ailleurs, nous avons souligné au chapitre 2 que des informations temporelles sont souvent associées aux multigraphes de co-appartenance ce qui permet de définir des multigraphes de co-appartenance dynamiques. Une potentielle direction de recherche future serait d'adapter la méthodologie de détection de communautés pour prendre en compte l'évolution temporelle de ces graphes. Pour cela, il est nécessaire de modifier les algorithmes du chapitre 3 de sorte que le sous-graphe d'intérêt évolue en fonction de l'évolution temporelle du multigraphe de co-appartenance. Il faut aussi généraliser les définitions des contextes formels et des jeux de données numériques au cas dynamique et remanier les algorithmes d'extraction de motifs des chapitres 4 et 5 pour qu'ils calculent l'évolution des motifs dynamiques qui varient en fonction de l'évolution temporelle du sous-graphe d'intérêt. Le résultat est alors un ensemble de communautés de nœuds temporelles qui évoluent avec le multigraphe de co-appartenance dynamique ce qui permet d'en faire une analyse longitudinale pour analyser son évolution historique mais aussi de construire des outils d'analyse en continu qui permettent de suivre en direct l'évolution des communautés du multigraphe de co-appartenance dynamique.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Annexe A

Graphes : définitions

Nous rappelons, dans cette annexe, les principaux concepts et propriétés associés aux graphes et définissons certains types de graphes qui ne l'ont pas été dans la section 2.1.

Adjacence, voisinage et degrés des nœuds

En reprenant la définition 1 d'un graphe présente dans la section 2.1 du chapitre 2, nous rappelons ici certains concepts associés à ses nœuds comme l'adjacence, le voisinage et le degré, et d'autres associés au graphe dans son intégralité comme la densité et la matrice laplacienne.

Définition 33. *Adjacence, degré et densité*

- *Deux noeuds d'un graphe sont dits **adjacents** s'ils ont au moins un lien en commun.*
- *Deux liens d'un graphe sont dits **adjacents** s'ils ont au moins un noeud en commun.*
- *Le **degré** d'un noeud v, noté $d(v)$, est égal au nombre de liens qui le relient à d'autres noeuds du graphe.*
- *Pour un graphe G, la distribution de degrés, notée $\mathcal{D}_G$, est la distribution probabiliste des degrés de ses noeuds.*
- *La **densité** d'un graphe G, notée $D(G)$, est définie par le rapport entre le nombre de liens existants dans G et le nombre de liens possibles dans G : $D(G) = \frac{2|E|}{|V|.(|V|-1)}$*

Définition 34. *Le **voisinage** d'un nœud v dans un graphe $G = (V, E)$ noté $\mathcal{N}(v)$ est le sous-graphe pour lequel l'ensemble de nœuds contient v et les nœuds adjacents à v, et l'ensemble des liens contient les liens incidents à v.*

Définition 35. *La **matrice laplacienne non normalisée** L d'un graphe $G = (V, E)$ est donnée par : $L = D - A$ où A est la matrice d'adjacence de G et D est la matrice diagonale contenant les degrés des nœuds de G.*

Connectivité dans un graphe

Les liens d'un graphe définissent sa structure de connectivité et permettent de calculer les distances entre toutes les paires de nœuds du graphe.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Définition 36. *Un **sous-graphe** de $G_1 = (V, E)$ est un graphe simple $G'_1 = (V', E')$ tel que $V' \subseteq V$ et $E' \subseteq E$.*

*Le **sous-graphe engendré** par un ensemble de nœuds V* est le sous-graphe maximal contenant les nœuds V* : $G^*_1 = (V^*, E_{V^*})$, i.e. $E_{V^*} = \{e = (v_1, v_2) \in E | v_1, v_2 \in V^*\}$.*

Définition 37. *Un **chemin** d'origine u et d'extrémité v dans un graphe $G = (V, E)$ est une suite $(u, ..., v)$ de nœuds deux à deux distincts telle que toute paire de nœuds successifs dans la suite ont un lien qui les relie dans le graphe.*

Définition 38. *Un graphe est **connexe** s'il contient des chemins reliants toutes les paires de nœuds.*
*Un sous-graphe connexe maximal est appelée **composante connexe** du graphe.*

Définition 39. *Un **graphe complet** est un graphe $G = (V, E)$ pour lequel toutes les paires de nœuds sont reliées par un lien. On a alors $\forall v_1, v_2 \in V, (v_1, v_2) \in E$.*

Définition 40. *Dans un graphe $G = (V, E)$, une **clique** est un sous-graphe complet $G_c = (V_c, E_c)$ i.e. tel que $E_c \subseteq E$ et $\forall v_1, v_2 \in V_c, (v_1, v_2) \in E_c$.*

Définition 41. *Un **cycle** est un chemin $(u, ..., v)$ dans $G = (V, E)$ avec la condition supplémentaire que $(u, v) \in E$.*

Définition 42. *La **distance** entre deux nœuds du graphe est définie comme la longueur du **plus court chemin** qui les relie.*

Définition 43. *Deux graphes $G = (V, E)$ et $G' = (V', E')$ sont **isomorphes** s'il existe une bijection $f : V \leftrightarrow V'$ telle que $\forall u, v \in V : (u, v) \in E \Leftrightarrow (f(u), f(v)) \in E'$.*
*L'application f est appelée **isomorphisme** des graphes G et G'.*

Plusieurs types de graphes

Il existe une multitude de types de graphes qui permettent de modéliser différentes relations entre un ensemble d'entités. En plus de ceux présentés à la section 2.1 du chapitre 2, les définitions suivantes concernent les graphes dirigés, les graphes bipartites, les graphes attribués et les graphes dynamiques.

Définition 44. *Un **graphe dirigé** $G_2 = (V, E)$ est un graphe simple pour lequel les liens ne sont pas symétriques. Les paires de noeuds dans l'ensemble des liens sont ordonnés en un noeud source et un noeud destination. La matrice d'adjacence de G_2 vérifie :*

$$\exists V_i, V_j \in V \text{tels que } A_{ij} \neq A_{ji}$$

Définition 45. *Un **graphe bipartite** $G_4 = (V_1, V_2, E)$ est un graphe simple possédant la propriété :*

$$\forall e_{ij} = (V_i, V_j) \in E, \text{on a } V_i \in V_1 \text{et } V_j \in V_2$$

Tous les liens de E possèdent une extrémité dans V_1 et la deuxième dans V_2.

Définition 46. *Un **graphe attribué sur les noeuds** $G_5 = (V, E, \mathbf{v_G})$ est un graphe simple comportant des informations supplémentaires sur les noeuds sous forme d'attributs. Ceux-ci peuvent être de différents types : numériques, catégoriques ou textuels. En notant I l'espace des attributs, l'application $\mathbf{v_G} : V \rightarrow I$ permet de retourner les attributs associés à un noeud du graphe.*

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Définition 47. *Un **graphe attribué sur les liens** $G_6 = (V, E, \mathbf{e_G})$ contient des attributs sur ses liens.* $\mathbf{e_G} : E \rightarrow I$ *permet d'associer les attributs aux liens du graphe.*

Définition 48. *Étant donné un segment temporel discret T, un **graphe dynamique** $G_8 = (V, E, T)$ est un graphe simple pour lequel les liens $e_{ij} = (V_i, V_j, t_d, t_f) \in E$ ont un temps de début $t_d \in T$ et un temps de fin $t_f \in T$.*

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Annexe B

Analyse formelle de concepts : définitions

Nous rappelons, dans cette partie, certaines définitions de structures algébriques pour introduire la théorie de l'analyse formelle de concepts en utilisant (Ignatov, 2017). En particulier, nous définissons les treillis et présentons certaines de leurs propriétés et leur relation avec les contextes formels et les motifs symboliques. A des fins d'illustration, nous utilisons un exemple de contexte formel issu d'une version simplifiée du multigraphe de co-appartenance $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{L})$ dans laquelle les attributs textuels sont écartés. Pour ce contexte, l'ensemble des objets est l'ensemble des couches du graphe $\mathcal{L}$ et celui des attributs correspond aux noeuds du graphe $\mathcal{V}$.

Rappels d'algèbre

Définition 49. *Une **relation binaire** R entre deux ensembles A et B est un ensemble de paires (a, b) avec $a \in A$ et $b \in B$. C'est un sous-ensemble de $A \times B$.*

Exemple B.1. Dans $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathcal{L})$, il y a une relation binaire d'appartenance R_m entre l'ensemble de couches $\mathcal{L}$ et l'ensemble de noeuds $\mathcal{V}$:$v_i R_m l_j$ veut dire "l'élément $v_i \in \mathcal{I}$ appartient à la couche $l_j \in \mathcal{L}$".

Définition 50. *Une **relation d'ordre partiel** est une relation binaire R sur un seul ensemble A vérifiant $\forall a, b, c \in A$:*

- *aRa (réflexive)*
- *aRb et $a \neq b \implies$ not bRa (anti-symétrique)*
- *aRb et bRc $\implies$ aRc (transitive)*

*On utilise le symbole $\leqslant$ pour l'ordre partiel. Un **ensemble partiellement ordonné** ou **poset** est une paire $(A, \leqslant)$ avec A un ensemble et $\leqslant$ un ordre partiel sur A.*

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Exemple B.2. L'ensemble des parties de $\mathcal{L}$ noté $\mathcal{P}(\mathcal{L})$ peut être partiellement ordonné en utilisant la relation d'inclusion $\leqslant_l = \subseteq$.

$\forall L_i, L_j \in \mathcal{P}(\mathcal{L})$,

$$L_i \leqslant_l L_j \iff L_i \subseteq L_j$$

La relation d'inclusion est réflexive, anti-symétrique et transitive.

Définition 51. *Etant donné un poset $(A, \leqslant)$ et $a, b, c \in A$:*

- *a est **le voisin inférieur** de $b \iff a \leqslant b$ et $\nexists\, c \mid a \leqslant c \leqslant b$;*
- *qui équivaut à b est **le voisin supérieur** de a.*

On note $a \prec b$.

Définition 52. *Etant donné un poset $(A, \leqslant)$ et $Q \subseteq A$:*

- *Une **borne inférieure** de Q est $l \in A$ tel que $l \leqslant q,\ \forall q \in Q$*
- *Une **borne supérieure** de Q est $l \in A$ tel que $q \leqslant l,\ \forall q \in Q$*
- *S'il existe un plus grand élément pour l'ordre partiel $\leqslant$ dans l'ensemble des bornes inférieures de Q, il est appelé **l'infimum (ou le meet)** de Q et est noté $\bigwedge Q$*
- *S'il existe un plus petit élément pour l'ordre partiel $\leqslant$ dans l'ensemble des bornes supérieures de Q, il est appelé **le supremum (ou le join)** de Q et est noté $\bigvee Q$*

Définition 53. *Un **treillis** est un poset dans lequel chaque paire d'éléments admet une borne supérieure et une borne inférieure. Un poset $\mathbf{A} = (A, \leqslant)$ est un **treillis** si :*

$$\forall a, b \in A, \exists \bigwedge \{a, b\} \text{ and } \exists \bigvee \{a, b\}$$

Exemple B.3. $(\mathcal{P}(\mathcal{L}), \leqslant_l)$ est un treillis.

$\forall L_i, L_j \in \mathcal{P}(\mathcal{L})$,

$$\bigwedge \{L_i, L_j\} = min_{\leqslant_l}\{L \in \mathcal{P}(\mathcal{L}) \mid (L_i \subseteq L) \wedge (L_j \subseteq L)\}$$
$$\bigvee \{L_i, L_j\} = max_{\leqslant_l}\{L \in \mathcal{P}(\mathcal{L}) \mid (L_i \supseteq L) \wedge (L_j \supseteq L)\}$$

Définition 54. *Un poset $\mathbf{A} = (A, \leqslant)$ est un **treillis complet** si :*

$$\forall Q \subseteq A, \exists \bigwedge Q \text{ and } \exists \bigvee Q$$

*Dans un treillis complet $\mathbf{A} = (A, \leqslant)$, le plus grand élément $\bigvee A$ est appelé **l'élément unité** et est noté $\mathbf{1_A}$. Le plus petit élément $\bigwedge A$ est appelé **l'élément zéro** et est noté $\mathbf{0_A}$.*

Exemple B.4. $(\mathcal{P}(\mathcal{L}), \leqslant_l)$ est un treillis complet.

$\forall \mathfrak{L} \subseteq \mathcal{P}(\mathcal{L})$,

$$\bigwedge \mathfrak{L} = min_{\leqslant_l}\{L \in \mathcal{P}(\mathcal{L}) \mid \forall L' \in \mathfrak{L},\ L' \subseteq L\} \quad \bigvee \mathfrak{L} = max_{\leqslant_l}\{L \in \mathcal{P}(\mathcal{L}) \mid \forall L' \in \mathfrak{L},\ L' \supseteq L\}$$

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Lemme 1. Bergman (1995) : Soient S, T deux ensembles, et $R \subseteq S \times T$ une relation. Pour $A \subseteq S$ et $B \subseteq T$, on écrit :

$$A^* = \{t \in T \mid \forall a \in A, \ aRt\} \subseteq T$$
$$B^* = \{s \in S \mid \forall b \in B, \ sRb\} \subseteq S$$

pour définir les deux applications *, la première de $\mathcal{P}(S)$ dans $\mathcal{P}(T)$ et la deuxième de $\mathcal{P}(T)$ dans $\mathcal{P}(S)$. Alors, pour $A, A' \subseteq S, \ B, B' \subseteq T$, on a :

- (i) $A \subseteq A' \Rightarrow A^* \supseteq A'^*$ et $B \subseteq B' \Rightarrow B^* \supseteq B'^*$
- (ii) $A^{**} \supseteq A$ et $B^{**} \supseteq B$
- (iii) $A^{***} = A^*$ et $B^{***} = B^*$
- (iv)** : $\mathcal{P}(S) \to \mathcal{P}(S)$ et ** : $\mathcal{P}(T) \to \mathcal{P}(T)$ sont des opérateurs de fermeture pour S et T respectivement.
- (v) Les ensembles $A^*, (A \subseteq S)$ sont précisément les sous-ensembles fermés de T, et les ensembles $B^*, (B \subseteq T)$ sont les sous-ensembles fermés de S, par rapport à l'opérateur de fermeture **.
- (vi) Les applications *, restreintes aux ensembles fermés, donnent un anti-isomorphisme (une bijection à ordre inversé) entre les treillis complets des sous-ensembles **-fermés de S et de T.

Définition 55. *Soient S et T deux ensembles, alors la paire d'applications* $* : \mathcal{P}(S) \to \mathcal{P}(T)$ *et* $* : \mathcal{P}(T) \to \mathcal{P}(S)$ *qui vérifie les conditions (i) et (ii) du* ***Lemme 1.*** *est appelée* ***connexion de Galois*** *entre les ensembles S et T.*

Exemple B.5. La paire d'applications :

$$\phi : \mathcal{P}(\mathcal{L}) \to \mathcal{P}(\mathcal{V})$$
$$L \to \{v \in \mathcal{V} \mid \forall l \in L, \ vR_m l\}$$

et

$$\psi : \mathcal{P}(\mathcal{V}) \to \mathcal{P}(\mathcal{L})$$
$$V \to \{l \in \mathcal{L} \mid \forall v \in V, \ vR_m l\}$$

forme une connexion de Gallois entre les ensembles $\mathcal{L}$ et $\mathcal{V}$.

Proof : $\forall L, L' \subseteq \mathcal{L}, \ \forall V, V' \subseteq \mathcal{V}$,
(i) On suppose $L \subseteq L'$, alors

$$\{v \in \mathcal{V} \mid \forall l \in L, \ vR_m l\} \supseteq \{v \in \mathcal{V} \mid \forall l \in L', \ vR_m l\}$$

Ce qui donne $\phi(L) \supseteq \phi(L')$.
D'une façon similaire, on suppose $V \subseteq V'$, alors

$$\{l \in \mathcal{L} \mid \forall v \in V, \ vR_m l\} \supseteq \{l \in \mathcal{L} \mid \forall v \in V', \ vR_m l\}$$

ce qui implique $\psi(V) \supseteq \psi(V')$.

(ii) On note $\phi(L) = V_L$: les listes dans L ont tous en commun les noeuds V_L. $\psi(V_L)$ retourne l'ensemble des couches L_c de $\mathcal{L}$ qui ont V_L parmi leurs noeuds. Alors, on a $L \subseteq L_c$ d'où $\psi(\phi(L)) \supseteq L$.
On note $\psi(V) = L_V$: Les noeuds V apparaissent dans les listes L_V. $\phi(L_V)$ retourne V_c l'ensemble des noeuds communs aux listes de L_V. Alors, on a $V \subseteq V_c$ d'où $\phi(\psi(V)) \supseteq V$.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Analyse formelle de concepts

Définition 56. *Un **contexte formel** $C = (G, M, D)$ est composé de deux ensembles G et M et d'une relation D entre G et M. Les ensembles G et M sont appelés les ensembles **objets** et **attributs** respectivement du contexte C. D est appelé le **dataset** du contexte. gDm or $(g, m) \in D$ veut dire que l'objet g possède l'attribut m.*

Exemple B.6. $C_L = (\mathcal{L}, \mathcal{V}, \mathcal{D})$, où $\mathcal{D}$ contient les relations entre les listes et leurs attributs (éléments listés et mots-clés), est un contexte formel.

Définition 57. *Un **motif symbolique** (ou motif d'itemsets) dans un contexte $C = (G, M, D)$ est un sous-ensemble d'attributs $B \subseteq M$. **L'image** d'un motif B est l'ensemble $A \subseteq G$ obtenu en appliquant l'opérateur * : $B^* = A$, c'est-à-dire telle que $\forall a \in A$ et $\forall b \in B$, $(a, b) \in D$.*

Définition 58. *Un **opérateur de fermeture** pour l'ensemble A est une application $\sigma : \mathcal{P}(A) \rightarrow \mathcal{P}(A)$ avec les propriétés suivantes $\forall X \subseteq A$:*

- $\sigma(\sigma(X)) = \sigma(X)$ *(idempotence)*
- $X \subseteq \sigma(X)$ *(extensité)*
- $X \subseteq Y \Rightarrow \sigma(X) \subseteq \sigma(Y)$ *(monotonicité)*

Exemple B.7. $\sigma_l : \mathcal{P}(\mathcal{L}) \rightarrow \mathcal{P}(\mathcal{L})$ tel que $\sigma_l = \psi \circ \phi$ est un opérateur de fermeture pour $\mathcal{L}$. $\sigma_a : \mathcal{P}(\mathcal{V}) \rightarrow \mathcal{P}(\mathcal{V})$ tel que $\sigma_a = \phi \circ \psi$ est un opérateur de fermeture pour $\mathcal{V}$.

Définition 59. *Une **classe d'équivalence** pour les motifs symboliques est un ensemble de motifs C_{eq} tel que $\forall B_1, B_2 \in C_{eq}$, $B_1^* = B_2^*$. Chaque classe d'équivalence contient un unique **motif fermé** B qui vérifie $\sigma(B) = B$. Le motif fermé est le plus grand élément de la classe d'équivalence selon la relation d'ordre de l'inclusion $\subseteq$.*

Définition 60. *Un **concept formel** dans un contexte $C = (G, M, D)$ est une paire (A, B) où B est un motif fermé c'est-à-dire tel que $\sigma(B) = B$, et A est l'image de B : $A^* = B$. Les ensembles A et B sont appelés, respectivement, **l'extention** et **l'intention** du concept formel (A, B).*

Définition 61. *L'ensemble des concepts formels du contexte C avec la relation D forment un treillis complet appelé **treillis des concepts** de C et noté $\underline{\mathfrak{B}}(C)$.*

Exemple B.8. En utilisant l'exemple de contexte $C_L = (\mathcal{L}, \mathcal{V}, \mathcal{D})$ et les opérateurs de fermeture σ_l et σ_a, on peut traverser le treillis de concepts $\underline{\mathfrak{B}}(C_L)$ et extraire les concepts qui vérifient certaines contraintes avec l'aide de l'algorithme Exhaustif (Sec. 4.3.3.1).

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Annexe C

Diagramme de synthèse

Le diagramme de la figure C.1 présente une synthèse de la méthodologie de détection de communautés dans les multigraphes de co-appartenance. Nous commentons les principales étapes de cette méthodologie :

(a) Construction du multigraphe de co-appartenance à partir des données de co-appartenance en suivant la procédure décrite dans la définition 6 du chapitre 2

(b) Sélection du sous-graphe d'intérêt dans le multigraphe de co-appartenance à partir d'un ensemble de nœuds graines et en utilisant un des deux algorithmes :
   — *Double couronnes* (section 3.4.2)
   — *Marche aléatoire* (section 3.4.3)

(c) Construction des structures de données pour l'extraction des motifs :
   — contexte formel : jeu de données à attributs binaires dérivé du sous-graphe d'intérêt (section 4.3)
   — jeu de données numériques dérivé du sous-graphe d'intérêt (section 5.3.1)

(d) Extraction de deux types de motifs :
   — Les motifs symboliques sont énumérés ou échantillonnés avec les algorithmes Exhaustif et MaximalSampling respectivement présentés à la section 4.3.3
   — Les top-k motifs d'intervalles diversifiés sont identifiés dans le jeu de données numériques à l'aide de l'algorithme Mimetic de la section 5.3.3

(e) Les images des motifs symboliques et des motifs d'intervalles correspondent à des communautés de nœuds dans les sous-graphes d'intérêt

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

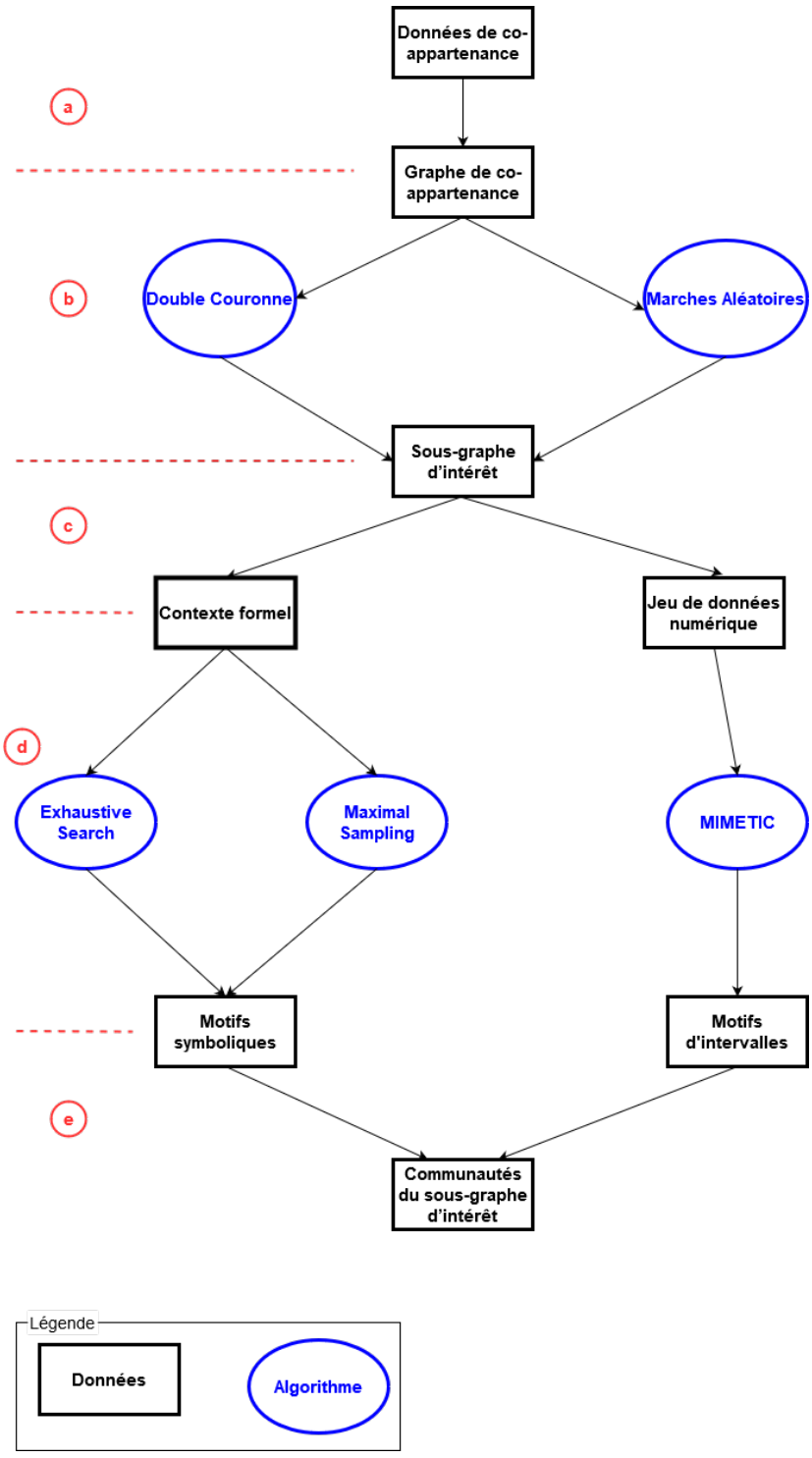


FIGURE C.1 – Diagramme de synthèse de la méthodologie de détection de communautés dans les multigraphes de co-appartenance

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Annexe D

Organization studies application

Institutional scholarship studies how individuals coexist and interact with social structures. Organizations and interorganizational relations within industries are a central focus of these studies. Hence, empirical research has so far largely relied on the observation of individual actors identified by their organizational attributes, and organizations identified by their industry characteristics. The flourishing of new types of social structures has sent an invitation to observe a broader range of actors beyond organizations stricto sensu, and to define the arena of interest beyond the boundaries of industry membership. However, in practice, these remain a favorite starting point of empirical investigations. In this article, we present a new method for the study of organizational fields that facilitates the identification of a large number and varied types of actors in a given field, provides a characterization of the relational structure of the field, and offers a content analysis on different sub-regions of the field. We test the method by replicating a previous study in the field of 'social impact of nonprofits', and show how it can contribute to operationalize mechanisms at play in the field. We conclude by noting that the principles of this method can extend beyond the dataset it is originally built on and facilitate a comparative approach to the study of fields. This contribution should enhance the value of the field as a theoretical construct by extending its operational reach.

The case of Twitter lists : Engines for social curation

At the aggregate level, Twitter lists happen to offer an unintended service : taken together, they are akin to a curation device effectively delineating millions of Twitter users in different groups and their associated topics. A study in computer science has shown that lists reveal rich and diverse sets of highly specialized and focused topical groups, spanning a variety of niche topics, at scale. This informational value could be leveraged to identify the topics of expertise of a given individual by scanning the names of the lists it is a member of. Lists also help identify top experts for a given issue by counting the Twitter

Methodology : A six-step procedure for the identification of actors in fields

The objective of this procedure is to identify actors of a field without imposing restrictions on the type or number of actors to be considered while putting the relational structure and

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

contents of the field in full view. The procedure relies on the classification of Twitter users in lists by fellow users, and is summarized as follows :

(1) Pick a small number of Twitter accounts ('seeds') which should be actors with a high relevance and visibility in the field of interest.

(2) For each seed, a network of similar Twitter users is identified because they are members in the same lists.

(3) 'Denoising' of the seed networks : Removing Twitter users which tend to be less strongly connected to the seed than the average using the *Marche aléatoire* algorithm.

(4) The networks obtained from each seed are merged : The outcome is a larger network, providing a vision of the actors populating the field and its surroundings.

(5) Visual exploration : Using an algorithm developed in the field of information visualization and network analysis, the network is laid out as a map to facilitate its interpretation.

(6) Sub-regions of the field are identified based on the patterns of connectivity between actors using the Louvain algorithm (Blondel et al., 2008) (actors densely connected with each other form a sub-region). Content analysis is performed on each sub-region to identify the topics of interest characterizing the actors of this sub-region. The analyst decides which sub-region(s) characterize the field, and which other sub-regions are better qualified as neighbors to the field. Sub-regions deemed to capture the field of interest can then become the new focus of analysis, while surrounding regions are ignored in subsequent steps.

The analyst iterates on step 6 until all sub-regions in view are judged to be constitutive of the field of interest. Steps 2 to 4 are performed with custom code written in Java and Python for the purpose of this study, and made available publicly (See Appendix I). Step 5 is conducted with Gephi, an open-source desktop software for graph visualization and exploration. Step 6 is conducted with Gephi (for the identification of sub-regions) and with custom code (for the content analysis),

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Bibliographie

Nazanin Afsarmanesh and Matteo Magnani. Finding overlapping communities in multiplex networks. *CoRR*, abs/1602.03746, 2016.

Simon Andrews. In-close2, a high performance formal concept miner. In Simon Andrews, Simon Polovina, Richard Hill, and Babak Akhgar, editors, *Conceptual Structures for Discovering Knowledge*, pages 50–62, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-22688-5.

James P. Bagrow and Erik M. Bollt. Local method for detecting communities. *Phys. Rev. E*, 72 :046108, Oct 2005.

Michael J. Barber and John W. Clark. Detecting network communities by propagating labels under constraints. *Physical Review E*, 80(2), Aug 2009. ISSN 1550-2376.

Asa Ben-Hur, Andre Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 2002 :6–17, 02 2002.

Mohamed Benabdelkrim, Clément Levallois, Jean Savinien, and Céline Robardet. Opening fields : A methodological contribution to the identification of heterogeneous actors in unbounded relational orders. *M@n@gement*, 23(1) :4–18, Mar. 2020a.

Mohamed Benabdelkrim, Jean Savinien, and Céline Robardet. Finding interest groups from twitter lists. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, SAC '20, page 1885–1887, New York, NY, USA, 2020b. Association for Computing Machinery. ISBN 9781450368667.

Mohamed Benabdelkrim, Céline Robardet, and Jean Savinien. Leveraging semantic for community mining in multilayer networks. *Proceedings of the Canadian Conference on Artificial Intelligence*, 6 2021. https ://caiac.pubpub.org/pub/k0erntsc.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null) :1137–1155, March 2003. ISSN 1532-4435.

George M. Bergman. *An Invitation to General Algebra and Universal Constructions*. Springer, Cham, 1995.

Michele Berlingerio, Fabio Pinelli, and Francesco Calabrese. ABACUS : apriori-based community discovery in multidimensional networks. *CoRR*, abs/1303.2025, 2013.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Parantapa Bhattacharya, Saptarshi Ghosh, Juhi Kulshrestha, Mainack Mondal, Muhammad Bilal Zafar, Niloy Ganguly, and Krishna P. Gummadi. Deep twitter diving : Exploring topical groups in microblogs at scale. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, CSCW '14, pages 197–210, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2540-0.

Bringmann Bjorn, Nijssen Siegfried, Tatti Nikolaj, Vreeken Jilles, and Zimmermann Albrecht. Tutorial : Mining sets of patterns. In *Tutorial ECML/PKDD*, 2010.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment*, 2008(10) :P10008, 2008.

Brigitte Boden, Stephan Günnemann, Holger Hoffmann, and Thomas Seidl. Mimag : mining coherent subgraphs in multi-layer graphs with edge labels. *Knowl. Inf. Syst.*, 50(2) :417–446, 2017.

Mario Boley, Thomas Gärtner, and Henrik Grosskreutz. Formal concept sampling for counting and threshold-free local pattern mining. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2010, April 29 - May 1, 2010, Columbus, Ohio, USA*, pages 177–188, 2010.

Mario Boley, Claudio Lucchese, Daniel Paurat, and Thomas Gärtner. Direct local pattern sampling by efficient two-step random procedures. In *International Conference on Knowledge Discovery and Data Mining – SIGKDD*, pages 582–590. ACM, 2011.

Mario Boley, Sandy Moens, and Thomas Gärtner. Linear space direct pattern sampling using coupling from the past. In *International Conference on Knowledge Discovery and Data Mining, SIGKDD*, pages 69–77. ACM, 2012.

Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. *CoRR*, abs/1406.3676, 2014.

Jean-François Boulicaut, Luc De Raedt, and Heikki Mannila. *Constraint-based mining and inductive databases*, volume 3848 of *Lecture Notes in Computer Science, LNCS*. Springer, February 2006.

Oualid Boutemine and Mohamed Bouguessa. Mining community structures in multidimensional networks. *ACM Transactions on Knowledge Discovery from Data*, 11 :1–36, 06 2017.

C. Bouveyron, P. Latouche, and R. Zreik. The stochastic topic block model for the clustering of vertices in networks with textual edges. *Statistics and Computing*, 28(1) :11–31, Oct 2016. ISSN 1573-1375.

Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Görke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity - np-completeness and beyond. *Univ., Fak. für Informatik, Bibliothek*, page 33, 12 2006.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Ronald L Breiger, Scott A Boorman, and Phipps Arabie. An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling. *Journal of Mathematical Psychology*, 12(3) :328–383, 1975. ISSN 0022-2496.

Vincenza Carchiolo, Alessandro Longheu, Michele Malgeri, and Giuseppe Mangioni. Communities unfolding in multislice networks. In *Complex Networks*, volume 116, pages 187–195. Springer, 2010.

Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. Representation learning for attributed multiplex heterogeneous network. *CoRR*, abs/1905.01669, 2019.

Vineet Chaoji, Mohammad Al Hasan, Saeed Salem, Jérémy Besson, and Mohammed J. Zaki. ORIGAMI : A novel and effective approach for mining representative orthogonal graph patterns. *Statistical Analysis and Data Mining*, 1(2) :67–84, 2008.

J. Chen, O. Zaïane, and R. Goebel. Local community identification in social networks. In *2009 International Conference on Advances in Social Network Analysis and Mining*, pages 237–242, July 2009.

Jingchun Chen and Bo Yuan. Detecting functional modules in the yeast protein–protein interaction network. *Bioinformatics*, 22(18) :2283–2290, 07 2006. ISSN 1367-4803.

Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70 :066111, Dec 2004.

Aaron Clauset, Cristopher Moore, and M. E. J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191) :98–101, May 2008. ISSN 1476-4687.

Leon Danon, Jordi Duch, Alex Arenas, and Albert Diaz-Guilera. Community structure identification. *Journal of Statistical Mechanics*, P09008 :93–113, 06 2007.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6) :391–407, 1990.

María José Del Jesus, Pedro González, Francisco Herrera, and Mikel Mesonero. Evolutionary fuzzy rule induction process for subgroup discovery : A case study in marketing. *Fuzzy Systems, IEEE Transactions on*, 15 :578 – 592, 09 2007.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019.

Stijn Van Dongen. Performance criteria for graph clustering and markov cluster experiments. Technical report, NATIONAL RESEARCH INSTITUTE FOR MATHEMATICS AND COMPUTER SCIENCE IN THE, 2000.

Vladimir Dzyuba, Matthijs van Leeuwen, and Luc De Raedt. Flexible constrained sampling with guarantees for pattern mining. *CoRR*, abs/1610.09263, 2016.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Daniel Edler, Ludvig Bohlin, and Martin Rosvall. Mapping higher-order network flows in memory and multilayer networks with infomap. *CoRR*, abs/1706.04792, 2017.

P Erdös and A Rényi. On random graphs i. *Publicationes Mathematicae Debrecen*, 6 :290–297, 1959.

Justin Fagnan, Osmar Zaïane, and Denilson Barbosa. Using triads to identify local community structure in social networks. In *Int. Conf. on Advances in Social Networks Analysis and Mining – ASONAM*, pages 108–112. IEEE Press, 2014.

Wenyi Fang, Xin Wang, Longzhao Liu, Zhaole Wu, Shaoting Tang, and Zhiming Zheng. Community detection through vector-label propagation algorithms, 2020.

Gary Flake, Robert Tarjan, and Kostas Tsioutsiouliklis. Graph clustering and minimum cut trees. *Internet Mathematics*, 1, 01 2003.

Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5) :75 – 174, 2010. ISSN 0370-1573.

Santo Fortunato and M Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, January 2007.

Alex A. Freitas. *Basic Concepts of Evolutionary Algorithms*, pages 79–106. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002. ISBN 978-3-662-04923-5.

Bernhard Ganter and Sergei O. Kuznetsov. Pattern structures and their projections. In Harry S. Delugach and Gerd Stumme, editors, *Conceptual Structures : Broadening the Base*, pages 129–142, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. ISBN 978-3-540-44583-8.

Bernhard Ganter, Rudolf Wille, and Karl Erich Wolff. *Beiträge zur Begriffsanalyse : Vorträge der Arbeitstagung Begriffsanalyse, Darmstadt 1986*. Wissenschaftsverlag, 1987.

Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Political discourse on social media : Echo chambers, gatekeepers, and the price of bipartisanship. In Pierre-Antoine Champin, Fabien L. Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis, editors, *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 913–922. ACM, 2018.

M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12) :7821–7826, 2002. ISSN 0027-8424.

Robert Godin, Rokia Missaoui, and Hassan Alaoui. Incremental concept formation algorithms based on galois (concept) lattices. *Computational Intelligence*, 11(2) :246–267, 1995.

Benjamin H. Good, Yves-Alexandre de Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4), Apr 2010. ISSN 1550-2376.

Derek Greene, Derek O'Callaghan, and Pádraig Cunningham. Identifying topical twitter communities via user list aggregation. *CoRR*, abs/1207.0017, 2012.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Steve Gregory. An algorithm to find overlapping community structure in networks. In Joost N. Kok, Jacek Koronacki, Ramón López de Mántaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron, editors, *Knowledge Discovery in Databases : PKDD 2007, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007, Proceedings*, volume 4702 of *Lecture Notes in Computer Science*, pages 91–102. Springer, 2007.

Aditya Grover and Jure Leskovec. node2vec : Scalable feature learning for networks. *CoRR*, abs/1607.00653, 2016.

R. Guimerà, S. Mossa, A. Turtschi, and L. A. N. Amaral. The worldwide air transportation network : Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences*, 102(22) :7794–7799, 2005. ISSN 0027-8424.

Obaida Hanteer, Roberto Interdonato, Matteo Magnani, Andrea Tagarelli, and Luca Rossi. Community detection in multiplex networks. *CoRR*, abs/1910.07646, 2019.

Manel Hmimida and Rushed Kanawati. Community detection in multiplex networks : A seed-centric approach. *Networks and Heterogeneous Media*, 10 :71–85, 03 2015.

Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels : First steps. *Social Networks*, 5(2) :109 – 137, 1983. ISSN 0378-8733.

Jianbin Huang, Heli Sun, Yaguang Liu, Qinbao Song, and Tim Weninger. Towards online multiresolution community detection in large-scale networks. In *PloS one*, 2011.

X. Huang, D. Chen, and T. et al. Ren. A survey of community detection methods in multilayer networks. *Data Min Knowl Disc*, pages 1–45, 2021.

Dmitry I. Ignatov. Introduction to formal concept analysis and its applications in information retrieval and related fields. *CoRR*, abs/1703.02819, 2017.

Roberto Interdonato, Andrea Tagarelli, Dino Ienco, Arnaud Sallaberry, and Pascal Poncelet. Local community detection in multilayer networks. *Data Min. Knowl. Discov.*, 31(5) :1444–1479, 2017.

Matthew Jackson. An overview of social networks and economic applications. *Handbook of Social Economics*, 1, 12 2011.

Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter : Understanding microblogging usage and communities. *of the 9th WebKDD and 1st SNA*, 43 :56–65, 01 2007.

LUCAS G. S. Jeub, MICHAEL W. MAHONEY, PETER J. MUCHA, and MASON A. PORTER. A local perspective on community structure in multilayer networks. *Network Science*, 5(2) : 144–163, 2017.

Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8) :888–905, 2000.

Bogumil Kaminski, Pawel Pralat, and François Théberge. An unsupervised framework for comparing graph embeddings. *CoRR*, abs/1906.04562, 2019.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Mehdi Kaytoue, Sergei Kuznetsov, and Amedeo Napoli. Revisiting numerical pattern mining with formal concept analysis. *IJCAI International Joint Conference on Artificial Intelligence*, 11 2011.

Mehdi Kaytoue, Marc Plantevit, Albrecht Zimmermann, Ahmed Anes Bendimerad, and Céline Robardet. Exceptional contextual subgraph mining. *Machine Learning*, 106(8) : 1171–1211, 2017.

B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49(2) :291–307, 1970.

Dongwoo Kim, Yohan Jo, Il chul Moon, and Alice Oh. Analysis of twitter lists as a potential source for discovering latent characteristics of users. In *CHI 2010 Workshop on Microblogging : What and How Can We Learn From It*, 2010.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

M. Kivela, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3) :203–271, Jul 2014. ISSN 2051-1329.

Arno J. Knobbe and Eric K. Y. Ho. Pattern teams. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Knowledge Discovery in Databases : PKDD 2006*, pages 577–584, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-46048-0.

Zhana Kuncheva and Giovanni Montana. Community detection in multiplex networks using locally adaptive random walks. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ASONAM '15, page 1308–1315, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450338547.

Sergei Kuznetsov. A fast algorithm for computing all intersections of objects in a finite semi-lattice. *Automatic Documentation and Mathematical Linguistics*, 27 :11–21, 01 1993.

Sergei Kuznetsov and Sergei Obiedkov. Comparing performance of algorithms for generating concept lattices. *J. Exp. Theor. Artif. Intell.*, 14 :189–216, 04 2002.

Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4), Oct 2008. ISSN 1550-2376.

Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3) : 033015, Mar 2009. ISSN 1367-2630.

Nada Lavrač, Branko Kavšek, Peter Flach, and Ljupčo Todorovski. Subgroup discovery with cn2-sd. *J. Mach. Learn. Res.*, 5 :153–188, December 2004. ISSN 1532-4435.

Conrad Lee and Padraig Cunningham. Community detection : Effective evaluation on large social networks. *Journal of Complex Networks*, 2 :19–37, 03 2014.

Matthijs Leeuwen and Arno Knobbe. Diverse subgroup set discovery. *Data Mining and Knowledge Discovery*, 25, 09 2012.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Ian X. Y. Leung, Pan Hui, Pietro Liò, and Jon Crowcroft. Towards real-time community detection in large networks. *Physical Review E*, 79(6), Jun 2009. ISSN 1550-2376.

Yixuan Li, Kun He, David Bindel, and John E. Hopcroft. Uncovering the small community structure in large networks : A local spectral approach. *CoRR*, abs/1509.07715, 2015.

Xin Liu, Weichu Liu, Tsuyoshi Murata, and Ken Wakita. A framework for community detection in heterogeneous multi-relational networks. *CoRR*, abs/1407.4989, 2014.

Tarcísio Lucas, Túlio C.P.B. Silva, Renato Vimieiro, and Teresa B. Ludermir. A new evolutionary algorithm for mining top-k discriminative patterns in high dimensional data. *Applied Soft Computing*, 59 :487 – 499, 2017. ISSN 1568-4946.

J. Ma and J. Fan. Local optimization for clique-based overlapping community detection in complex networks. *IEEE Access*, 8 :5091–5103, 2020.

Kevin T. Macon, Peter J. Mucha, and Mason A. Porter. Community structure in the united nations general assembly. *CoRR*, abs/1010.3757, 2010.

J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

Marvin Meeng, Wouter Duivesteijn, and Arno Knobbe. *ROC<sc>search</sc> — An ROC-guided Search Strategy for Subgroup Discovery*, pages 704–712. LWA, 2014.

M. Meil and David Heckerman. An experimental comparison of model-based clustering methods. *Machine Learning*, 42 :9–29, 01 2001.

M. Meila. Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98 :873–895, 2007.

Tomas Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, 2010.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013a.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013b. Association for Computational Linguistics.

Sandy Moens and Bart Goethals. Randomly sampling maximal itemsets. In *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics, IDEA@KDD 2013, Chicago, Illinois, USA, August 11, 2013*, pages 79–86, 2013.

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

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Tsuyoshi Murata and Tomoyuki Ikeya. A new modularity for detecting one-to-many correspondence of communities in bipartite networks. *Advances in Complex Systems*, 13(01) : 19–31, 2010.

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

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

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering : Analysis and an algorithm. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems : Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 849–856. MIT Press, 2001.

E. M. Norris. An algorithm for computing the maximal rectangles in a binary relation. *Revue Roumaine de Mathématiques Pures et Appliquées*, 23(2) :243–250, 1978.

Jan Outrata and Vilém Vychodil. Fast algorithm for computing fixpoints of galois connections induced by object-attribute relational data. *Inf. Sci.*, 185 :114–127, 02 2012.

Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435 :814–818, 07 2005.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk : Online learning of social representations. *CoRR*, abs/1403.6652, 2014.

S. Pramanik, R. Tackx, A. Navelkar, J. Guillaume, and B. Mitra. Discovering community structure in multilayer networks. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 611–620, 2017.

Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E*, 76 :036106, Sep 2007.

Vineeth Rakesh, Dilpreet Singh, Bhanukiran Vinzamuri, and Chandan K. Reddy. Personalized recommendation of twitter lists using content and network information. In *Conference on Weblogs and Social Media, ICWSM*, 2014.

Matthew J. Rattigan, Marc Maier, and David Jensen. Graph clustering with network structure indices. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 783–790, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933.

Jörg Reichardt and Stefan Bornholdt. When are networks truly modular ? *Physica D : Nonlinear Phenomena*, 224(1-2) :20–26, Dec 2006. ISSN 0167-2789.

Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4) :1118–1123, 2008. ISSN 0027-8424.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Holger Schwenk. Continuous space language models. *Computer Speech & Language*, 21(3) : 492–518, 2007. ISSN 0885-2308.

Jirí Síma and Satu Elisa Schaeffer. On the np-completeness of some graph cluster measures. *CoRR*, abs/cs/0506100, 2005.

Michael Steinbach, Levent Ertöz, and Vipin Kumar. The challenges of clustering high dimensional data. *Univ. Minnesota Supercomp. Inst. Res. Rep.*, 213, 01 2003.

Andrea Tagarelli, Alessia Amelio, and Francesco Gullo. Ensemble-based community detection in multilayer networks. *Data Mining and Knowledge Discovery*, 09 2017.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE : large-scale information network embedding. *CoRR*, abs/1503.03578, 2015.

Fei Tian, Bin Gao, Qing Cui, Enhong Chen, and Tie-Yan Liu. Learning deep representations for graph clustering. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1293–1299. AAAI Press, 2014.

Charles F Van Loan. *Matrix computations*. JHU Press, 1983.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

Srikar Velichety and Sudha Ram. Examining lists on twitter to uncover relationships between following, membership and subscription. In *World Wide Web – WWW*, pages 673–676. ACM, 2013.

Jilles Vreeken, Matthijs Leeuwen, and Arno Siebes. Krimp : Mining itemsets that compress. *Data Min. Knowl. Discov.*, 23 :169–214, 07 2011.

David L. Wallace. Comment. *Journal of the American Statistical Association*, 78(383) :569–576, 1983.

Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1225–1234, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322.

Liaoruo Wang, T. Lou, J. Tang, and J. Hopcroft. Detecting community kernels in large social networks. *2011 IEEE 11th International Conference on Data Mining*, pages 784–793, 2011.

Joyce Jiyoung Whang, David F. Gleich, and Inderjit S. Dhillon. Overlapping community detection using neighborhood-inflated seed expansion. *CoRR*, abs/1503.07439, 2015.

Dennis M. Wilkinson and Bernardo A. Huberman. A method for finding communities of related genes. *Proceedings of the National Academy of Sciences*, 101(suppl 1) :5241–5248, 2004. ISSN 0027-8424.

Rudolf Wille. Restructuring lattice theory : An approach based on hierarchies of concepts. In Ivan Rival, editor, *Ordered Sets*, pages 445–470, Dordrecht, 1982. Springer Netherlands. ISBN 978-94-009-7798-3.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés

Yuto Yamaguchi, Toshiyuki Amagasa, and Hiroyuki Kitagawa. Tag-based user topic discovery using twitter lists. In *ASONAM*, pages 13–20. IEEE Computer Society, 2011.

Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowl. Inf. Syst.*, 42(1) :181–213, January 2015. ISSN 0219-1377.

Hongming Zhang, Liwei Qiu, Lingling Yi, and Yangqiu Song. Scalable multiplex network embedding. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 3082–3088. ijcai.org, 2018.

C. Zhou, Y. Liu, X. Liu, Zhongyi Liu, and Jun Gao. Scalable graph embedding for asymmetric proximity. In *AAAI*, 2017.

Cette thèse est accessible à l'adresse : http://theses.insa-lyon.fr/publication/2021LYSEI100/these.pdf
© [M. Benabdelkrim], [2021], INSA Lyon, tous droits réservés