République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 - Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique

Option : Système Informatique

Thème

Détection des communautés basé sur l'importance des noeuds dans le réseau

---

Présenté par : Ouartsi Abdallah

Membres de jury :

N	Nom	Qualité
1	Bouroueih Douadi	Président
2	Louafi Wafa	Encadrante
3	Benhamida Nadjet	Examinateur

Septembre 2023

Remerciement

Tout d'abord, je remercie Allah le tout puissant de m'avoir donné le courage et la patience nécessaires à mener ce travail à son terme.

Tout d'abord, je remercie Allah le tout puissant de m'avoir donné le courage et la patience nécessaires à mener ce travail à son terme.

J'exprime notre profonde et sincère expression de remerciements a mon encadrante **Mme. Louafi Wafa**, d'avoir dirigé ce travail et patienter avec moi et pour la confiance qu'elle m'a accordé, pour sa disponibilité, ses conseils et ses idées innovantes tout au long de notre travail. Son œil critique m'a été très précieux pour structurer le travail et pour améliorer la qualité des différentes sections.

Je remercie vivement les membres du jury d'avoir accepté de siéger et de juger notre travail. Nous les remercions aussi pour l'honneur qu'ils nous font en acceptant d'examiner ce mémoire.

Aussi, je ne pouvais pas laisser passer l'occasion de saluer chaleureusement les enseignants du département de l'informatique qui nous ont permis de recevoir un enseignement de qualité.

Enfin mes remerciements les plus sincères à ma famille et à tous ceux qui ont contribué de près ou de loin à la réalisation de ce modeste travail.

Dédicace

Je dédie ce modeste travail à ma merveilleuse mère, dont l'amour et le soutien sont inestimables. Merci d'être toujours là pour moi, avec ton amour inconditionnel et ta sagesse infinie. Tu es la personne la plus précieuse de ma vie.

À mon père, le roc de la famille, qui m'a enseigné la force, la persévérance et la bienveillance. Ton exemple est une source d'inspiration constante pour moi, et je suis fier d'être ton enfant.

Et à mon cher frère Ilyas, avec qui j'ai partagé tant de moments inoubliables et qui est bien plus qu'un frère pour moi, il est un ami précieux. Ton rire contagieux et ton soutien inconditionnel illuminent ma vie.

À mon encadrante dévouée, je tiens à exprimer ma profonde gratitude pour votre soutien, votre guidance et votre dévouement tout au long de notre collaboration.

À mes amis extraordinaires, Wassim, Ayoub, Djassim, Hani, Anis, Seif, Bourhan, Oussama, Mouh, Akrem, Mouhmed et Abdnour, votre amitié est un trésor inestimable que je chéris chaque jour.

À l'équipe incroyable de Fast Food Le Prince, ainsi qu'à Djassim, Bilel Gro, Amine La Chi, Khaled Zdad, Borhan et Racim, vous avez été des compagnons de route exceptionnels.

À tous ceux qui me sont chers, à vous tous, merci pour votre soutien indéfectible et votre contribution à cette étape importante de ma vie. Merci

-Abdallah

Résumé

Avec l'essor des réseaux sociaux, la tâche de détection des communautés dans les réseaux est devenue de plus en plus difficile ces dernières années.

Dans le but de détecter les communautés, de nombreux algorithmes ont été proposés pour détecter des communautés disjoints. Le défi majeur de la détection des communautés du monde réel est de déterminer des communautés stables. Les sommets superposés appartiennent à certains communautés sont donc difficile d'être détectée. Dand ce memoire, nous avons développé une nouvelle méthode de détection de communautés basée sur la densité, notre méthode faire des regroupements selon des itérations avec certain critére de similarité.

Notre approche se distingue par son efficacité, sa simplicité et sa facilité d'implémentation. Nous avons comparé notre algorithme à plusieurs algorithmes de pointe en utilisant des réseaux réels, en évaluant les résultats à l'aide de la mesure de modularité Q. Les résultats que nous avons obtenus sont considérés comme acceptables.

Mots clés :

Détection des communautés , Densité local , Modularité, regroupement

Abstract

With the rise of social networks, the task of community detection in networks has become increasingly challenging in recent years.

In order to detect communities, numerous algorithms have been proposed to identify disjoint communities. The major challenge in real-world community detection is determining stable communities. Overlapping nodes belonging to multiple communities are therefore difficult to detect. In this thesis, we have developed a new community detection method based on density, where our method forms clusters through iterations using a specific similarity criterion.

Our approach stands out for its efficiency, simplicity, and ease of implementation. We compared our algorithm to several state-of-the-art algorithms using real networks, evaluating the results using the modularity measure Q. The results we obtained are considered acceptable.

Keywords :

Community detection , Local density , Modularity, clustering.

Table des matières

Table des figures

Liste des tableaux

Introduction

Context

Aujourd'hui, les réseaux sont devenus un moyen de communication indispensable et omniprésent. Avec l'avènement d'Internet et des médias sociaux, les individus, les organisations et même les machines sont connectés de manière complexe. Ces réseaux permettent de partager des informations, de collaborer, de diffuser des idées et de créer des communautés virtuelles.

La représentation des réseaux sous forme de graphes permet une étude et une compréhension plus aisées de leur structure. Dans cette représentation, chaque acteur du réseau est symbolisé par un nœud ou sommet, et les relations entre individus sont représentées par des arêtes[10]. Ainsi, les concepts de la théorie des graphes peuvent être appliqués à l'analyse des réseaux.

L'analyse de tels réseaux est issue des travaux de mathématiciens des graphes et a attiré l'attention dans d'autres domaines en raison de sa large application dans différents domaines, y compris la détection de communauté, qui est l'une des méthodes les plus fondamentales et classiques. question. La plupart des réseaux d'intérêt présentent des structures communautaires, dans lesquelles les nœuds sont organisés en groupes appelés communautés, clusters ou modules[11].

La détection de communauté implique de partitionner le réseau en groupes ou modules, chaque groupe représentant une communauté. Pour être considéré comme une communauté, un groupe ou un module ne doit pas être vide, ses nœuds doivent faire partie d'un graphe, chaque communauté doit être distincte (c'est-à-dire que l'ensemble des nœuds de deux communautés différentes ne peut pas être le même) et toutes les communautés doivent renvoyer un ensemble de nœuds dans le graphique.[12].

La recherche de structures de regroupement dans les réseaux sociaux, l'analyse des réseaux professionnels, la biologie des réseaux, voire l'analyse des réseaux criminels sont toutes importantes. Cela permet de mettre en évidence des structures cachées, des interactions complexes et des dynamiques sociales dans les réseaux. Plusieurs approches peuvent être utilisées pour identifier ces structures. Chaque approche présente ses propres avantages et limites, [3, 13, 14].

Cependant, malgré les développements dans le domaine de la détection des communautés dans les réseaux, le grand nombre de méthodes disponibles conduit à une question fondamentale : une méthode proposée basée sur l'importance des nœuds est-elle capable de relever le défi de la détection des communautés dans les réseaux présentant des chevauchements, c'est-à-dire des nœuds appartenant simultanément à plusieurs communautés ? Ceci, tout en gérant les réseaux de plus en plus volumineux et complexes qui émergent constamment dans notre monde interconnecté.

Pour évaluer la qualité des communautés détectées, nous utilisons la Modularité.

Objectif Principal

L'objectif principal de cette recherche est de développer une approche de détection communautaire qui est stable, précis et efficace dans les réseaux réels. Pour ce faire, nous avons mis en œuvre une nouvelle méthode de détection de communauté qui repose principalement sur l'identification des noeuds dans le réseau qui présentent une densité élevée, et de les regrouper aves ces voisins en tant que communautés.

Organisation du mémoire

Le mémoire se compose de quatre chapitres, il est organisé de la manière suivante :

Le premier chapitre :"La détection communautaire dans les graphes"

Dans ce chapitre,nous présentons approfondie met en lumière l'importance de la théorie des graphes et de la détection de communautés dans l'analyse des réseaux, en particulier

dans le contexte des réseaux sociaux, tout en insistant sur l'importance des mesures d'évaluation pour garantir des résultats pertinents.

Le deuxième chapitre :"L'importance des nœuds dans un réseau"

Présente différentes échelles pour évaluer l'importance des nœuds. Le chapitre se termine par une discussion détaillée de la densité, donnant des définitions et des formules pour la détection des communautés

Le troisième chapitre :"Méthodologie"

Commencons par explorer l'architecture fondamentale de notre systeme, en detaillant ensuite les differentes etapes du processus de conception, ainsi que les algorithmes novateurs que nous proposons. Enfin, pour clore ce chapitre, nous illustrerons notre approche par le biais d'un exemple concret, mettant en lumiere la demarche que nous avons adoptee de maniere tangible.

Le quatrième chapitre :"Implmentation"

Ce chapitre se divise en quatre sections : tout d'abord, une présentation des outils de programmation utilisés, puis une exposition de notre système. Ensuite, nous présentons les résultats obtenus avec les algorithmes existants sur nos jeux de données. Enfin, nous comparons les résultats de notre méthode avec celles des méthodes existantes pour conclure ce chapitre de manière éclairante.

Chapitre 1

La détection communautaire dans les graphes

1.1 Introduction

En effet, de nombreux systèmes complexes dans divers domaines, tels que la biologie, l'informatique, la linguistique, le commerce et bien d'autres, peuvent être représentés de manière abstraite sous la forme de réseaux. Un réseau peut être défini comme un ensemble d'entités, appelées nœuds, qui sont reliées entre elles par des liens ou des interactions [15]. Par exemple, dans un réseau social en ligne tel que Facebook, les entités sont les utilisateurs du réseau, qui sont représentés par des nœuds, et les liens sont les amitiés ou les relations entre les utilisateurs.

La structure d'un réseau peut être extrêmement complexe, avec de multiples connexions entre les nœuds. C'est dans cette structure que la détection des communautés intervient. L'objectif de la détection des communautés est de trouver des groupes d'entités qui ont des interactions étroites entre elles, tout en étant moins interconnectées avec les autres groupes. Ces groupes sont appelés communautés et peuvent être considérés comme des sous-ensembles du réseau qui partagent des caractéristiques ou des propriétés communes.

Afin de faciliter et de rendre pratique la détection de communautés dans un réseau donné, il est nécessaire d'utiliser une représentation graphique, telle que la représentation par graphe.

Dans ce chapitre, nous présentons l'état de l'art se rapportant à la détection de communautés,t, on va tout d'abord donner qullque Notations pour théorie des graphes et expliquer ce qu'une communauté et comment là représenter avant Analyser les objectifs et Examiner les méthodes de détection des communautés.Après on va retenir les travaux connexes les plus connus dans le domaine de détection des communautés.

1.2 Notations pour la théorie des graphes

Cette section fournit des définitions de certains éléments de la théorie des graphes.

1.2.1 Définitions

Graphe :

Un graphe est un ensemble de nœuds (ou sommets) reliés par des arêtes (ou liens)[16]. (Voir figure 4.1)

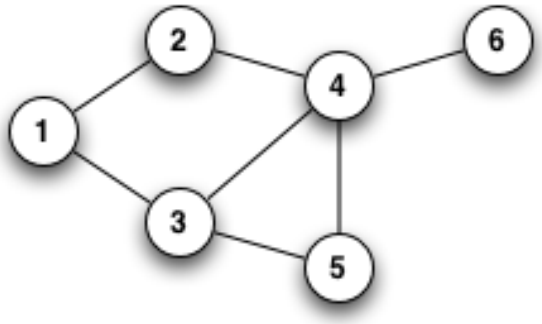

FIGURE 1.1 – Représentation d'un graphe à 6 nœuds et 7 arêtes .

Noeuds :

En théorie des graphes, un nœud (ou sommet) est un élément fondamental d'un graphe, utilisé pour représenter une entité ou un objet[17],Dans la figure 4.2, on peut observer que les points A, B et C sont des noeuds.

FIGURE 1.2 – Représentation des nœuds.

Lien :

C'est une relation entre deux nœuds d'un graphe. Les liens sont souvent représentés par des arêtes ou des arcs, qui sont des éléments qui relient deux nœuds ensemble[17].

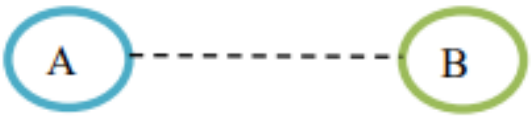

FIGURE 1.3 – Représentation d'un lien entre deux acteurs.

Degré :

En théorie des graphes, un degré fait référence au nombre de sommets adjacents à un sommet donné. Autrement dit, le degré d'un sommet est le nombre de liens qui le connectent aux autres sommets du graphe[18].

$$\text{Le degré du sommet} = \text{Nombre d'arêtes connectées au sommet} \tag{1.1}$$

Incidence :

Une arête $(u, v) \in E$ est dite incidente aux noeuds u et v, et ces noeuds sont dits voisins dans le graphe. Autrement dit, l'arête relie directement les noeuds u et v, et ces deux noeuds sont considérés comme étant adjacents ou voisins dans le graphe[18].

Graphe pondéré :

Un graphe dans lequel chaque arête est associée à un poids ou une valeur numérique[19].

Graphe orienté :

Un graphe dans lequel les arêtes ont une direction spécifique, c'est-à-dire qu'elles vont d'un sommet à un autre dans un sens donné[18]. (Voir figure 4.4)

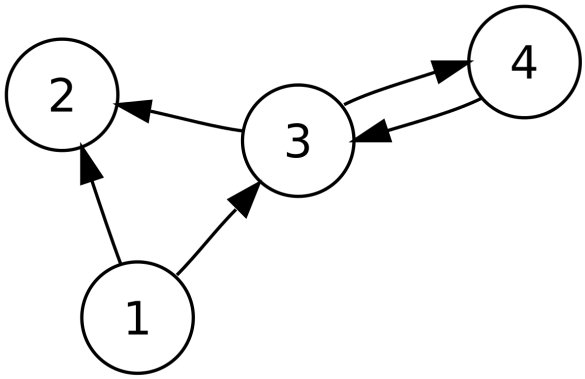

FIGURE 1.4 – Exmple de graphe orienté.

Graphe non orienté :

Un graphe dans lequel les arêtes n'ont pas de direction spécifique, c'est-à-dire qu'elles relient simplement deux sommets sans indiquer de sens[16].

Sous-graphe

Un sous-graphe est un graphe qui est formé à partir d'un graphe initial en sélectionnant un sous-ensemble de ses nœuds et de ses arêtes. Autrement dit, un sous-graphe est un graphe qui est obtenu en retirant certains nœuds et/ou certaines arêtes du graphe d'origine[16].

Formellement, soit G = (V, E) un graphe initial.

Un sous-graphe de G est un graphe G' = (V', E'), où V' est un sous-ensemble des nœuds de V ($V' \subseteq V$) et E' est un sous-ensemble des arêtes de E ($E' \subseteq E$) tels que chaque arête dans E' a ses deux extrémités dans V'.

Chemin :

Une séquence d'arêtes consécutives reliant deux sommets distincts dans le graphe[18].

Graphe connexe :

Un graphe dans lequel il existe un chemin entre chaque paire de sommets[16]. (Voir figure 4.5)

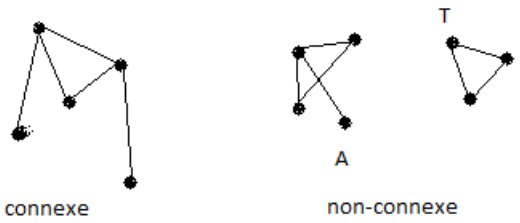

FIGURE 1.5 – Exmple de graphe orienté.

Graphe biparti

Un graphe dont les sommets peuvent être divisés en deux ensembles distincts, tels que toutes les arêtes relient un sommet d'un ensemble à un sommet de l'autre ensemble[20].

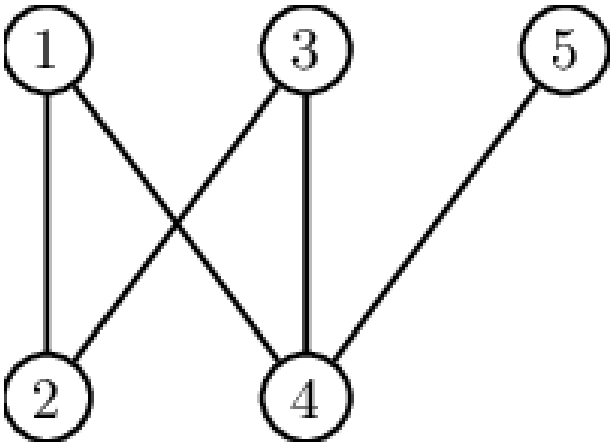

FIGURE 1.6 – Exmple de graph biparti.

1.3 Communautés

La communauté est un terme qui désigne un groupe de personnes qui partagent des intérêts, des valeurs ou des objectifs communs. Cela peut être une communauté locale, telle qu'un quartier ou un village, ou une communauté plus large, comme une communauté en ligne ou une communauté professionnele[21].

Selon Girvan et Newman[22]éfinir une communauté comme un groupe de Les liaison internes sont plus importantes que les relations externes.

En général, une communauté peut être définie comme un groupe de personnes qui partagent des traits, des intérêts, des valeurs, des objectifs ou des liens communs et qui

communiquent régulièrement entre elles.

La représentation graphique dans la Figure 4.7 montre une simplification visuelle de trois communautés distinctes, où chaque communauté est délimitée par des cercles en pointillés.

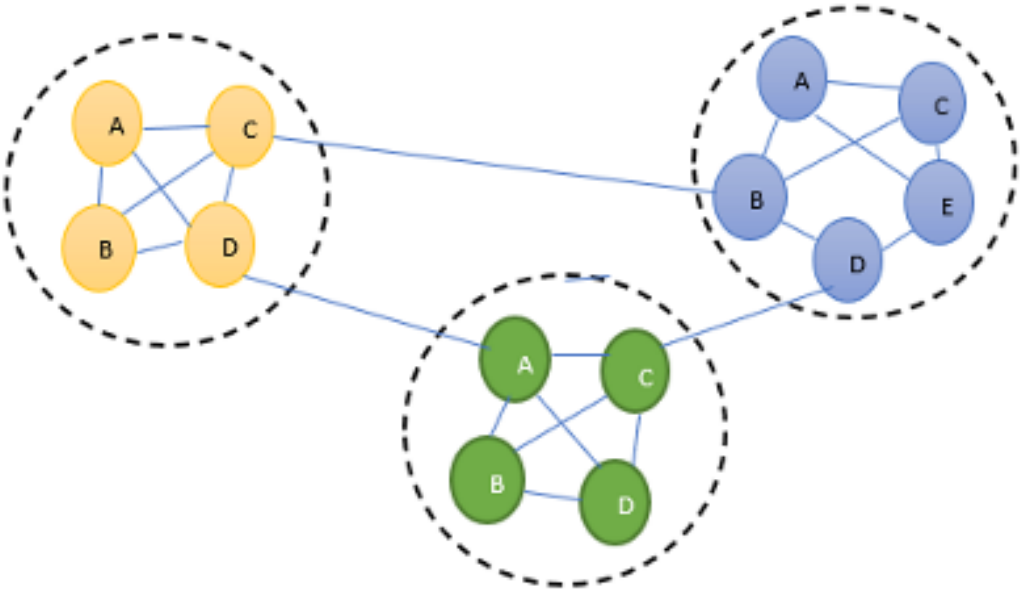

FIGURE 1.7 – Représentation de Trois communautés

1.3.1 Détection de communauté

La détection de communauté est une tâche d'analyse de réseau visant à identifier des groupes d'entités étroitement liées dans un réseau. L'objectif est de trouver des sous-groupes d'entités qui ont des liens plus forts les uns avec les autres que les entités extérieures au groupe. Cette tâche a fait l'objet de nombreuses recherches dans le domaine de l'analyse de réseau et a de nombreuses applications dans divers domaines tels que les médias sociaux, la biologie, la sociologie et la recommandation de contenu[11].
L'objectif principal de la détection de communauté dans les réseaux est de comprendre et d'identifier la structure sous-jacente des relations entre les entités.

Voici quelques objectifs courants associés à la détection de communauté[11, 23]

1. **Compréhension de la structure du réseau :** La détection de communauté permet de révéler les modèles de connexion, les clusters ou les sous-graphes significatifs présents dans le réseau.

2. **dentification des groupes fonctionnels :** La détection de communauté peut être utilisée pour identifier des groupes d'entités qui partagent des fonctions ou des rôles similaires dans le réseau.
3. **Détection de communautés influentes :** L'objectif peut être de trouver des communautés qui jouent un rôle central ou influent dans le réseau.
4. **Analyse de la polarisation ou de l'opinion :** La détection de communauté peut être utilisée pour analyser la polarisation des opinions dans un réseau social en identifiant des groupes qui partagent des perspectives ou des croyances similaires.
5. **Recommandation personnalisée :** En identifiant les communautés auxquelles un utilisateur appartient, il devient possible de proposer des recommandations personnalisées telles que des amis potentiels, des produits ou des contenus pertinents. Ces recommandations sont basées sur les intérêts et les affiliations communs entre les membres de ces communautés, ce qui permet de fournir des suggestions plus ciblées et précises.

1.3.2 Les méthodes de détection des communautés

Trouver des partitions communautaires est un problème central dans l'analyse de réseau, et de nombreuses recherches ont été consacrées à résoudre ce problème depuis les travaux pionniers de Girvan et Newman. Cependant, certaines questions restent ouvertes. En effet, contrairement aux graphes aléatoires, les réseaux complexes, les réseaux sociaux, présentent généralement des propriétés topologiques non triviales.

La détection des communautés en général est un domaine de recherche visant à identifier des sous-groupes cohérents et fortement interconnectés au sein d'un réseau. Les réseaux dans lesquels on recherche des communautés peuvent représenter différents types de systèmes tels que les réseaux sociaux, les réseaux biologiques, les réseaux de transport, etc. La détection des communautés est importante car elle permet de mieux comprendre la structure et les interactions au sein de ces réseaux.

Les méthodes de détection des communautés varient en fonction des caractéristiques du réseau, des objectifs de l'analyse et des hypothèses sous-jacentes. La plupart de ces classifications sont basées sur les types des algorithmes de détection des communautés et leurs principes méthodologiques.Fortunato a mené une étude approfondie et classé les

techniques en huit catégories[24].

Méthodes basées sur la recherche de sous-graphes denses :

Ces approches permettent d'identifier des sous-graphes compacts qui peuvent représenter des communautés. Elles utilisent des techniques telles que la détection de cliques, les mesures de densité de connexions ou la recherche de motifs fréquents pour repérer ces sous-graphes[25].

Méthodes basées sur la détection de structures hiérarchiques :

Ces méthodes visent à identifier des structures de communautés à différents niveaux d'organisation. Elles utilisent des approches de partitionnement hiérarchique pour révéler la structure des communautés à différentes échelles[4].

Méthodes basées sur l'optimisation de critères :

Ces méthodes formulent la détection des communautés comme un problème d'optimisation, où l'objectif est de maximiser ou minimiser des critères spécifiques. La modularité, qui mesure la qualité de la partition en communautés, est souvent utilisée comme critère d'optimisation.

Méthodes basées sur l'optimisation :

Ces méthodes formulent la détection des communautés comme un problème d'optimisation, cherchant à trouver la meilleure partition du réseau selon certains critères. Par exemple, la maximisation de la modularité est souvent utilisée comme critère d'optimisation.

Méthodes basées sur l'évaluation de partitions aléatoires :

Ces approches comparent la structure réelle du réseau avec des partitions aléatoires pour évaluer la significativité des communautés détectées. L'objectif est de trouver des partitions qui diffèrent de manière statistiquement significative des partitions aléatoires.

Méthodes basées sur la propagation d'étiquettes :

Ces méthodes attribuent initialement des étiquettes aux nœuds du réseau, puis les propagent en fonction de règles prédéfinies. Les nœuds partageant les mêmes étiquettes sont regroupés dans une communauté.

Méthodes basées sur l'apprentissage non supervisé :

Ces approches utilisent des techniques d'apprentissage automatique, telles que le clustering, pour détecter les communautés. Elles exploitent les caractéristiques topologiques ou les attributs des nœuds pour regrouper les nœuds en communautés.

Méthodes basées sur les modèles de mélange :

Ces méthodes considèrent les communautés comme des ensembles de nœuds générés par des modèles de mélange probabiliste. Elles cherchent à estimer les paramètres du modèle pour obtenir la meilleure partition en communautés.

Méthodes basées sur les random-walks :

Ces approches utilisent les random-walks pour explorer le réseau et identifier les structures de communautés. Les nœuds visités fréquemment par les random-walks sont regroupés dans la même communauté

La détection des communautés est un domaine de recherche actif, et de nombreuses méthodes et approches sont en constante évolution. Les chercheurs explorent de nouvelles techniques, évaluent et comparent les méthodes existantes, et cherchent à développer des approches plus efficaces et adaptées aux différents types de réseaux et aux défis spécifiques de chaque domaine d'application.

1.3.3 Les Algorithmes de détection de communauté

Algorithme Louvain

C'est une méthode populaire pour la détection de communautés basée sur l'optimisation de la modularité. Il a été proposé par Vincent D. Blondel, Jean-Loup Guillaume et Renaud

Lambiotte en 2008[26].Il le fait en affectant d'abord chaque nœud à sa propre communauté. Il examine ensuite chaque nœud et évalue le gain de modularité qui peut être obtenu en déplaçant ce nœud dans chacune des communautés de ses voisins. Le nœud est alors placé dans le voisinage voisin qui offre le plus grand gain de modularité.

Ce processus est répété pour chaque nœud du réseau, et l'algorithme s'arrête dès qu'aucun gain de modularité supplémentaire ne peut être obtenu. À ce stade, les communautés obtenues sont combinées pour former un graphique de quotient, puis le même processus de détection de communauté est effectué sur celui-ci. Ce processus est répété jusqu'à ce que le processus récursif n'améliore plus la modularité.

En résumé, l'algorithme de Louvain est un algorithme itératif qui utilise des approches ascendantes et descendantes pour identifier les communautés optimales dans un réseau en maximisant la modularité'[13, 27].

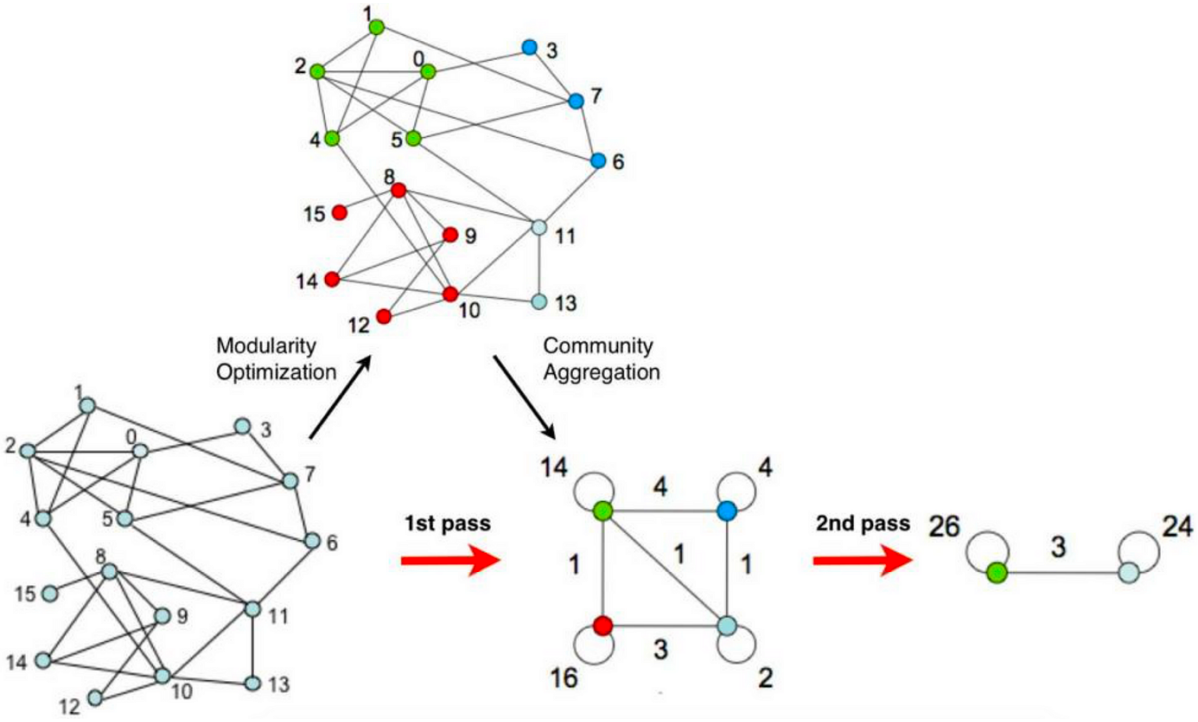

FIGURE 1.8 – Visualisation des étapes de l'algorithme de Louvain[1]

Algorithme Label Propagation (Propagation d'étiquettes)

L'algorithme de propagation d'étiquettes est une méthode de classification non supervisée pour regrouper des objets en fonction de leurs interactions ou de leurs similitudes dans un réseau.L'idée principale de cet algorithme est que chaque nœud se voit initialement attri-

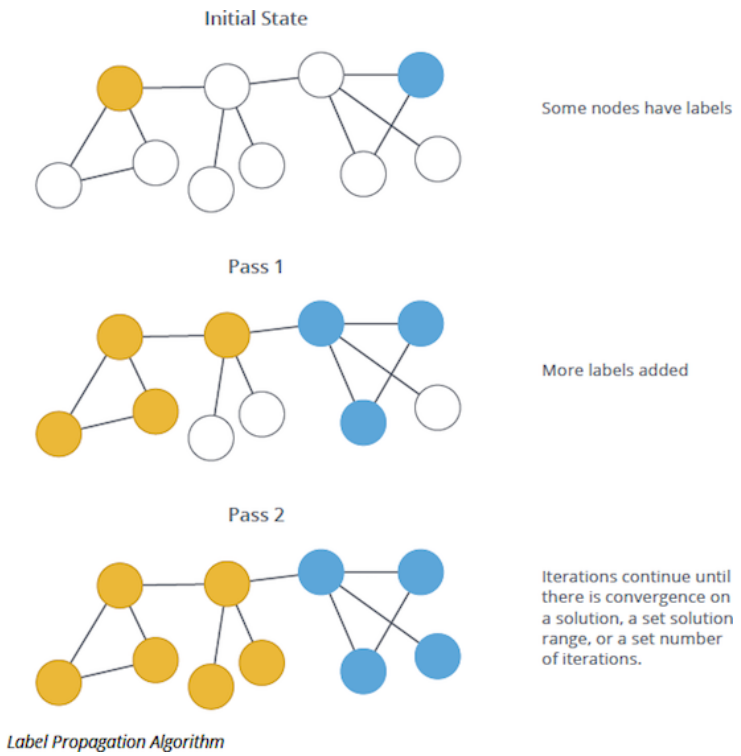

FIGURE 1.9 – Les étapes de l'algorithme de Label Propagation[2]

buer une étiquette unique.Par la suite, chaque nœud remplacera son étiquette par l'étiquette du plus grand nombre de ses voisins (ou un nombre aléatoire en cas d'égalité). Après plusieurs tentatives, un même label est souvent associé à des membres d'une même communauté.En fin,tous les libellés identiques entre eux font partie d'une communauté[28].

L'algorithme Label Propagation est relativement simple et efficace, ce qui en fait une approche populaire pour la détection de communautés dans les réseaux. Cependant, il peut être sensible à l'ordre de propagation des étiquettes, ce qui peut entraîner des résultats différents selon l'ordre choisi. Par conséquent, plusieurs exécutions de l'algorithme avec différents ordres de propagation des étiquettes peuvent être nécessaires pour obtenir des résultats plus fiables et stables.

Algorithme Girvan-Newman

D'après[29, 30],l'algorithme Girvan-Newman, également connu sous le nom de méthode de coupure de graphe (Graph Clustering), est un algorithme de détection de communautés basé sur la mesure de la modularité. Cet algorithme vise à identifier les arêtes les plus importantes dans le réseau qui, lorsqu'elles sont retirées, provoquent une division significative des communautés existantes.

Voici les étapes de l'algorithme Girvan-Newman :

1. Calcul de la modularité initiale : La modularité est une mesure de la qualité de la division des nœuds en communautés. Dans cette étape, la modularité initiale du réseau est calculée.
2. Calcul des centralités des arêtes : Les centralités des arêtes, telles que la centralité intermédiaire (betweenness centrality), sont calculées pour toutes les arêtes du réseau. La centralité intermédiaire mesure le nombre de fois où une arête est utilisée sur le chemin le plus court entre deux autres nœuds du réseau.
3. Suppression des arêtes centrales : Les arêtes ayant les centralités les plus élevées sont supprimées du réseau. Cette suppression des arêtes vise à créer une division entre les communautés en coupant les liens les plus importants.
4. Calcul de la modularité mise à jour : Après la suppression des arêtes, la modularité du nouveau réseau est recalculée.
5. Répétition des étapes 2 à 4 : Les étapes 2 à 4 sont répétées jusqu'à ce qu'un critère d'arrêt soit atteint. Ce critère peut être le nombre de divisions de communautés souhaité ou la diminution de la modularité en dessous d'un seuil prédéfini.
6. Sélection de la partition optimale : À la fin de l'algorithme, différentes partitions du réseau sont obtenues à partir des étapes de suppression des arêtes. La partition optimale est sélectionnée en fonction de la modularité maximale obtenue.

L'algorithme Girvan-Newman est itératif et peut être coûteux en termes de temps de calcul, en particulier pour les grands réseaux. Cependant, il est largement utilisé car il fournit une approche bottom-up pour la détection de communautés en identifiant les arêtes les plus importantes pour la structure communautaire.

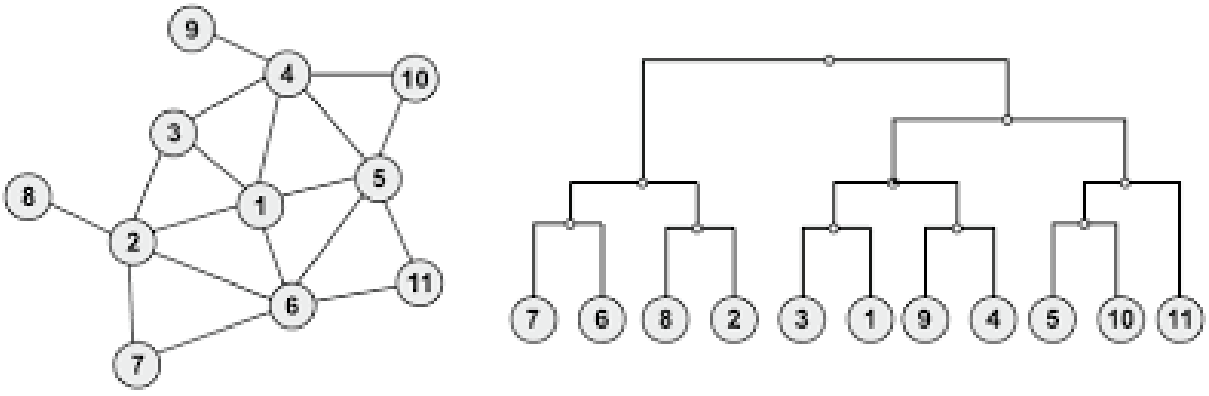

FIGURE 1.10 – Exemple d'un dendrogramme hiérarchique pour Newman[3]

Algorithme Edge Betweenness

L'algorithme edge betweenness (ou "intermédiarité des arêtes" en français) est une méthode de détection de communautés basée sur l'identification des arêtes qui sont les plus importantes pour connecter différents groupes de nœuds dans un réseau. Cet algorithme repose sur le calcul de l'intermédiarité de chaque arête, qui représente le nombre de chemins les plus courts qui passent par cette arête[4].

L'idée de base de l'algorithme est de retirer les arêtes ayant l'intermédiarité la plus élevée de manière itérative jusqu'à ce que le réseau se divise en plusieurs communautés. Les arêtes retirées sont alors considérées comme les "ponts" reliant différentes communautés et les nœuds restants forment les différentes communautés.

L'algorithme edge betweenness est souvent utilisé en combinaison avec d'autres algorithmes de détection de communautés pour améliorer leur efficacité et leur précision. Cependant, cet algorithme est souvent coûteux en termes de calculs et peut être lent sur de grands réseaux.

Algorithme Walktrap

Walktrap est un algorithme de détection de communauté basé sur l'idée que les communautés sont constituées d'ensembles de nœuds qui sont souvent transportés ensemble par des marches aléatoires. Cet algorithme est conçu pour détecter des communautés de taille arbitraire sur des graphiques non pondérés ou pondérés[4]. il fonctionne comme suit :

1. Pour chaque nœud du graphique, des marches aléatoires sont simulées à partir de ce nœud en utilisant une longueur de marche fixe
2. Les marches aléatoires qui s'arrêtent à des nœuds similaires sont combinées en communautés.
3. Les communautés sont fusionnées en supercommunautés jusqu'à ce que le critère de terminaison soit rempli.

Le critère d'arrêt peut être déterminé soit par le nombre de communautés, soit par la différence de modularité entre deux itérations consécutives. La modularité est une mesure de la qualité qui montre comment les nœuds connectés dans une communauté sont comparés à leur connectivité aux nœuds extérieurs à la communauté.

il est efficace pour détecter des communautés de taille arbitraire et peut être appliqué à des graphiques non pondérés ou pondérés. Cependant, il est plus lent que certains autres algorithmes de détection de communauté et peut avoir des difficultés à détecter les communautés avec des liens faibles ou des structures hiérarchiques complexes.

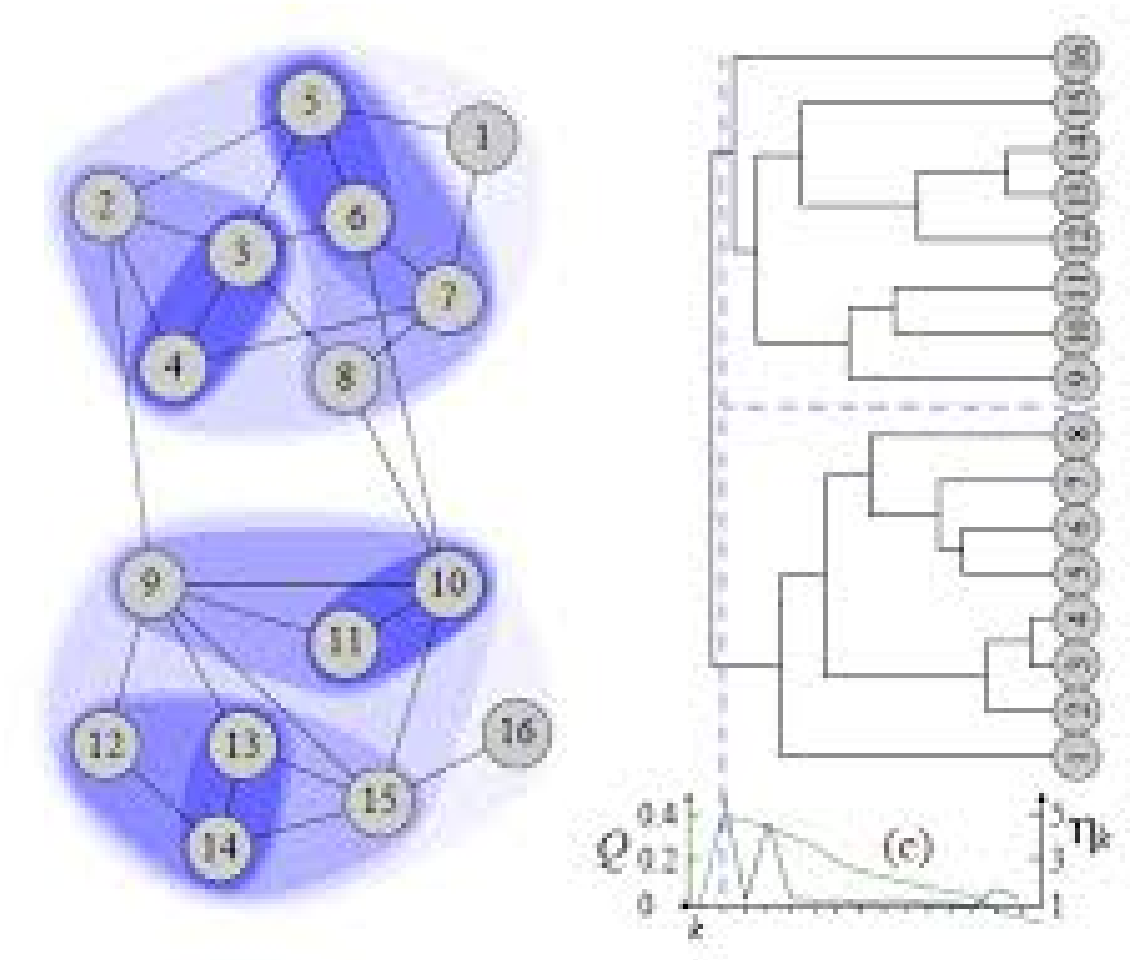

FIGURE 1.11 – Exmple de réseau a 16 sommets divise en 2 communautés avec WalKtrap[4]

Algorithme Fast Greedy

Cet algorithme de détection de communautés basé sur la maximisation de la modularité dans les réseaux complexes. Il a été proposé par Clauset, Newman et Moore en 2004.il fonctionne en calculant la modularité Q pour chaque communauté potentielle formée par la fusion de deux communautés voisines. La fusion qui conduit à la plus grande augmentation de modularité est choisie et répétée jusqu'à ce qu'aucune fusion supplémentaire n'améliore la modularité globale[31].

Algorithme Clique Percolation Method (CPM)

Cet algorithme a été proposée par Palla, Derényi, Farkas et Vicsek en 2005.L'idée principale de l'algorithme CPM est de rechercher des cliques et de les utiliser comme base pour détecter des communautés en fusionnant les cliques voisines. Cela permet une détection de communautés plus flexible et peut révéler des structures de communautés complexes dans les réseaux[32].

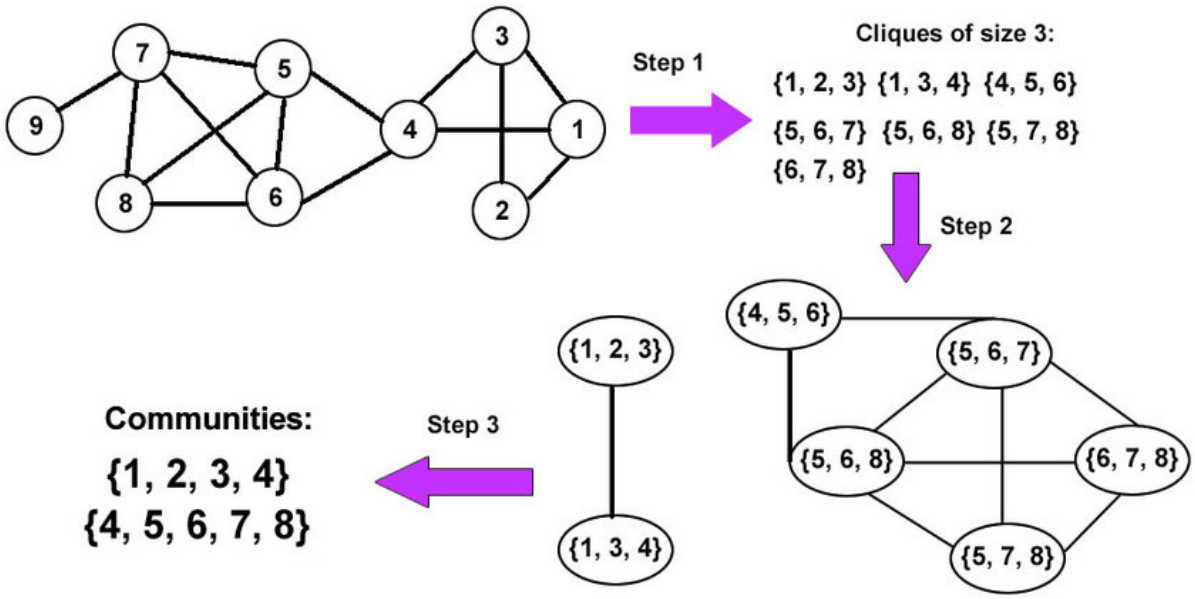

FIGURE 1.12 – Un exemple de l'algorithme de percolation de clink avec k = 3[5]

1.4 Réseaux étudiés

1.4.1 Les Réseaux réels

Les réseaux réels sont des structures interconnectées qui existent dans le monde réel.ils sont utilisé régulièremen comme référence pour tester les algorithmes de détection de communautés. Ces réseaux sont composés d'entités ou de noeuds, qui peuvent représenter des personnes des organisations, des lieux, des objets ou tout autre élément, et de liens ou de connexions qui décrivent les relations entre ces entité.

Voici un tableau qui présente quelques exemples de réseaux réels largement utilisés :

Réseau	Noeuds	Arêtes	Nombre de Communautés
Zachary	34	78	2
Football	115	613	12
Dauphins	62	159	2
Books	105	441	3

TABLE 1.1 – Paramètres des réseaux du monde réel.

Réseaux des dauphins :

Les Réseaux des dauphins[33],également connu sous le nom de "cadolphins", est un exemple classique de réseau réel utilisé dans le domaine de l'étude des réseaux sociaux animaux. Ce réseau représente les interactions sociales entre les dauphins d'une communauté spécifique.ils est composé de 62 nœuds et 159 arêtes. Les nœuds représentent les dauphins individuels et les arêtes représentent les interactions fréquentes entre les dauphins. Chaque arête indique une connexion ou une interaction observée entre deux dauphins.(voir figure 1.13)

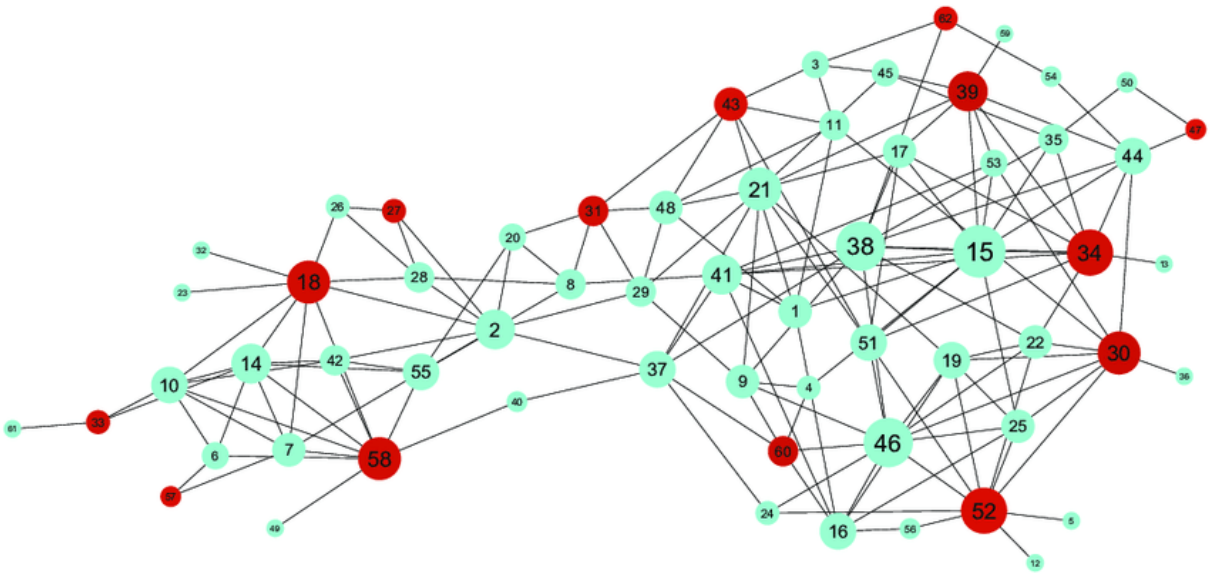

FIGURE 1.13 – le réseau dauphins de Lusseau[6]

Réseaux de Football Américain

Le réseau de football Américain est un exemple de réseau réel largement étudié et utilisé dans le domaine du sport et de l'analyse des réseaux sociaux[34]. Ce réseau représente les interactions entre les joueurs de football et les connexions entre les équipes.

Dans un réseau de football, les nœuds sont les joueurs et les arêtes représentent les

connexions entre ces joueurs. Ces connexions peuvent être basées sur des interactions directes telles que les passes entre les joueurs pendant un match, les relations d'amitié en dehors du terrain, les coéquipiers dans une équipe ou d'autres types de relations sociales.Ce réseau est constitué de douze communautés,115 nœuds et 613 liens.(Voir figure 4.14)

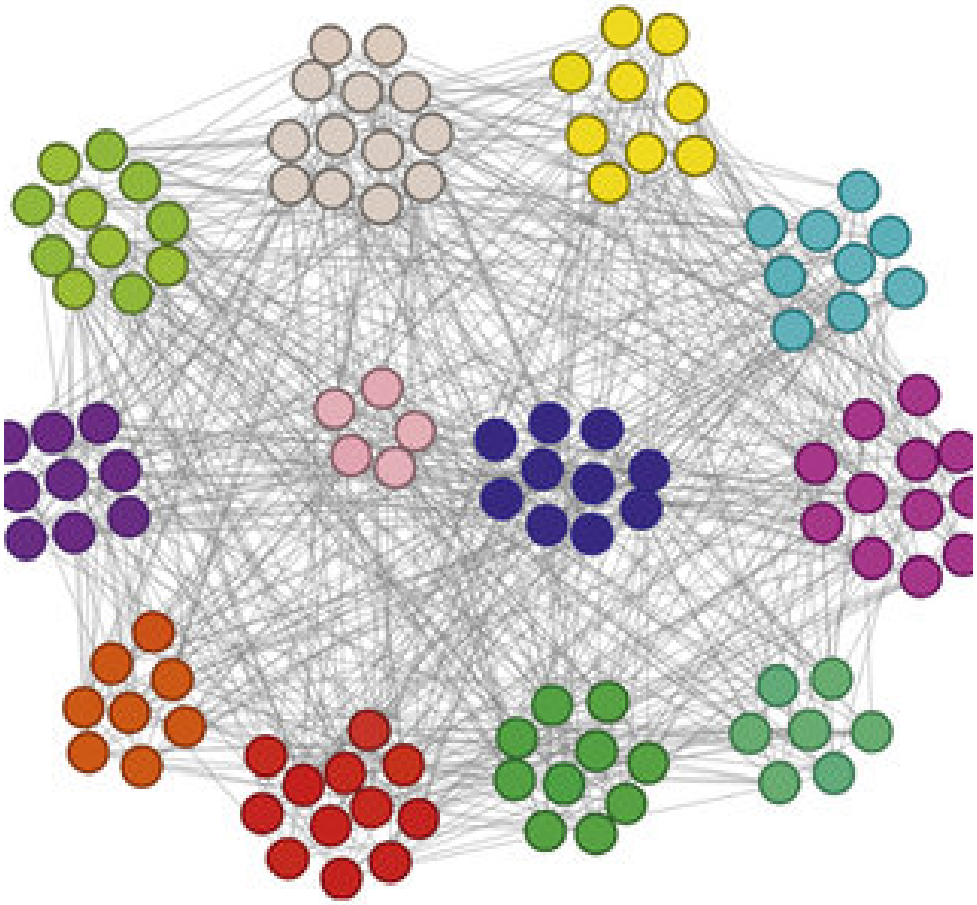

FIGURE 1.14 – La structure communautaire du réseau Football[7]

Réseaux Zachary de club de karaté

Le réseau Zachary de club de karaté est un célèbre exemple de réseau réel qui a été étudié par Wayne W. Zachary dans les années 1970[35].Il est un exemple de réseau réel qui a été étudié dans le domaine de l'analyse des réseaux sociaux.. Il se compose de 34 noeuds, qui représentent les membres d'un club de karaté dans l'université de San Francisco aux Etas Unis, qui ont été observés au cours d'unepériode de trois ans, et 78 arêtes qui représentaient la connexion et l'interaction entre les membres. Ce réseau comporte deux

communautés.

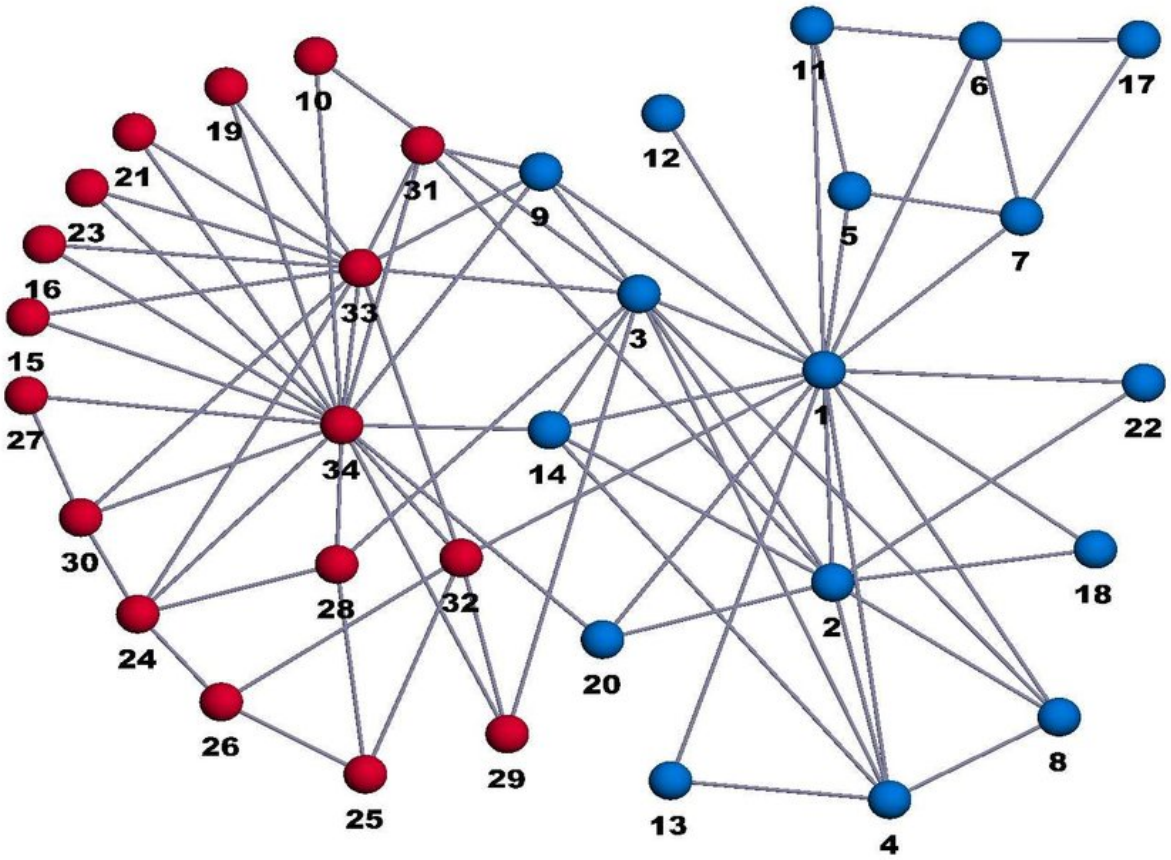

FIGURE 1.15 – Le réseau de club de Karaté de Zachary[8]

Réseau de livres politiques

Dans un réseau de livres politiques, les nœuds représentent les livres individuels et les arêtes représentent les liens ou les connexions entre ces livres. Ces liens peuvent être établis en fonction de divers critères, tels que les références bibliographiques, les citations, les relations conceptuelles ou les similitudes thématiques[36].Ce réseau est constitué de trois communautés,105 nœuds et 441 liens.(Voir figure 4.16)

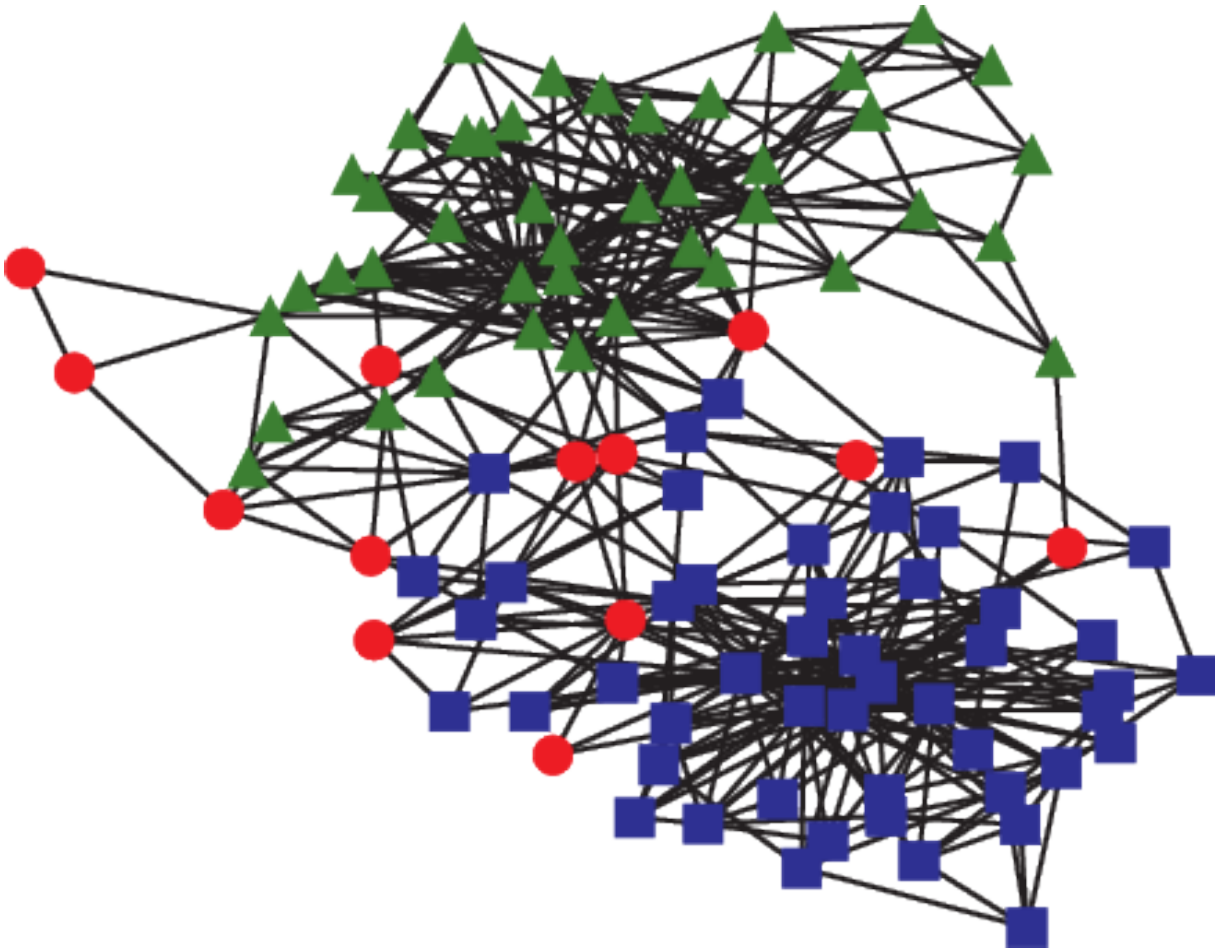

FIGURE 1.16 – Réseau de livres politiques[9]

1.4.2 Les Réseaux synthétiques

Les réseaux artificiels sont des réseaux générés de manière contrôlée et basés sur des modèles et des algorithmes spécifiques. Contrairement aux réseaux réels qui représentent des systèmes existants dans le monde réel, les réseaux artificiels sont conçus pour étudier des phénomènes spécifiques, simuler des interactions ou tester des hypothèses dans un environnement contrôlé. nous allons choisir L'algorithme LFR utilise plusieurs paramètres pour générer un réseau synthétique

Benchmark LFR

Le benchmark LFR (Lancichinetti-Fortunato-Radicchi) est un outil couramment utilisé pour évaluer et comparer les performances des algorithmes de détection de communautés dans les réseaux complexes. Il a été développé par Andrea Lancichinetti, Santo Fortunato

et Filippo Radicchi en 2008[37].

Le benchmark LFR fonctionne en utilisant une approche de génération de réseau basée sur des règles spécifiques. Il prend en compte plusieurs paramètres, tels que le nombre total de nœuds, la distribution de degrés, la distribution de tailles de communautés, la fraction de nœuds intercommunautaires et le degré moyen[38].

L'algorithme de référence LFR a généré le réseau synthétique en suivant ces étapes :

Paramètres d'entrée [37] :

LFR-benchmark-graph(n, tau1, tau2, μ, average-degree, min-degree, max-degree, min-community, max-community, tol=1e-07, max-iters)

1. n : Nombre total de nœuds dans le réseau.
2. tau1 : Exposant de distribution de la taille des communautés.
3. tau2 : Exposant de distribution de la connectivité des nœuds intercommunautaires.
4. μ : Fraction de nœuds intercommunautaires.
5. average-degree : Degré moyen du réseau.
6. min-degree : Degré minimum des nœuds dans le réseau.
7. max-degree : Degré maximum des nœuds dans le réseau.
8. min-community : Taille minimale d'une communauté.
9. max-community : Taille maximale d'une communauté.
10. tol : Tolérance utilisée pour déterminer la fin de l'algorithme.
11. max-iters : Nombre maximal d'itérations pour l'algorithme (facultatif).

Initialisation :

L'algorithme initialise les structures de données nécessaires pour la génération du réseau synthétique.

Génération des communautés :

L'algorithme génère les communautés du réseau en utilisant la distribution de taille des communautés spécifiée par tau1. Il assigne les nœuds aux communautés en respectant les contraintes de taille minimale et maximale.

Génération des nœuds intercommunautaires :

L'algorithme génère les nœuds intercommunautaires en utilisant la distribution de connectivité spécifiée par tau2. Il connecte ces nœuds aux communautés existantes en respectant la fraction µ.

Génération des liens internes à chaque communauté :

L'algorithme génère les liens internes à chaque communauté en fonction du degré moyen spécifié. Il assure que chaque nœud dans une communauté est connecté à un nombre approprié de voisins.

Génération des liens entre les communautés :

L'algorithme génère les liens entre les communautés en fonction du degré moyen et de la fraction de nœuds intercommunautaires. Il assure que chaque nœud intercommunautaire est connecté à un nombre approprié de nœuds d'autres communautés.

Terminaison :

Une fois que les connexions entre les nœuds ont été établies selon les règles spécifiées, l'algorithme se termine et renvoie le réseau synthétique généré.

1.5 Les mésures d'évaluations

Une mésure d'évaluation est une fonction qui évalue la proximité entre deux couvertures d'un même texte[39]. Il existe de nombreuses mesures pour évaluer l'efficacité des algorithmes de détection de communauté. Ces métriques incluent :

1.5.1 La Modularité (Q)

La modularité (Q) est une mesure de l'évaluation de la qualité d'une partition de réseau en communautés. Elle a été introduite par Newman et Girvan dans leur article de 2004[36].

La modularité est définie comme la différence entre le nombre réel d'arêtes (liens) à l'intérieur des communautés et le nombre attendu d'arêtes dans une configuration nulle (où les liens sont répartis au hasard) [3, 40] :

$$Q = \frac{1}{2m} + \sum_{ij} \left(A_{ij} - \frac{k_i \cdot k_j}{2m} \right) \delta(c_i, c_j) \tag{1.2}$$

où :

Aij : Représente la matrice d'adjacence du réseau.

ci : Représente la communauté à laquelle le noeud i est assigné

c : Représente le nombre de communautés.

ki : Représente le degré du nœud i.

kj : C'est le degré du nœud j.

m : C'est le nombre de nombre d'arêtes dans un réseau.

Ainsi, la fonction $\delta(c_i, c_j)$ peut être défini comme suit :

$$\delta(c_i, c_j) = \begin{cases} 1, & \text{si le nœud } i \text{ et le nœud } j \text{ sont dans la même communauté,} \\ 0, & \text{sinon.} \end{cases}$$

La valeur de Q est effectivement comprise entre -1 et +1. Plus la valeur de Q est proche de 1, plus la force de la structure communautaire dans le réseau est élevée, ce qui indique une meilleure qualité de la détection des communautés. Une valeur de Q proche de 1 suggère une forte séparation entre les communautés et une connectivité plus faible entre les communautés[3].

1.5.2 NMI (Normalized Mutual Information)

La Mutual Information (MI), ou information mutuelle en français, est une mesure qui quantifie la dépendance entre deux variables aléatoires. Elle mesure la quantité d'information partagée par ces variables.La NMI est souvent utilisée dans le domaine de l'apprentissage automatique et de la classification, notamment lors de l'évaluation de la similarité entre deux ensembles de clusters ou de partitions[41].

La formule générale pour calculer la NMI est la suivante :

$$\text{NMI}(A, B) = \frac{\text{MI}(A, B)}{\sqrt{H(A) \cdot H(B)}} \tag{1.3}$$

où :

MI(A, B) est la Mutual Information entre les ensembles A et B.

H(A) et H(B) sont les entropies des ensembles A et B respectivement.

La NMI donne une valeur entre 0 et 1, où 0 indique une indépendance complète entre les ensembles et 1 indique une parfaite correspondance entre les ensembles[41].

1.5.3 L'indice de Rand (RI)

L'Indice de Rand (Rand Index en anglais) est une mesure de similarité qui évalue la qualité d'une partition ou d'un regroupement (clustering) en comparant les associations entre les paires d'éléments dans la partition. Il est souvent utilisé pour évaluer la performance d'un algorithme de clustering en comparant ses résultats avec une vérité de terrain (ground truth)[42].

La formule générale pour calculer l'Indice de Rand (RI) est la suivante :

$$RI = \frac{a + b}{a + b + c + d} \tag{1.4}$$

où :

a : représente le nombre de paires d'éléments dans la même classe dans la vérité de terrain et dans la même classe dans la partition générée par l'algorithme de clustering.

b : représente le nombre de paires d'éléments dans des classes différentes dans la vérité de terrain et dans des classes différentes dans la partition générée.

c : représente le nombre de paires d'éléments dans la même classe dans la vérité de terrain et dans des classes différentes dans la partition générée.

d : représente le nombre de paires d'éléments dans des classes différentes dans la vérité de terrain et dans la même classe dans la partition générée.

L'Indice de Rand fournit une valeur entre 0 et 1, où 0 indique une concordance aléatoire entre les partitions et 1 indique une concordance parfaite. Une valeur proche de 1 indique une bonne qualité de la partition générée par rapport à la vérité de terrain.

1.6 Conclusion

Dans ce chapitre, l'accent est mis sur la définition et la reconnaissance des communautés, ainsi que sur certaines définitions de la théorie des graphes. Nous avons également discuté des mesures pour évaluer la détection communautaire, telles que la modularité (Q), la NMI et l'indice de rand (RI).

Dans la suite, nous allons présenter l'importance des nœuds dans un réseau, ainsi qu'une présentation détaillée de la densité que nous allons utiliser dans notre approche.

Chapitre 2

L'importance des nœuds dans un réseau

2.1 Introduction

Lorsqu'il s'agit de détecter des communautés dans les réseaux, différentes approches ont été proposées. Parmi ces approches, il y a celles qui se basent sur l'importance des nœuds dans le réseau. Ces méthodes exploitent le concept selon lequel certains nœuds jouent un rôle central ou influent dans la structure du réseau, ce qui peut être indicatif de leur appartenance à des communautés spécifiques.

Comprendre l'importance des nœuds dans un réseau est essentiel pour analyser et interpréter la structure et le fonctionnement du système étudié. Dans ce chapitre, nous abordons les notions de base concernant l'influence des nœuds dans un réseau. En outre, nous présentons différentes mesures utilisées pour évaluer l'importance des nœuds. Nous achevons ce chapitre par une explication détaillée de la densité, en fournissant des définitions et des formules de calcul pour la détection des communautés.

2.2 Notions de base

Dans un réseau, les nœuds sont des éléments essentiels qui interagissent entre eux pour transmettre des informations, partager des ressources ou prendre des décisions. L'influence

des nœuds dans un réseau dépend de plusieurs facteurs, tels que leur connectivité, leur position et leur capacité à influencer les autres nœuds.

Voici quelques notions de base sur l'influence des nœuds dans un réseau :

2.2.1 Connectivité

La connectivité d'un nœud se réfère à ses liens et à sa capacité à communiquer avec d'autres nœuds du réseau. Les nœuds fortement connectés ont tendance à avoir une plus grande influence, car ils peuvent atteindre et être atteints par un plus grand nombre de nœuds[23].

2.2.2 Centralité

Centralité : La centralité d'un nœud mesure son degré de centralité dans le réseau. Un nœud central est celui qui est souvent emprunté par les autres nœuds pour atteindre différents points du réseau. Les nœuds centraux ont généralement une influence plus importante sur les flux d'informations ou les décisions prises dans le réseau[43].

2.2.3 Puissance

Certains nœuds peuvent avoir plus de puissance ou de ressources que d'autres, ce qui leur confère une influence plus grande[43]. Par exemple, dans un réseau social, les utilisateurs ayant un grand nombre de followers ou une forte influence sociale ont tendance à être des nœuds puissants.

2.2.4 Diffusion d'informations

Dans un réseau, les nœuds peuvent diffuser des informations à leurs voisins ou à d'autres nœuds avec lesquels ils sont connectés. Certains nœuds peuvent avoir une capacité plus élevée à influencer la diffusion d'informations, en raison de leur position ou de leur popularité[44].

2.2.5 Effet de seuil

L'effet de seuil, également connu sous le nom de contagion seuil, se réfère à un phénomène dans lequel l'adoption ou la propagation d'une idée, d'un comportement ou d'une innovation dans un réseau dépend d'un certain seuil de participation ou d'adoption parmi les nœuds du réseau. L'idée centrale de l'effet de seuil est que l'adoption d'une nouvelle idée se propage de manière non linéaire, avec un changement brusque lorsque suffisamment de nœuds adoptent cette idée[45].

2.3 Mesures pour évaluer l'importance des nœuds dans un réseau

2.3.1 Mesures de centralité

Le degré de centralité

Le degré de centralité est une mesure couramment utilisée pour évaluer l'importance des nœuds dans un réseau. Cette mesure se réfère au nombre de liens directs qu'un nœud a avec d'autres nœuds du réseau. Plus un nœud a un degré élevé, plus il est considéré comme central, car il est directement connecté à un grand nombre d'autres nœuds.[43].

La formule mathématique pour calculer la centralité de degré d'un nœud dans un réseau non orienté est la suivante :

$$Cdeg(v) = k(v) \tag{2.1}$$

Cdeg(v) est la centralité de degré du nœud v k(v) est le degré du nœud v, c'est-à-dire le nombre de liens qu'il possède.

Dans le cas d'un réseau orienté, on peut calculer la centralité de degré entrant (in-degree centrality) et la centralité de degré sortant (out-degree centrality). La formule mathématique pour ces mesures est :

$$Cdeg - in(v) = \sum Aij \tag{2.2}$$

$$Cdeg - out(v) = \sum Aij \tag{2.3}$$

Cdeg-in(v) est la centralité de degré entrant du nœud v

Cdeg-out(v) est la centralité de degré sortant du nœud v

Aji représente l'élément à la j-ème ligne et i-ème colonne de la matrice A (pour la centralité de degré entrant)

Aij représente l'élément à la ième ligne et jème colonne de la matrice A (pour la centralité de degré sortant).

Centralité de proximité (Closeness centrality)

La centralité de proximité, également connue sous le nom de closeness centrality en anglais, est une mesure utilisée dans l'analyse des réseaux pour évaluer l'importance d'un nœud en fonction de sa proximité avec tous les autres nœuds du réseau. Cette mesure quantifie la distance moyenne entre un nœud donné et tous les autres nœuds du réseau[43, 46].

La formule mathématique de la centralité de proximité pour un nœud spécifique est la suivante :

$$C(x) = \frac{1}{\sum_y d(x, y)} \tag{2.4}$$

C(x) représente la centralité de proximité du nœud x.

d(x, y) représente la distance entre les nœuds x et y.

La somme est effectuée sur tous les nœuds y du réseau, à l'exception du nœud x.

Centralité d'intermédiarité (Betweenness centrality)

La centralité d'intermédiarité (betweenness centrality en anglais) est une mesure utilisée dans l'analyse des réseaux pour évaluer l'importance d'un nœud en fonction du nombre de fois où il se trouve sur le chemin le plus court entre deux autres nœuds du réseau. Cette mesure quantifie la capacité d'un nœud à faire le lien entre différents sous-ensembles du réseau[46].

La formule mathématique de la centralité d'intermédiarité pour un nœud spécifique est la suivante :

$$C(x) = \sum_{s \neq x \neq t} \frac{\sigma_{st}(x)}{\sigma_{st}} \tag{2.5}$$

C(x) représente la centralité d'intermédiarité du nœud x.

σ_{st}représente le nombre total de chemins les plus courts entre les nœuds

$\sigma_{st}(x)$représente le nombre de ces chemins qui passent par le nœud

La somme est effectuée sur tous les nœuds s et t du réseau, à l'exception du nœud x.

2.3.2 Autres mesures de centralité

Centralité de PageRank (PageRank centrality)

La centralité de PageRank est une mesure utilisée dans l'analyse des réseaux pour évaluer l'importance d'un nœud en se basant sur la structure globale du réseau. Elle a été développée par Larry Page et Sergey Brin, les fondateurs de Google, pour évaluer la pertinence des pages Web dans les résultats de recherche[47].

La formule mathématique de la centralité de PageRank pour un nœud spécifique est la suivante :

$$PR(x) = (1 - d) + d * \sum_{y \in B(x)} \frac{PR(y)}{L(y)} \tag{2.6}$$

PR(x) représente la centralité de PageRank du nœud x.

d est un facteur d'amortissement entre 0 et 1, généralement défini à 0,85.

B(x) représente l'ensemble des nœuds qui ont un lien entrant vers x.

PR(y) représente la centralité de PageRank du nœud y.

L(y) représente le nombre de liens sortants du nœud y.

Centralité de vecteur propre (Eigenvector centrality)

La centralité de vecteur propre est une mesure utilisée dans l'analyse des réseaux pour évaluer l'importance d'un nœud en se basant sur les liens vers les autres nœuds du réseau.

Elle attribue une valeur de centralité à chaque nœud en fonction de la centralité des nœuds connectés à celui-ci[43].

La formule mathématique de la centralité de vecteur propre pour un nœud spécifique est la suivante :

$$C(x) = \frac{1}{\lambda_1} \sum_{y} A(x, y)C(y) \tag{2.7}$$

C(x) représente la centralité de vecteur propre du nœud x.

λ est la valeur propre dominante de la matrice d'adjacence.

A(x, y) est l'élément de la matrice d'adjacence qui indique la présence ou l'absence de lien entre les nœuds x et y.

La somme est effectuée sur tous les nœuds y du réseau.

2.4 la densité d'un noeuds dans un réseau

2.4.1 Définitions

La densité des nœuds est un concept qui évalue l'intensité des liens et la connectivité d'un nœud spécifique dans un réseau. Elle est utilisée pour quantifier à quel point un nœud est densément connecté à ses voisins ou à d'autres nœuds du réseau. Effectivement, la densité des nœuds peut être définie de différentes manières selon le contexte du réseau étudié[48].

Il existe plusieurs approches courantes pour définir la densité des nœuds dans un réseau. Voici quelques-unes d'entre elles :

2.4.2 La densité locale

La densité locale, également connue sous le nom de densité des nœuds ou densité locale des arêtes, est une mesure qui évalue la densité des liens entre les nœuds directement connectés à un nœud spécifique dans un réseau[49, 15]

La densité locale peut être calculée différemment dans un graphe orienté et non orienté.Dans un graphe non orienté, on utilise le terme "arêtes" pour décrire les connexions entre les

nœuds, tandis que dans un graphe orienté, on utilise le terme "arcs" pour représenter les connexions directionnelles[15].

Voici les formules correspondantes :

Graphe non orienté

$$\text{Densité local} = \frac{\text{E'}}{\text{E}} \tag{2.8}$$

E' : Nombre d'arêtes réelles entre les voisins du nœud.

E : Nombre total d'arêtes possibles entre les voisins du nœud.

$$\text{E} = \frac{(\text{nombre de voisins} * (\text{nombre de voisins - 1}))}{2}. \tag{2.9}$$

Graphe orienté

$$\text{Densité local} = \frac{\text{A'}}{\text{A}} \tag{2.10}$$

A' : Nombre d'arcs possibles entre les voisins du nœud.

A : Nombre total d'arcs possibles entre les voisins du nœud.

$$\text{A} = (\text{nombre de voisins}) * (\text{nombre de voisins-1}). \tag{2.11}$$

Ces formules permettent de calculer la densité locale dans le contexte spécifique d'un graphe orienté ou non orienté, en évaluant la densité des connexions entre les voisins d'un nœud donné.

La densité locale est une mesure importante pour comprendre la structure locale d'un réseau et peut être utilisée pour identifier des groupes ou des clusters de nœuds densément connectés dans un voisinage restreint[49].

2.4.3 La densité globale

La densité globale d'un réseau est une mesure qui évalue le degré de connectivité et de densité des liens à l'échelle du réseau dans son ensemble. Elle quantifie la proportion d'arêtes ou d'arcs réels par rapport au nombre maximal d'arêtes ou d'arcs possibles dans le réseau[49, 15].

La formule pour calculer la densité globale d'un graphe non orienté est la suivante :

$$\text{Densité globale} = \frac{\text{Nombre d'arêtes réelles dans le graphe}}{\text{Nombre total d'arêtes possibles dans le graphe}} \tag{2.12}$$

Le nombre d'arêtes réelles correspond au nombre d'arêtes effectivement présentes dans le graphe, tandis que le nombre total d'arêtes possibles dépend du nombre de nœuds dans le graphe. Dans un graphe non orienté, chaque paire de nœuds distincts a une seule arête possible.

Pour un graphe orienté, la formule de densité globale est légèrement différente :

$$\text{Densité globale} = \frac{\text{Nombre d'arcs réels dans le graphe}}{\text{Nombre total d'arcs possibles dans le graphe}} \tag{2.13}$$

Le nombre d'arcs réels représente le nombre d'arcs effectivement présents dans le graphe, tandis que le nombre total d'arcs possibles dépend du nombre de nœuds dans le graphe. Dans un graphe orienté, chaque paire de nœuds distincts peut avoir deux arcs possibles (dans les deux sens).

La densité globale fournit une mesure de la densité générale des liens dans le réseau. Une densité globale élevée indique que le réseau est densément connecté, tandis qu'une densité globale faible indique que le réseau est plus dispersé avec moins de liens.

2.4.4 Différence entre densité locale et densités globale

En effet, la densité globale est une mesure qui évalue la proportion de liens présents dans l'ensemble du réseau par rapport au nombre total de liens possibles. Elle permet de savoir dans quelle mesure le réseau est connecté de manière globale. Plus la densité globale est élevée, plus les nœuds sont liés les uns aux autres et plus le réseau est dense.

D'un autre côté, la densité locale est une mesure qui évalue la proportion de liens présents autour d'un nœud spécifique par rapport au nombre total de liens possibles entre ce nœud et ses voisins immédiats. Elle permet de savoir dans quelle mesure un nœud est connecté à ses voisins. Plus la densité locale est élevée, plus le nœud est connecté à ses voisins et plus il est dense localement.

2.4.5 Utilisation de la densité pour identifier les nœuds importants dans la détection des communautés

la densité peut être utilisée pour identifier les nœuds importants dans la détection des communautés. Les nœuds situés dans des régions de densité élevée peuvent jouer un rôle clé dans la formation et le maintien des communautés en agissant comme des ponts entre les différentes parties du réseau[37]. Nous présenterons des techniques telles que la centralité de proximité ou la centralité d'intermédiarité, qui utilisent la densité pour évaluer l'importance des nœuds dans la structure communautaire.

Voici quelques approches courantes pour utiliser la densité dans l'identification des nœuds importants :

Centralité de proximité basée sur la densité :

La centralité de proximité mesure la distance moyenne entre un nœud donné et tous les autres nœuds du réseau. Dans le contexte de la détection des communautés, la centralité de proximité peut être calculée en tenant compte de la densité. Les nœuds ayant une centralité de proximité élevée dans les régions densément connectées sont considérés comme importants pour la formation et le maintien des communautés[50].

Voici un exemple illustrant l'utilisation de la centralité de proximité basée sur la densité dans la détection des communautés :

Supposons que nous ayons un réseau social où les utilisateurs sont représentés par des nœuds et les amitiés entre les utilisateurs sont représentées par des liens. Nous voulons détecter les communautés dans ce réseau en identifiant les nœuds importants en termes de centralité de proximité basée sur la densité.

Pour calculer la centralité de proximité basée sur la densité, nous suivons les étapes suivantes :

1. Calculer la densité locale pour chaque nœud : La densité locale d'un nœud est définie comme le rapport entre le nombre de liens réellement présents entre ses voisins et le nombre total de liens possibles entre ces voisins. Cela mesure à quel point les voisins d'un nœud sont interconnectés.
2. Identifier les nœuds importants : Les nœuds ayant une centralité de proximité basée sur la densité élevée sont considérés comme importants pour la détection des communautés. Ces nœuds se trouvent généralement dans les régions densément connectées du réseau et jouent un rôle clé dans la structure des communautés.

Par exemple, si nous avons un nœud A avec des voisins B, C et D, et que la densité locale de B, C et D est élevée, alors le nœud A aura une centralité de proximité basée sur la densité élevée. Cela indique que le nœud A est situé dans une région densément connectée et est important pour la formation et le maintien des communautés.

Centralité d'intermédiarité basée sur la densité :

La centralité d'intermédiarité mesure le degré de contrôle ou d'influence qu'un nœud exerce sur la communication entre les autres nœuds du réseau. Dans les régions de densité élevée, les nœuds qui agissent en tant que passerelles ou ponts entre différentes parties du réseau ont souvent une centralité d'intermédiarité élevée. Ces nœuds jouent un rôle important dans la connectivité entre les communautés et sont donc considérés comme importants pour la détection des communautés[51].

Voici un exemple concret de l'utilisation de la centralité d'intermédiarité basée sur la densité dans la détection des communautés :

Supposons que nous ayons un réseau de collaborations scientifiques, où les chercheurs sont représentés par des nœuds et les co-auteurs de publications communes sont représentés par des liens. Nous voulons détecter les communautés de chercheurs dans ce réseau en identifiant les nœuds importants en termes de centralité d'intermédiarité basée sur la densité.

Dans ce cas, les nœuds avec une intermédiarité élevée dans les régions de densité élevée peuvent jouer un rôle essentiel dans la communication entre différentes communautés de

chercheurs.

Voici comment l'approche peut être appliquée :

1. Calcul de la densité locale : Pour chaque nœud, nous calculons la densité locale en mesurant le nombre de co-auteurs communs entre ses voisins.
2. Calcul de la centralité d'intermédiarité basée sur la densité : Pour chaque nœud, nous calculons la centralité d'intermédiarité en prenant en compte la densité locale de ses voisins. Les nœuds situés dans des régions de densité élevée avec une intermédiarité élevée seront considérés comme importants pour la détection des communautés.
3. Identification des nœuds importants : Les nœuds ayant une centralité d'intermédiarité basée sur la densité élevée sont considérés comme importants dans la communication entre les différentes communautés de chercheurs. Ces nœuds peuvent servir de ponts ou de passerelles entre les communautés et faciliter l'échange d'informations.

En utilisant cette approche, nous pouvons identifier les chercheurs qui jouent un rôle clé dans la communication et la collaboration entre les différentes communautés scientifiques. Ces nœuds importants peuvent être utilisés pour mieux comprendre la structure du réseau de collaborations et faciliter l'identification des communautés scientifiques.

Méthodes de partitionnement basées sur la densité :

Certaines méthodes de partitionnement, telles que la méthode de Louvain ou la méthode de détection de communautés basée sur le label propagation, utilisent la densité du réseau pour déterminer les partitions de communautés. Ces méthodes cherchent à maximiser la densité interne des communautés tout en minimisant la densité entre les communautés. Les nœuds qui se trouvent à l'intérieur des communautés densément connectées sont considérés comme importants pour la détection des communautés[52, 53].

Voici un exemples concrets de méthodes de partitionnement basées sur la densité pour la détection des communautés dans les réseaux :

Méthode de détection de communautés basée sur le label propagation :

Cette méthode repose sur la propagation de l'information à travers les liens du réseau pour identifier les communautés. Elle utilise la densité des liens pour déterminer les étiquettes des nœuds. Au départ, chaque nœud est attribué à une étiquette unique. Ensuite, l'information se propage à travers les liens du réseau, et les nœuds ajustent leurs étiquettes en fonction des étiquettes de leurs voisins. Ce processus de propagation itère jusqu'à ce que les étiquettes convergent vers une configuration stable. Les nœuds avec des étiquettes similaires sont regroupés dans la même communauté de densité élevée[53].

2.4.6 Impact de la densité sur l'importance des nœuds dans les réseaux

Une densité élevée dans un réseau peut avoir un impact significatif sur la résilience, l'efficacité et la diffusion de l'information au sein des communautés[54].

Voici quelques points importants à considérer :

Propagation rapide de l'information :

Dans un réseau dense, la proximité et la connectivité élevée entre les nœuds favorisent la propagation rapide de l'information. Lorsqu'un nœud reçoit une information, il peut la transmettre rapidement à ses voisins qui, à leur tour, la diffusent à d'autres nœuds. Cette propagation rapide facilite la transmission efficace de l'information au sein des communautés, ce qui peut conduire à une meilleure coordination et à une prise de décision plus rapide.

Résilience aux défaillances :

La densité élevée dans un réseau peut renforcer sa résilience face aux défaillances. Étant donné qu'il existe de nombreux liens entre les nœuds, la défaillance d'un seul nœud ou d'un seul lien a moins d'impact sur la connectivité globale du réseau. Les nœuds adjacents peuvent prendre en charge la transmission de l'information et maintenir le flux de communication, minimisant ainsi les interruptions potentielles. Cela rend le réseau plus résistant aux défaillances locales.

Résilience aux attaques ciblées :

Une densité élevée peut également renforcer la résilience du réseau face aux attaques ciblées. En raison de la connectivité dense, les attaques visant un nœud ou un groupe de nœuds spécifiques peuvent avoir un impact limité sur l'ensemble du réseau. Les chemins de communication alternatifs et la redondance des liens permettent de contourner les zones affectées, limitant ainsi la propagation des dommages. Cela rend plus difficile pour un attaquant de perturber efficacement le réseau.

La densité d'un réseau mesure l'importance d'un nœud en termes de connections. Plus précisément, elle indique le nombre de connexions qu'un nœud a avec d'autres nœuds dans le réseau. La densité d'un nœud peut être calculée en comparant le nombre de connexions qu'il possède par rapport au nombre maximum de connexions possibles dans le réseau.La densité d'un nœud peut être utilisée pour évaluer son rôle et son influence dans le réseau. Les nœuds avec une densité élevée sont souvent considérés comme centraux ou essentiels, car ils sont fortement connectés et peuvent jouer un rôle clé dans la diffusion d'informations ou la transmission de flux dans le réseau[55].

2.5 la densité des nœuds pour une détection des communautés dans un réseau

D'après une récente étude [39],La densité des nœuds est une mesure importante utilisée dans la détection des communautés dans un réseau. Elle fait référence au nombre de connexions existantes entre les nœuds d'un réseau, exprimé en termes de densité de liens. L'analyse de la densité des nœuds peut offrir plusieurs avantages pour une détection efficace des communautés.

Voici les étapes générales pour utiliser la densité des nœuds dans la détection des communautés[56, 57] :

Construction du réseau :

Convertissez vos données en une représentation de réseau appropriée, comme une matrice d'adjacence ou une liste d'arêtes. Cette représentation permet de capturer les relations entre les nœuds du réseau.

Calcul de la densité des nœuds :

Pour chaque nœud du réseau, calculez sa densité en mesurant le nombre de connexions qu'il a avec ses voisins. Vous pouvez définir la densité en fonction du nombre de liens ou de la pondération des liens, en fonction des caractéristiques spécifiques de votre réseau.

Identification des communautés :

Utilisez des algorithmes de détection des communautés qui intègrent la densité des nœuds. Certains algorithmes populaires incluent l'algorithme de Louvain, l'algorithme de détection de communautés basé sur la propagation de l'étiquette (Label Propagation), ou encore les méthodes basées sur la modularité. Ces algorithmes cherchent à maximiser la densité intra-communautaire tout en minimisant la densité inter-communautaire.

Évaluation des résultats :

Évaluez la qualité des communautés détectées en utilisant des mesures d'évaluation telles que la modularité, la conductance, ou d'autres mesures spécifiques à votre cas d'utilisation. Ces mesures vous aideront à évaluer la cohésion interne des communautés et leur séparation les unes des autres.

Il est important de noter que la détection des communautés dans un réseau peut être un problème complexe et qu'il existe plusieurs approches et algorithmes disponibles. Les étapes mentionnées ci-dessus fournissent une vue d'ensemble générale, mais il peut être nécessaire d'adapter ces étapes en fonction des caractéristiques spécifiques de votre réseau et des objectifs de votre étude.

2.5.1 Avantages de la densité pour la détection des communautés

On peut citer les avantages suivants[58]

- La densité des nœuds met en évidence les zones du réseau où les nœuds sont fortement connectés entre eux, facilitant ainsi la détection des communautés denses.
- Elle permet de détecter des structures locales ou des clusters au sein du réseau, révélant des sous-groupes de nœuds étroitement liés qui peuvent former des communautés distinctes.
- Le calcul de la densité des nœuds est généralement simple et efficace, surtout pour les réseaux de petite à moyenne taille.
- La densité des nœuds peut être adaptée pour détecter des communautés chevauchantes, c'est-à-dire des nœuds qui appartiennent à plusieurs communautés. En identifiant les zones de densité intermédiaire, la densité peut révéler des régions où les communautés se chevauchent, ce qui peut être pertinent dans certains types de réseaux.
- elle permet de détecter des sous-communautés au sein de communautés plus larges. En identifiant les zones denses à l'intérieur des communautés, la densité peut révéler des sous-structures et des hiérarchies communautaires.

2.5.2 Inconvénients de la densité pour la détection des communautés

Parmi les inconvénients majeurs de la densité, on peut citer les points suivants[58, 59, 60]

- La densité des nœuds se concentre sur les liens forts et néglige souvent les liens faibles, ce qui peut entraîner une négligence des communautés liées principalement par des liens faibles.
- Elle peut être influencée par les nœuds atypiques ou les outliers, ce qui peut introduire des biais dans la détection des communautés.
- La densité des nœuds peut être sensible à la résolution choisie pour la détection des communautés, ce qui peut conduire à des résultats de détection différents selon la résolution.

— Elle peut dépendre d'un seuil de densité défini pour distinguer les communautés, ce qui peut être subjectif et influencer les résultats de détection.
— Dans de grands réseaux, l'interprétation de la densité peut être plus difficile en raison du grand nombre de connexions possibles.

2.6 La détection des communautés fonctionne avec la densité

La méthode de détection de communautés basée sur la densité repose sur l'idée que les nœuds d'une communauté sont densément connectés entre eux, formant ainsi des régions de densité élevée dans le graphe. Cette approche permet de détecter des communautés de forme arbitraire, qui ne sont pas nécessairement connexes ou bien séparées par des frontières claires.

Bien que le problème de la détection de communauté dans les réseaux soit un sujet relativement nouveau, il a rapidement donné lieu à un large corpus de travaux, dans lesquels nous citons :

Martin Ester, Hans-Peter Kriegel, Jörg Sander et Xiaowei Xu[14] ont proposés une nouvelle approche de détection de communauté basée sur la densité,appelée DBSCAN (Density-Based Spatial Clustering of Applications with Noise en anglais). L'idée principale de DBSCAN est de partitionner les points d'un ensemble de données en plusieurs clusters en fonction de la densité de leurs voisinages. L'algorithme ne requiert pas de connaître le nombre de clusters à l'avance et peut identifier des clusters de formes arbitraires.

Le fonctionnement de DBSCAN est le suivant :

— L'algorithme commence par sélectionner un point de données non visité et explore son voisinage dans un rayon spécifié (ϵ). Si le nombre de points dans le voisinage est supérieur ou égal à un seuil (MinPts), alors le point est considéré comme un noyau (core point).
— Une fois qu'un noyau est identifié, l'algorithme étend le cluster en ajoutant les points atteignables à partir de ce noyau. Un point est considéré comme atteignable à partir d'un noyau s'il se trouve dans le voisinage ϵ de ce noyau ou s'il peut être atteint par une chaîne de points voisins.

— L'algorithme répète les étapes précédentes pour chaque point atteignable, formant ainsi un cluster.
— Les points qui ne peuvent pas être atteints à partir d'un noyau sont considérés comme des points de bruit et ne sont pas inclus dans un cluster.

L'algorithme DBSCAN est sensible aux paramètres ϵ et MinPts. Le paramètre définit la portée du voisinage à explorer, tandis que MinPts détermine le seuil minimum de points requis pour qu'un point soit considéré comme un noyau. Le choix de ces paramètres dépend de la structure et de la densité des données.

DBSCAN offre plusieurs avantages par rapport à d'autres algorithmes de clustering. Il peut gérer des ensembles de données de taille variable et est capable de détecter des clusters de forme arbitraire. De plus, il est moins sensible aux valeurs aberrantes (outliers) que certains autres algorithmes de clustering.

En 1999 par Michael Ankerquist, Markus M. Brünnig, Hans-Peter Kriegel et Jörg Sander[61] ont proposé un algorithme amélioré qui est l'algorithme OPTICS (Ordering Points To Identify the Clustering Structure).il est un algorithme de clustering basé sur la densité, qui étend et améliore l'algorithme DBSCAN.

L'objectif principal d'OPTICS est d'identifier la structure de clustering d'un ensemble de données en prenant en compte la densité des points. L'algorithme attribue à chaque point un certain niveau d'accessibilité, qui mesure à quel point le point est atteignable depuis un autre point voisin dans l'espace des données. En utilisant ces niveaux d'accessibilité, l'algorithme génère un diagramme OPTICS (OPTICS reachability plot), qui révèle la structure de densité dans les données.

Tout d'abord,OPTICS commence par sélectionner un point de données non visité et explore son voisinage dans un rayon spécifié (ϵ),comme DBSCAN.Ensuite,L'algorithme calcule ensuite la distance de chaque point voisin par rapport au point actuel et attribue à chaque point un niveau d'accessibilité. Le niveau d'accessibilité est défini comme la distance maximale entre le point actuel et tous les points voisins déjà visités.En outre,il utilise ces niveaux d'accessibilité pour construire un diagramme OPTICS, également appelé le diagramme reachability plot. Ce diagramme représente les points dans l'ordre de leur accessibilité croissante. Les points denses sont représentés par des zones plates dans le diagramme, tandis que les transitions abruptes indiquent les frontières entre les clus-

terse.Enfin,en analysant le diagramme OPTICS, on peut identifier les clusters en fonction de la structure de densité. Les zones plates dans le diagramme correspondent aux points denses qui forment les clusters, tandis que les transitions abruptes indiquent les frontières entre les clusters.

OPTICS offre plusieurs avantages par rapport à DBSCAN. Il est capable de détecter des clusters de forme arbitraire et de trouver les points de bruit plus efficacement. De plus, le diagramme OPTICS fournit une visualisation de la structure de densité des données, ce qui peut faciliter l'interprétation des résultats.

Cependant, OPTICS peut être plus coûteux en termes de temps de calcul par rapport à DBSCAN, car il nécessite de calculer les niveaux d'accessibilité pour chaque point. De plus, l'interprétation du diagramme OPTICS peut parfois être subjective et nécessite une certaine expertise.

Le travail de Peter Hartigan et Manchek Wong[62]ont proposés un algorithme de clustering basé sur la densité qui vise à détecter des clusters en utilisant des estimations locales de densité,appelée DENCLUE (DENsity-based CLUstEring). L'idée principale de DENCLUE est de modéliser la densité des données à l'aide de fonctions de densité et d'identifier les régions de densité élevée pour former des clusters. Le fonctionnement de DENCLUE est le suivant :

- Estimation de densité : DENCLUE utilise des fonctions de densité, telles que la fonction de noyau gaussien, pour estimer la densité locale autour de chaque point de données. Ces estimations de densité sont utilisées pour identifier les modes locaux.
- Attraction de densité : À partir des estimations de densité, l'algorithme calcule la force d'attraction de densité pour chaque point, en prenant en compte les contributions des points voisins et les estimations locales de densité. La force d'attraction de densité mesure à quel point un point est attiré vers les régions de densité élevée.
- Formation de clusters : Les points dont la force d'attraction de densité dépasse un certain seuil sont regroupés pour former des clusters. L'algorithme tient également compte de la proximité spatiale entre les points lors de la formation des clusters.

DENCLUE est un algorithme de clustering basé sur la densité qui utilise des estimations locales de densité pour détecter des clusters. Il offre une approche flexible et capable de détecter des clusters de forme complexe. Cependant, il peut nécessiter une attention particulière dans le choix des paramètres et être coûteux en termes de calcul pour les

ensembles de données volumineux.

L'inconvénient de l'approche proposée est L'estimation de densité dans DENCLUE peut être coûteuse en termes de calcul, en particulier pour les ensembles de données volumineux. Le traitement de grandes quantités de données peut nécessiter des ressources computationnelles importantes et augmenter considérablement le temps d'exécution de l'algorithme.

2.7 Conclusion

Dans l'ensemble, ce chapitre donne un aperçu des concepts de base de l'influence et de l'importance des nœuds dans les réseaux. Sur la base des nœuds importants et du travail de découverte communautaire associé, nous détaillons les densités utilisées pour quantifier la quantité de nœuds que nous utilisons dans notre approche.

Le chapitre suivant est consacré à la conception de l'approche proposée.

Chapitre 3

Méthodologie

3.1 Introduction

La détection de communautés dans les graphes est une problématique fondamentale dans le domaine de l'analyse des réseaux.

Dans ce contexte, cette étude propose une méthode novatrice de détection de communautés, axée sur l'utilisation de la densité locale a densité avec une itération et une fusion progressive des communautés en fonction de critére de similarité nous présentons d'abord les objectifs de notre système, ensuite,nous décrivons l'architecture du système, les étapes de notre conception,Cette méthode vise également à résoudre le problème des nœuds chevauchants au sein des communautés, en mettant en œuvre une étape de regroupment.

La section suivante détaillera la méthodologie mise en œuvre, nous présentons d'abord les objectifs de notre système, ensuite, nous décrivons l'architecture du système, les étapes de notre conception, ainsi que les algorithmes proposés, nous achevons ce chapitre par une conclusion.

3.2 Problématique

La détection des communautés au sein des réseaux est un sujet d'étude crucial dans de nombreux domaines tels que la sociologie, la biologie des réseaux, et l'analyse des médias sociaux. Cependant, les approches traditionnelles de détection des communautés négligent

souvent l'importance des nœuds, ce qui peut entraîner des résultats moins précis et une compréhension limitée de la structure du réseau. Par conséquent, la problématique principale qui se pose est de savoir comment améliorer la détection des communautés en prenant en compte l'importance des nœuds ? Comment pouvons-nous améliorer la précision de ces algorithmes de détection tout en gérant les réseaux de plus en plus volumineux et complexes qui émergent constamment dans notre monde interconnecté ? Comment pouvons-nous relever le défi de la détection des communautés dans les réseaux qui présentent des chevauchements, c'est-à-dire des nœuds appartenant à plusieurs communautés simultanément ?

3.3 Objectif

L'objectif principal de ce travail peut être de permettre aux utilisateurs de détecter et d'analyser les communautés disjointes dans un réseau ou un graphe donné basant sur l'importance des noueds, En identifiant les nœuds centraux et influents à l'aide la densité qui nous aide à choisir les bonnes communautés pour les nœuds

afin d'atteindre les principaux objectifs :

- Détection efficace et rapide des communautés.
- Faire une méthode simple, facile à implémenter , et applicable aux grands réseaux.
- faire le regroupment des communautés similaires
- fournir une visualisation du réseau avec les communautés identifiées, ce qui permet une meilleure compréhension et une analyse visuelle des structures de communautés.

3.4 Architcture du systeme

Notre application se compose de trois parties principales :

1. Générer et afficher des graphes à partir de différentes sources de données.
2. Calculer la densité locale de chaque nœud dans un graphe et triez les nœuds en ordre décroissant de densité.
3. Explourer les commnautés similaires et faire le regroupment

L'architecture globale de notre système est présentée dans le schéma suivant (voir la figure 4.1) :

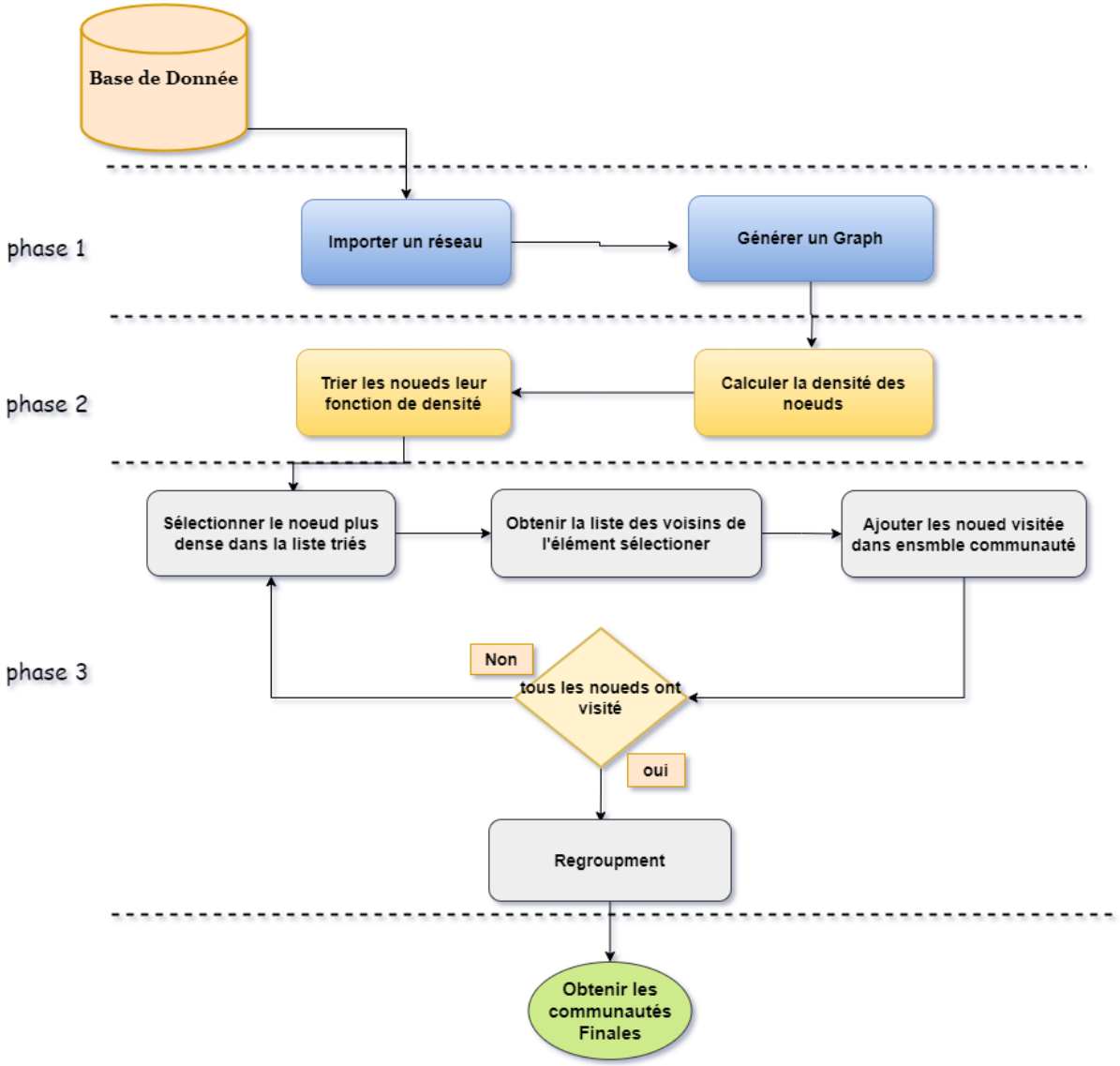

FIGURE 3.1 – Architecture générale du système.

3.5 Conception détaillée de l'approche proposée

L'approche proposée consiste à analyser des graphes à partir de différentes sources de données et à appliquer des techniques de détection de communautés pour identifier des sous-groupes de nœuds fortement connectés.

Nous commençons par extraire les données pertinentes à partir des différentes sources de données,ensuite Calculer la densité de nœuds et trier par ordre décroissant. A la fin en

explorant les noeuds voisins qui renvoi une liste des commnautés similaires aprés nous faire une regroupmment, ce qui améliore l'efficacité de la détection des commnautés.

L'approche proposé se déroule en trois phases :

3.5.1 La première phase

Cette phase consiste à importer une base de données réelle à la convertir en graphe.

3.5.2 La deuxième phase

Cette phase déroule en deux étapes :

1. Calculer la densité local des noeuds selon equation 2.8.
2. Classer les noeuds du réseau par ordre décroissant du la densité .

3.5.3 La troisième phase

Cette phase se déroule en trois étapes :

1. Sélectionner le premier élément de la liste des nœuds triés par densité.
2. Explourer les voisins des nœuds jusqu'à ce que tous les nœuds aient été visités et stocker dans une liste.

3. **Regroupment**

   Dans cette etape,on va rgrouper les communautés similaires basé sur la notion de seuil de similarité entre deux communautés en fonction du chevauchement entre les communautés.Voici les conditions spécifiques pour chaque regroupment dans notre fusion :

   — **Regroupment 1 :**

   Comparer toutes les paires possibles de communautés dans la liste des communautés : Si seuil(deux communautés) <= 1 Les communautés sont fusionnées en ajoutant les éléments de la deuxième communauté à la première, puis à supprimer les doublant et eliminer la liste vide.

Obtenir une liste chevauchement contenant les éléments répétés uniques parmi les membres de toutes les communautés.

— **Regroupment 2 :**

Si la liste de chevauchement contient plus d'un élément

Comparer toutes les paires possibles de communautés dans la liste des communautés : Si seuil(deux communautés) <= 3 Les communautés sont fusionnées de la même manière que dans la première itération.

Obtenir une liste chevauchement contenant les éléments répétés uniques parmi les membres de toutes les communautés.

— **Regroupment 3 :**

Si la liste de chevauchement contient plus de 10 élément :

Comparer toutes les paires possibles de communautés dans la liste des communautés : Si seuil(deux communautés) <= 5 Fusion similaire aux étapes précédentes.

Obtenir une liste chevauchement contenant les éléments répétés uniques parmi les membres de toutes les communautés.

— **Regroupment 4 :**

Si la liste de chevauchement contient plus de 50 élément :

Comparer toutes les paires possibles de communautés dans la liste des communautés : Si seuil(deux communautés]) <= 10 Fusion similaire aux étapes précédentes.

Obtenir une liste chevauchement contenant les éléments répétés uniques parmi les membres de toutes les communautés.

— **Regroupment 5 :**

Si la liste de chevauchement contient plus de 100 élément :

Comparer toutes les paires possibles de communautés dans la liste des communautés : Si seuil(deux communautés) <= 20 Fusion similaire aux étapes précédentes.

Obtenir une liste chevauchement contenant les éléments répétés uniques parmi

les membres de toutes les communautés.

3.6 Algorithme de l'approche

Entrées :G = (V, E) : le graphe initial.

Sorties : C1, C2, C3, ..., Ck_1 : les communautés.

3.6.1 1 Algorithme de la première phase

Les fonctions

Reseau-real() :permet de charger un fichier d'arêtes,d'afficher des informations sur le graphe et de retourner le graphe créé G.(par exmple : le reseau karate)

Affiche-graphe() : : une fonction qui permet d'afficher le graphe G.

3.6.2 Pseudo code

Algorithm 1 Reseau_real

```
1: Entrées : Aucune
2: Sortie : Graphe G
3: Début
4:    Lire 'txt.txt'
5:       Retour G
6: Fin
```

Algorithm 2 Affiche-graphe

```
1: Début
2:    Dessiner_graphe(G)
3:    G.visible
4: Fin
```

3.6.3 1 Algorithme de La deuxième phase

Les fonction

calculer_densité :une fonction qui permet calculer la densité local des noeuds du graph G selon equation 2.8.

Classer_les noeuds :une fonction qui permet triez lzs noeuds leur fonction de densité.

Algorithm 3 densité

1: **Entrées** : Graphe G ,noeud_densité : un dictionnaire pour stocker les densités calculées pour chaque nœud
2: **Sortie** :liste_noeuds
3: **Début**
4: Pour chaque nœud node dans le graphe G
5: noeud_densité = calculer_densité(noeud)
6: Fin pour
7: liste_noeud = Classer_les noeuds(noeud_densité)
8: **Fin**

3.6.4 Algorithme de La troisième phase

Les fonction

voisins() : une fonction qui retourne une liste des voisins d'un nœud.

grouping(L) :une fonction qui permet de regrouper des éléments similaires en utilisant un critére de similarité pour déterminer à quel point deux éléments sont considérés comme similaires pour fusioner les commnautés similaire.

Algorithm 4 Detecter_communautés

```
1: Entrées :graph G , ll = [] , d = densité()
2: Sortie :Communautés disjointes
3: Début
4:    Pour chaque i de d :
5:       l = voisins(i)
6:       Ajouter i à la fin de la liste l
7:       Ajouter la liste l à la liste ll
8:       Concaténer la liste l à la liste g
9:       g = Supprimer les doublons de la liste g
10:      Si |g|=|G| :
11:      Sortir de la boucle principale
12:   Fin pour
13:   resultat = grouping(ll)
14:   affiche("resultat")
15: End
```

3.7 Un exemple illustratif

Pour montrer l'efficacité de notre approche, on utilise l'exemple suivant : Notre exemple est un graphe avec 9 nœuds et 14 arrêtes, ce graphe est disjoint et non orienté. Tous les schémas qu'on va utiliser au cours de ces exemples sont de notre application.

Notre exemple se déroule en trois phases :

3.7.1 La première phase

A ce stade, nous allons importer un réseau de nœuds et 14 arêtes (exp1) et le convertir en graphe.

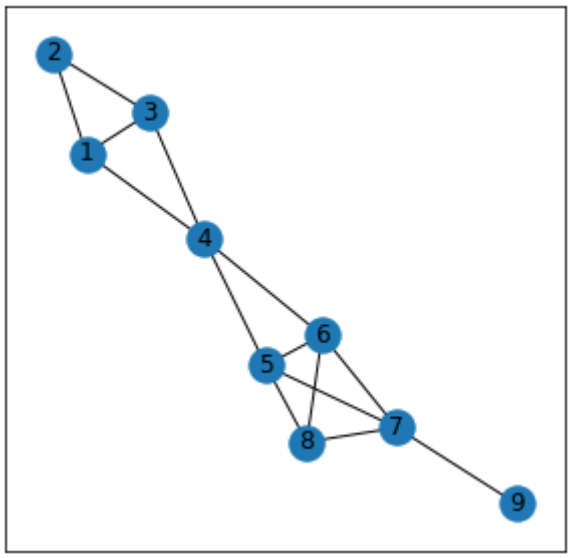

FIGURE 3.2 – Représentation graphique du exp1.

3.7.2 La deuxième phase

dans cette phase,notre algorithme calcule la densité local des noeuds et les triée en ordre décroissante.

Voila la figure 3.3 illustre les résultats de cette étape.

Résultats triés par densité

```
Liste triée par densité décroissante:
Node: 2, Densité: 0.8333333333333334
Node: 9, Densité: 0.7
Node: 7, Densité: 0.6
Node: 8, Densité: 0.6
Node: 1, Densité: 0.5333333333333333
Node: 3, Densité: 0.5333333333333333
Node: 4, Densité: 0.4642857142857143
Node: 5, Densité: 0.42857142857142855
Node: 6, Densité: 0.42857142857142855
```

FIGURE 3.3 – Resultat de la liste triée par densité de exp1.

3.7.3 La troisième phase

cette phase se déroule :

1_**Sélectionne le premier elemment de la liste triée.**

Dans l'example exp1 on a**le noeud 2**

2_**Obtenir les voisins des noeuds a partir du noeud plus dense jusqu'à ce que tous les nœuds aient été visités**

le noeud 2 :

['1', '3', '2']

le noued 9 :

['1', '3', '2']

['7', '9']

le noued 7 :

['1', '3', '2']

['7', '9']

['5', '6', '8', '9', '7']

le noeud 8 :

['1', '3', '2']

['7', '9']

['5', '6', '8', '9', '7']

['5', '6', '7', '8']

le noeud 1 :

['1', '3', '2']

['7', '9']

['5', '6', '8', '9', '7']

['5', '6', '7', '8']

['4','3','2','1']

voila tout les nœuds aient été visités

3_**faire le regroupment** on a la liste des communautés similaires suivant :

communaute O :['1', '3', '2']

communaute 1 :['7', '9']

communaute 2 :['5', '6', '8', '9', '7']

communaute 3 :['5', '6', '7', '8']

communaute 4 :['4','3','2','1']

nous utilisons deux boucles imbriquées pour comparer toutes les paires possibles de communautés dans la liste des communautés. La variable i représente l'indice de la première communauté dans la boucle extérieure, tandis que la variable j représente l'indice de la deuxième communauté dans la boucle intérieure. Ainsi, lorsque i vaut 0, cela signifie que vous comparez la communauté à l'indice 0 (communauté 0) avec toutes les autres communautés dont l'indice est supérieur à 0 (communauté 1, communauté 2, etc.).

La comparison entre les commnautés avec le calcule du seuil de similarité.

Similarité entre communauté 0 et communauté 1 : 2

Similarité entre communauté 0 et communauté 2 : 3

Similarité entre communauté 0 et communauté 3 : 3

Similarité entre communauté 0 et communauté 4 : 0

Similarité entre communauté 1 et communauté 2 : 0

Similarité entre communauté 1 et communauté 3 : 0

Si le seuil <= 1 Les communautés sont fusionnées en ajoutant les éléments de la deuxième communauté à la première, puis à supprimer les doublant et eliminer la liste vide.

Liste des communautés :

Communauté 1 : ['6', '9', '7', '5', '8']

Communauté 2 : ['4', '3', '2', '1']

Enfin,nous prendrons la liste des communautés appliquer la fonction (chev) renverra une liste contenant les éléments qui se répètent parmi tous les membres de toutes les communautés.

Nbr Communautés 2

Communauté 1 : [’6’, ’9’, ’7’, ’5’, ’8’]

Communauté 2 : [’4’, ’3’, ’2’, ’1’]

Le nombre des noeuds chevauchantes est : 0

Le schema 3.4 montre un affichage graphique des communautés détectées.

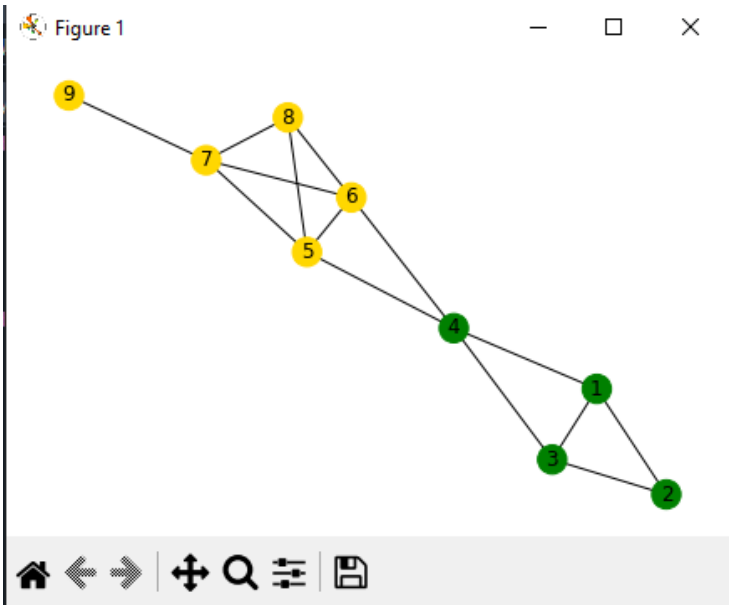

FIGURE 3.4 – Représentation graphique des communautés détectées de cet exemple

3.8 Conclusion

Dans ce chapitre, nous proposons une nouvelle méthode pour détecter les communautés disjointes dans les réseaux, et montrons à travers un exemple illustratif détaillé que notre méthode est capable de détecter facilement les communautés disjointes, et même si nous répétons l’exécution plusieurs fois, les nœuds superposés ne changent pas non plus. En raison de l’ordre que nous faisons.

Dans le chapitre suivant, nous définissons l’environnement de travail matériel et logiciel, implémentons notre algorithme principal de détection de communauté et comparons leurs résultats avec d’autres algorithmes à travers quelques exemples.

Chapitre 4

Implmentation

4.1 Introduction

Ce chapitre présente les outils et les langages utilisés pour mettre en œuvre les méthodes proposées dans le chapitre précédent. Nous présentons ensuite les résultats obtenus après avoir comparé notre méthode avec plusieurs algorithmes de l'état de l'art.

4.2 Environnement de travail

4.2.1 Environnement matériel

Nous utiliserons un micro-ordinateur qui répond aux spécifications suivantes pour toutes nos installations et nos tests :

- `Processeur : Intel(R) Core(TM) i5 CPU  M540 @ 2.53GHz ;`
- `Mémoire Installée (RAM) : 4 Go ;`
- `Type de système : système d'exploitation 64bits,Processeur64.`

Notre application a été développée sous un système d'exploitation Windows 10 de 64 bits où on a utilisé le langage de programmation Python 3.8 sous Anaconda.

4.2.2 Environnement de logiciel

Nous avons utilisé le langage de programmation Python la version 3.9.8

python™ est un langage de programmation polyvalent, interprété, de haut niveau et convivial. Il a été créé par Guido van Rossum et a été publié pour la première fois en 1991.

Voici quelques fonctionnalités communes de Python :

- **Syntaxe claire et lisible :** Python se concentre sur la lisibilité du code, ce qui en fait un langage facile à apprendre et à comprendre. Cette syntaxe utilise l'indentation pour séparer les blocs de code, ce qui rend le code Python beaucoup plus lisible.
- **Langage polyvalent :**Python peut être utilisé pour développer une large gamme d'applications, des scripts simples aux applications Web complexes, y compris le traitement des données, l'intelligence artificielle, l'apprentissage automatique, les données de science des données, l'automatisation des tâches, etc.
- **Communauté active :** Python a une communauté de développeurs très active. Il existe de nombreux modules développés par la communauté et des packages tiers qui étendent les fonctionnalités de base de Python.

L'installation de python est gratuite et facile, il suffit de le télécharger dans le site : https://www.python.org/downloads/.(Voir figure 4.1)

FIGURE 4.1 – Le site d'installation de python

4.2.3 Plateforme et IDE

Après avoir installé Python, nous avons procédé à l'installation de la plateforme et de tous les logiciels dont nous avions besoin, pour lesquels nous avons choisi Anaconda.

ANACONDA est une plateforme open source qui offre un environnement de travail adapté à l'utilisation de langages de programmation open source tels que R et Python pour le traitement de données à grande échelle, l'analyse prédictive et le calcul scientifique.L'avantage d'Anaconda réside dans sa capacité à fournir un ensemble intégré d'outils, de packages et de gestion d'environnement, ce qui facilite la mise en œuvre de projets de recherche environnementale nécessitant une analyse avancée des données.

Téléchargez et installez la dernière version appropriée de la plate-forme Anaconda basée sur le système d'exploitation de l'ordinateur de l'utilisateur et la dernière version de Python à partir du site Web d'Anaconda :

https://www.anaconda.com/download.(Voir figure 4.2)

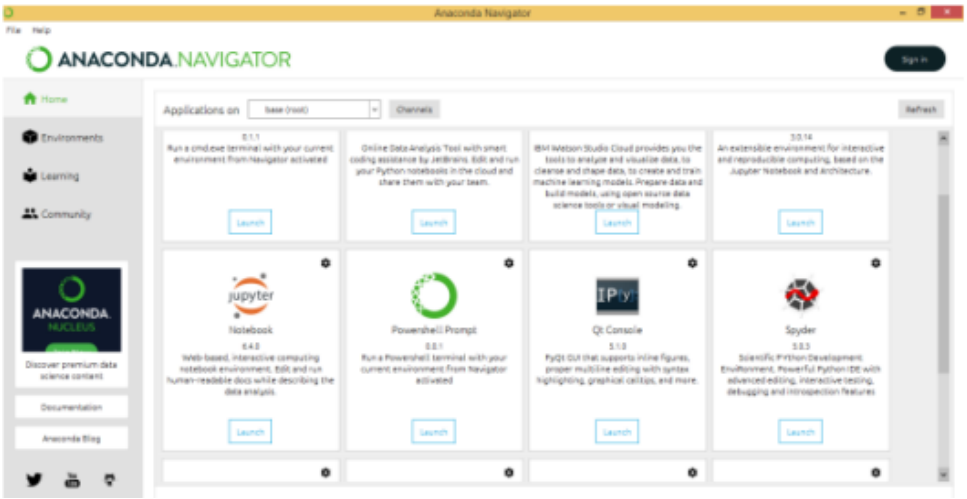

FIGURE 4.2 – Anaconda Navigator

4.2.4 l'environnement de développement (IDE) Spyder

Après avoir installé Python etla plateforme et de tous les logiciels.Nous avons utilisé l'environnement de développement (IDE) Spyder

SPYDER est un environnement de développement intégré (IDE) pour Python. Il est open source et écrit en Python. Spyder a été conçu spécifiquement pour les scientifiques, les ingénieurs et les analystes de données, en offrant des fonctionnalités avancées pour faciliter leur travail. Il offre une interface conviviale avec une disposition flexible des panneaux pour faciliter le développement et l'analyse de code. Il prend également en charge l'intégration avec d'autres bibliothèques scientifiques populaires, telles que NumPy, SciPy et Matplotlib, ce qui en fait un choix populaire parmi les scientifiques utilisant Python. La figure 4.3 présente l'Interface de l'IDE Spyder

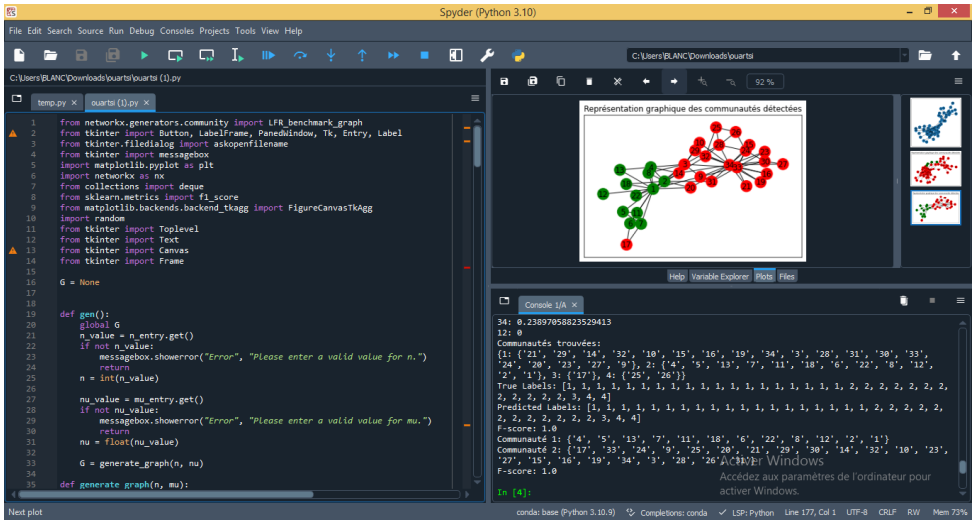

FIGURE 4.3 – Interface de l'IDE Spyder

4.2.5 Bibliothèques

Une bibliothèque est un ensemble de fonctions, de classes et de modules préécrits.Les bibliothèques Python sont écrites en utilisant le langage Python lui-même, et elles peuvent être importées dans d'autres programmes pour utiliser leurs fonctions et classes.

On a utilisé plusieurs bibliothèques dans ce travail qui sont :

NetworkX : est une bibliothèque Python dédiée à la création, à la manipulation et à l'analyse de structures de graphes. Elle fournit des outils pour travailler avec des graphes dirigés et non dirigés, ainsi que des fonctionnalités avancées pour l'analyse des réseaux[63].

Tkinter : est une bibliothèque standard de Python qui permet de créer des interfaces graphiques (GUI) pour des applications de bureau. Le nom Tkinter est dérivé de "Tk

interface", car il s'agit d'une interface Python pour le toolkit Tk, qui est un ensemble d'outils graphiques multiplateformes[64].

Matplotlib : est une bibliothèque Python largement utilisée pour la création de visualisations graphiques, notamment des graphiques, des diagrammes, des histogrammes, des nuages de points et bien plus encore. Elle permet de générer des représentations visuelles de données de manière efficace et flexible[65].

Community :est une bibliothèque Python utilisée pour détecter la structure de la communauté dans les graphes. Elle met en œuvre et permet de visualiser les algorithmes de détection des communautés. En utilisant cette bibliothèque, vous pouvez analyser un graphe et identifier les groupes de nœuds étroitement liés qui forment des communautés distinctes[66].

scikit-learn : également connu sous le nom de sklearn, est une bibliothèque populaire en Python pour l'apprentissage automatique (machine learning). Elle fournit des outils simples et efficaces pour la modélisation de données et l'application d'une variété d'algorithmes d'apprentissage automatique[67].

Pour installer les packages requis, entrez simplement le code suivant à l'invite Anaconda Powershell :

NetworkX : pip install Networkx.

Community : pip install Community

Matplotlib : pip install matplotlib.

Le téléchargement des packages nécessite l'accès à l'internet.

4.2.6 Présentation du système

Dans cette section on va présenter quelque module de notre application.La figure 4.4 ci-dessous montre l'interface principale de notre application,elle est composé

- Importation des données réels.
- Affichage du graphe des données à traiter.
- Application de l'algorithme de la méthode proposée sur ces données.

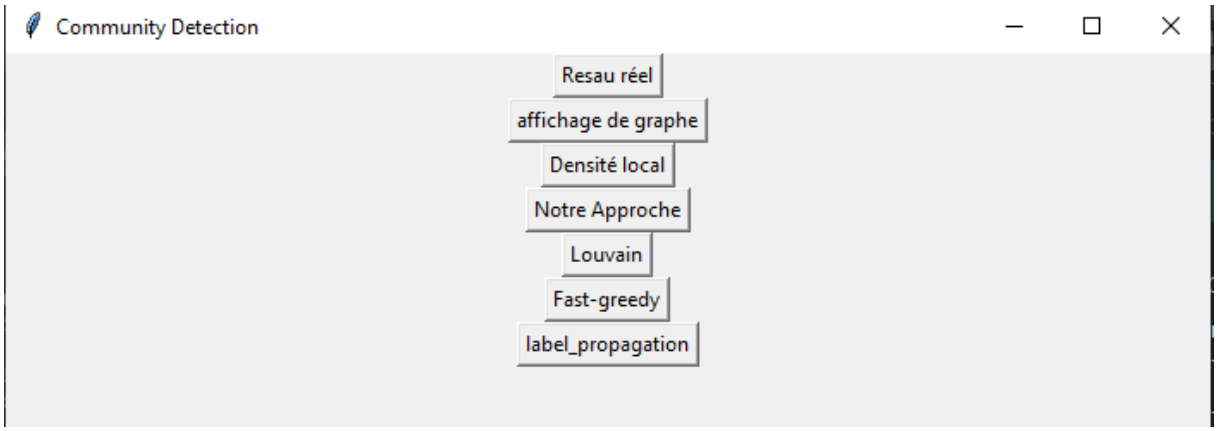

FIGURE 4.4 – Interface générale de l'application

Dans cette section, nous expliquons quelques parties.

1_Importation des données réels :

En cliquant sur le bouton « Réseau réel », une fenêtre s'affichera (voir Figure 4.5) permettant de sélectionner un fichier texte .txt représentant la base de données réelle.

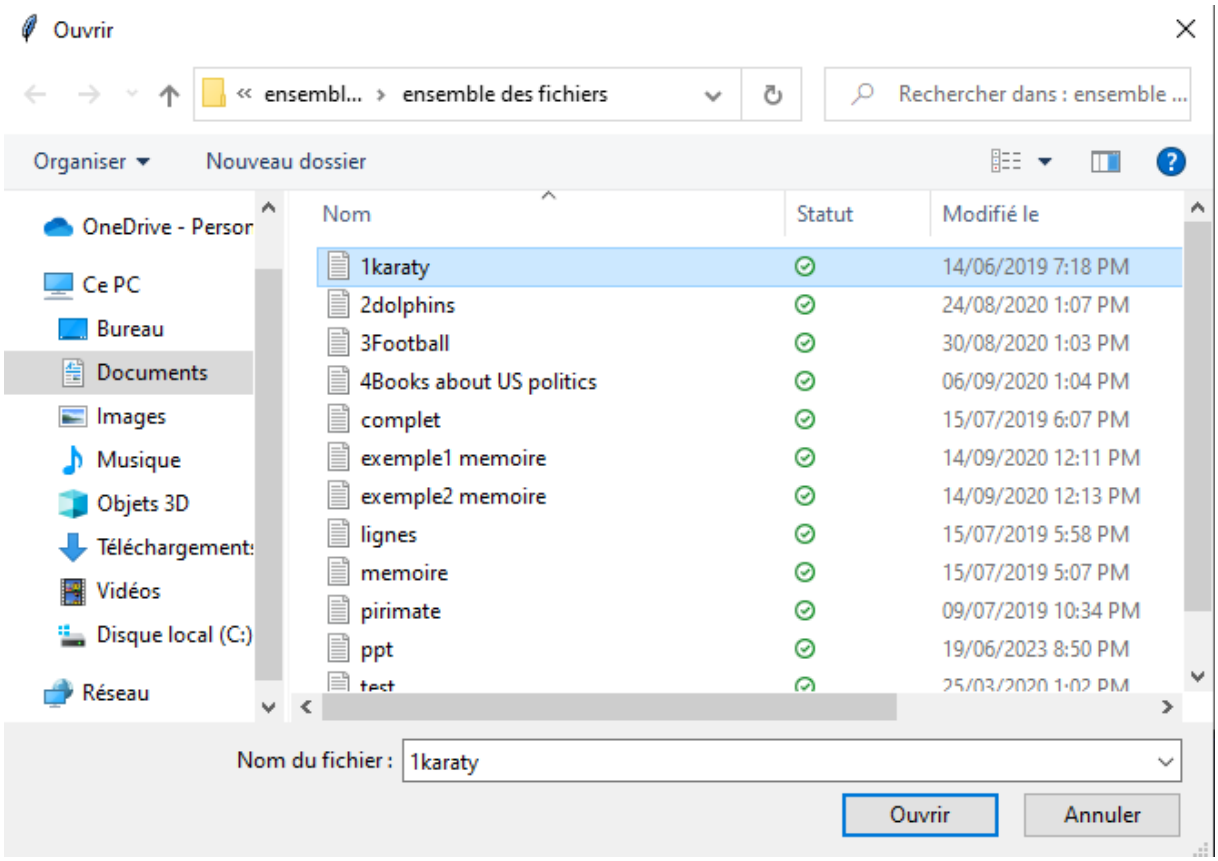

FIGURE 4.5 – Importation des bases de données

2_calcul densité locale

En cliquant sur le boton densité local s'affiche Liste triée par densité décroissante(voir figure 4.6).

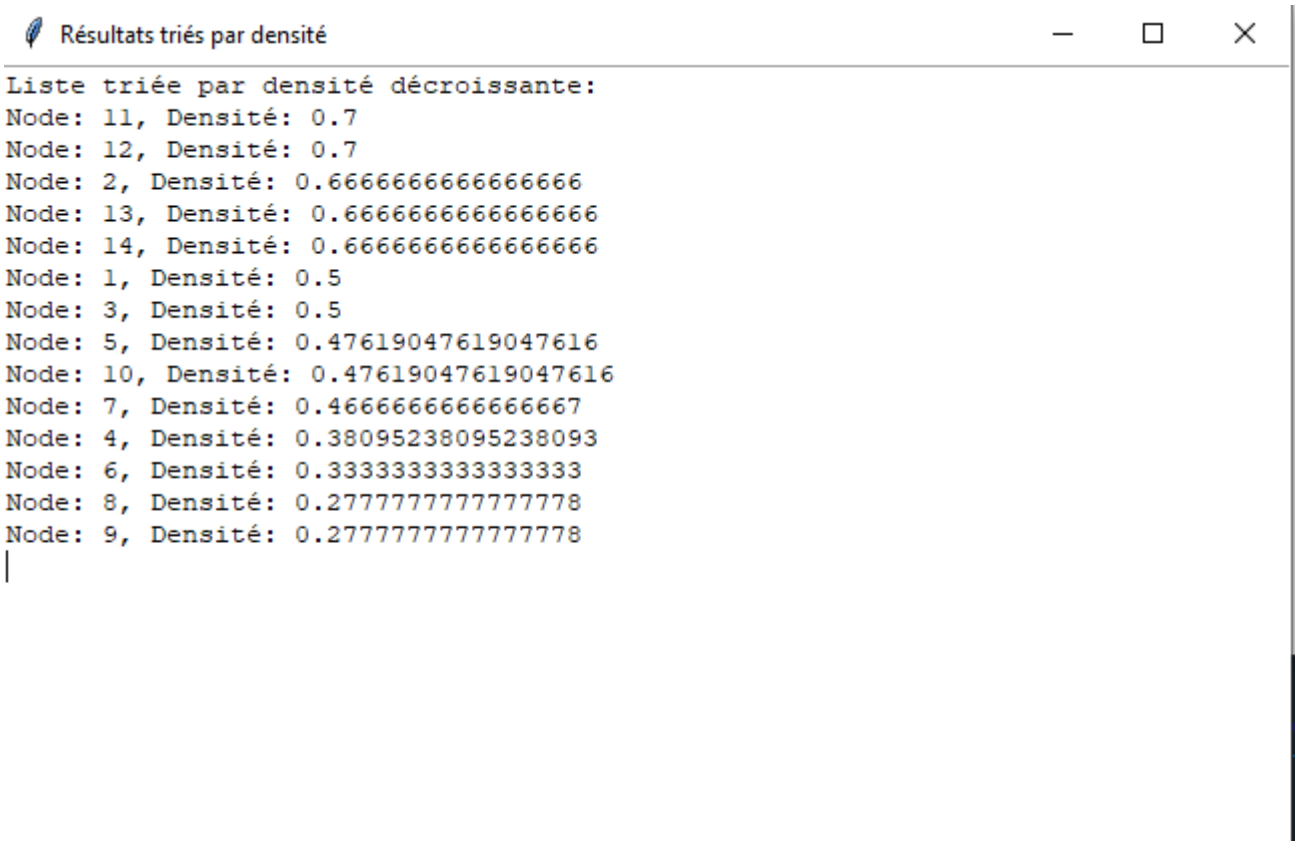

FIGURE 4.6 – Liste triée par densité décroissante du réseau exp2

3_Affichage du graphe des données à traiter :

: En cliquant sur le bouton Affichage du graphe, une représentation graphique du réseau s'affiche (voir figure 4.7).

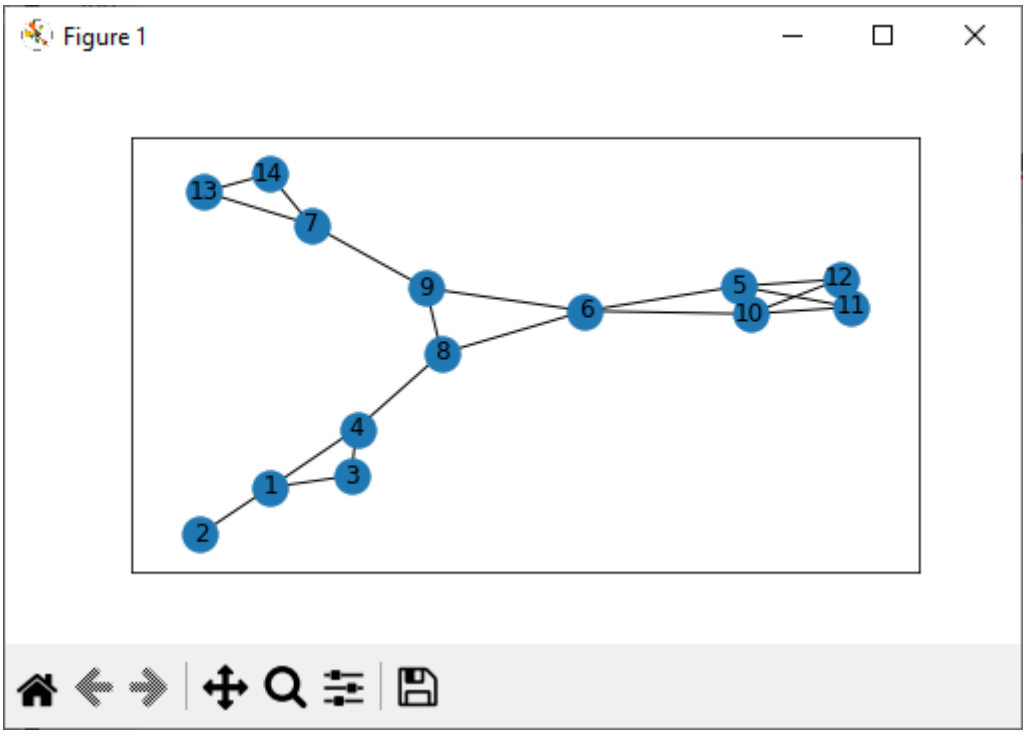

FIGURE 4.7 – Représentation graphique du réseau de exp2

4_Application de l'algorithme de la méthode proposée sur ces données :

Cliquer sur le bouton « Notre méthode » fera apparaître deux fenêtres. L'un montre une représentation graphique (voir Figure 4.8) et l'autre montre le nombre de communautés et la modularité du réseau (par exemple réseau exp2) après application de l'algorithme à ce réseau (le réseau exp2 est un réseau de 14 nœuds et 19 arrêtes) (voir Figure 4.8).

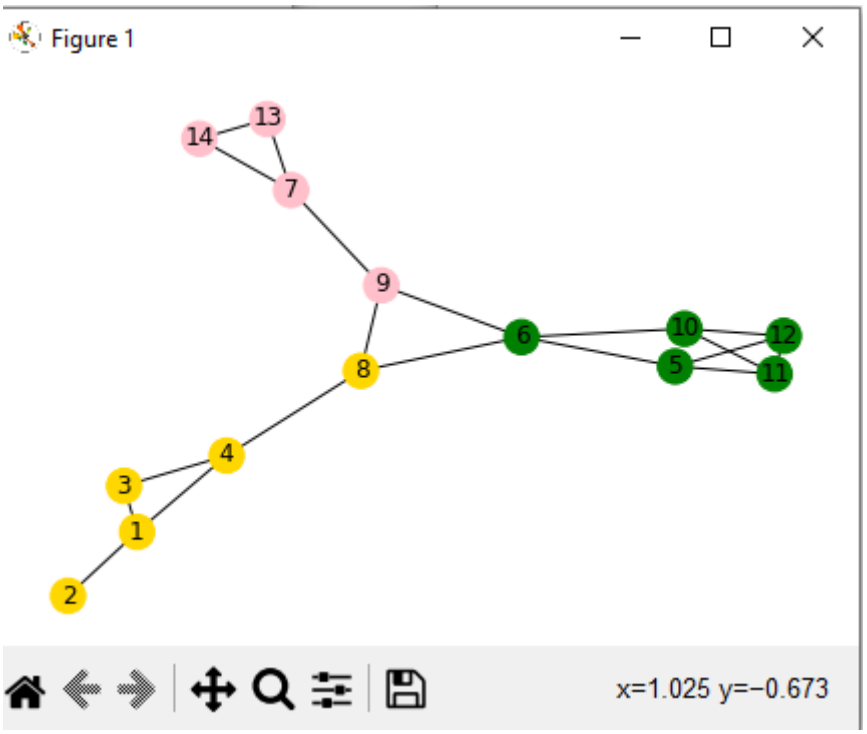

FIGURE 4.8 – Représentation graphique de l'exécution de notre l'algorithme sur le réseau de exp2

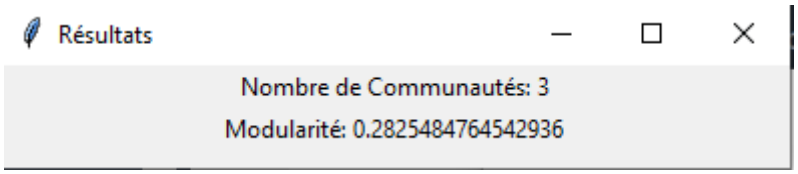

FIGURE 4.9 – : Résultat de l'exécution de notre l'algorithme sur le réseau de exp2

5_Application des algorithmes décrits dans la section 1.3.3

4.3 Résultats Discussion

La détection de communautés dans les réseaux complexes est effectivement un domaine de recherche complexe et délicat en raison de l'absence de définition précise d'une "bonne" communauté. L'évaluation de la pertinence des communautés détectées par un algorithme peut être subjective et dépendre des objectifs spécifiques de l'analyse. Cependant, il existe plusieurs critères et approches qui peuvent être utilisés pour évaluer la qualité des résultats d'un algorithme de détection de communautés.

Dans votre travail, la performance de l'algorithme proposé a été évaluée à l'aide d'une mesure d'évaluation : la modularité (Q). De plus, vous avez testé l'algorithme sur des réseaux

réels (karaty, dolphin, livres politiques, Football américain) où le nombre de communautés est déjà connu, en comparant vos résultats avec ceux des algorithmes connus,soit Louvin [?] , Label Propagation [28]et Fast Gerdey [68].

Club de karaté de Zachary

La comparison de notre algorithme avec le club de karaté de Zachary est une approche courante pour évaluer les performances des algorithmes de détection de communautés. Le club de karaté de Zachary[35]est un réseau bien connu qui est souvent utilisé comme un cas d'étude standard dans ce domaine.ce réseau est composé de 34 nœuds, représentant les membres d'un club de karaté, et 78 arêtes, qui représentent les interactions (amitiés ou affiliations) entre ces membres.

Les figures ci-dessus nous montre le résultat d'exécution de notre methode et les algorithmes connus sur le réseau de Zachary

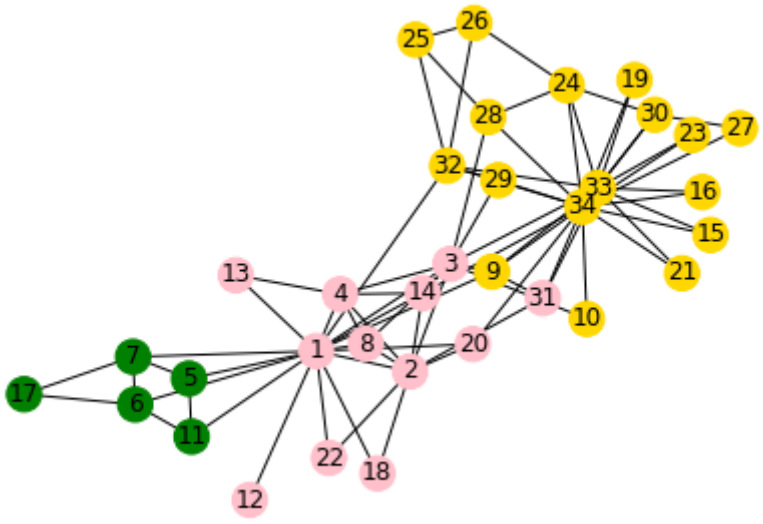

FIGURE 4.10 – Représentation graphique trouvée par notre methode pour le réseau de Zachary

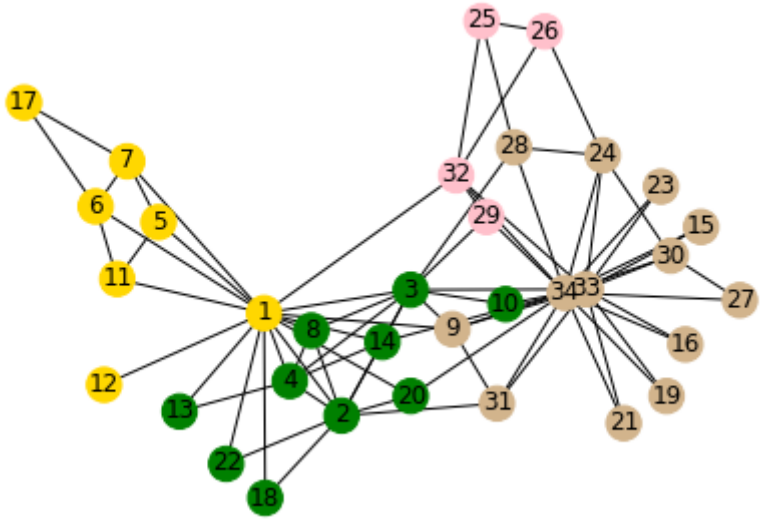

FIGURE 4.11 – Représentation graphique trouvée par l'algorithme de louvin pour le réseau de Zachary

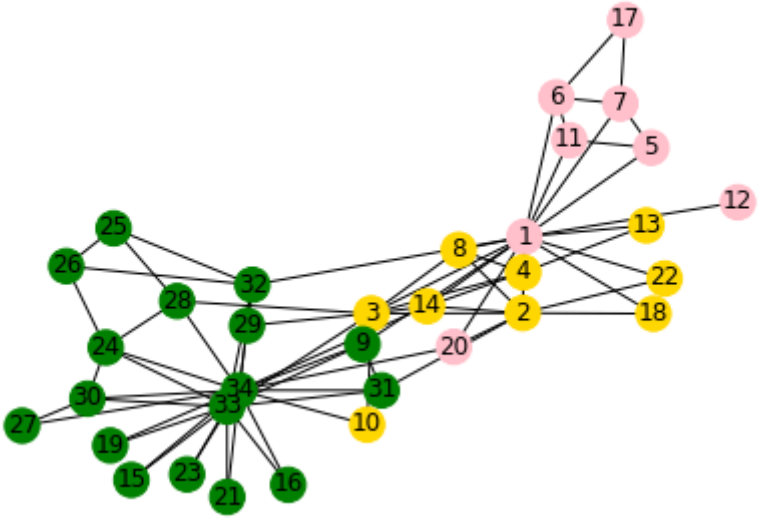

FIGURE 4.12 – Représentation graphique trouvée par l'algorithme de Fast-Gerdy pour le réseau de Zachary

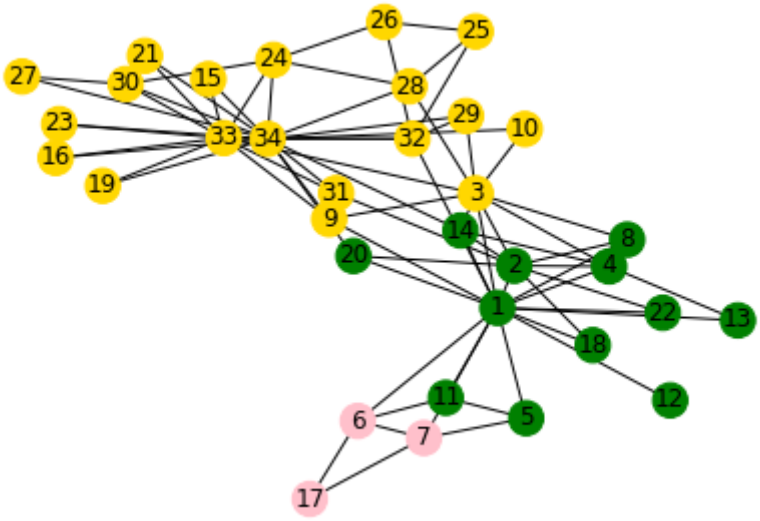

FIGURE 4.13 – Représentation graphique trouvée par l'algorithme de label-propagation pour le réseau de Zachary

Méthodes	Nombre de communautés	Modularité (Q)
Notre méthode	3	0.27
Louvain	4	0.31
Fast-Greedy	3	0.20
Label-propagation	3	0.06

TABLE 4.1 – Résultat d'exécution des algorithmes sur le réseau de Zachary.

Les dauphins de Lusseau

Une deuxième comparaison a été faite avec le réseau des dauphins de Lusseau[33]. Ce réseau contient 62 nœuds et 159 liens. La méthode proposée reconnaissait trois communautés. Ce réseau est essentiellement composé de deux communautés.

Les figures ci-dessus nous montre le résultat d'exécution de notre methode et les algorithmes connus sur le réseau de dauphins :

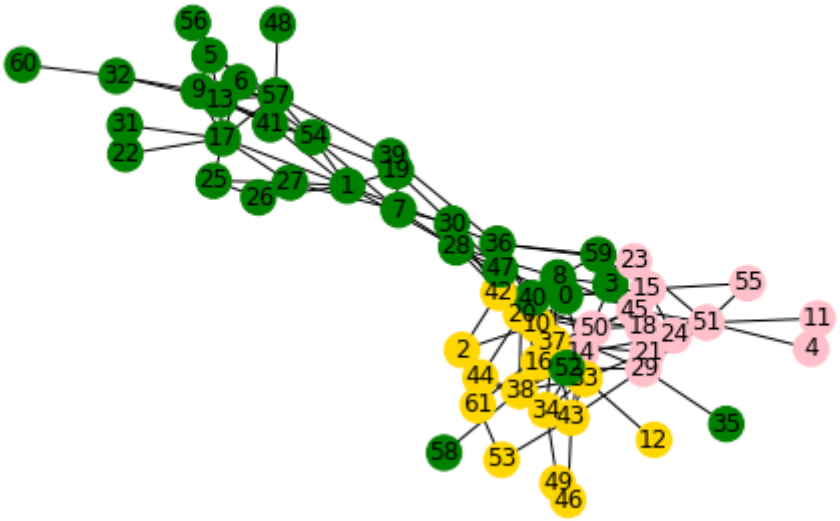

FIGURE 4.14 – Représentation graphique trouvée par notre methode pour le réseau de dauphins

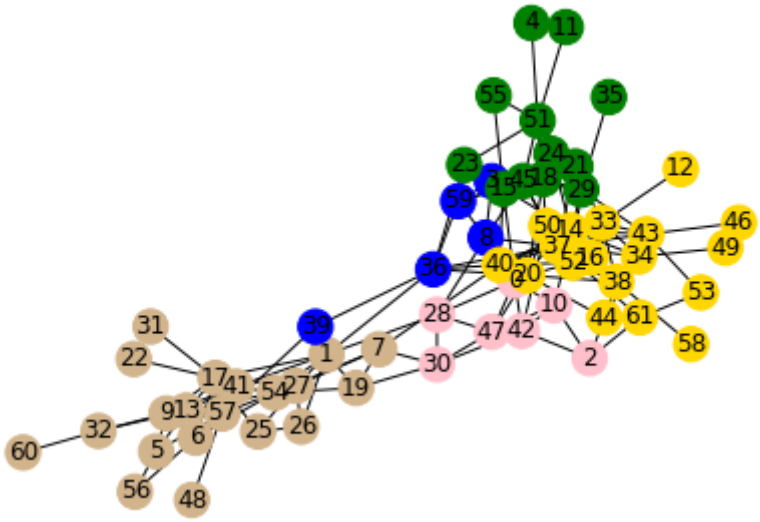

FIGURE 4.15 – Représentation graphique trouvée par l'algorithme de louvin pour le réseau de dauphins

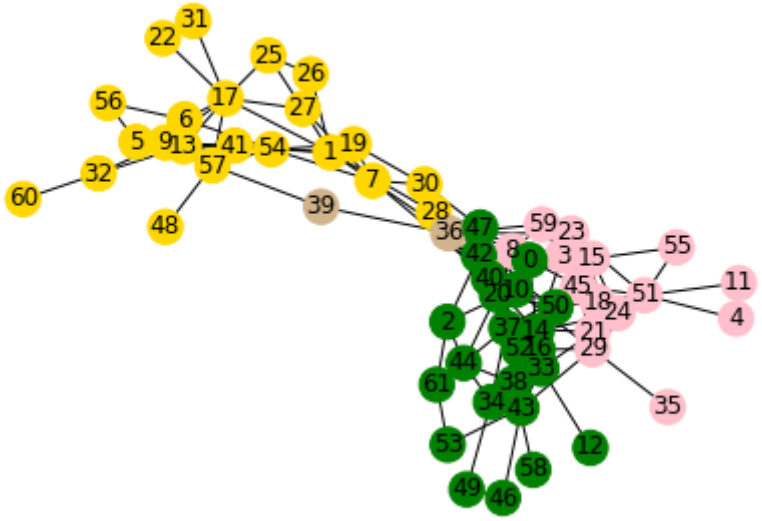

FIGURE 4.16 – Représentation graphique trouvée par l'algorithme de Fast-Gerdy pour le réseau de dauphins

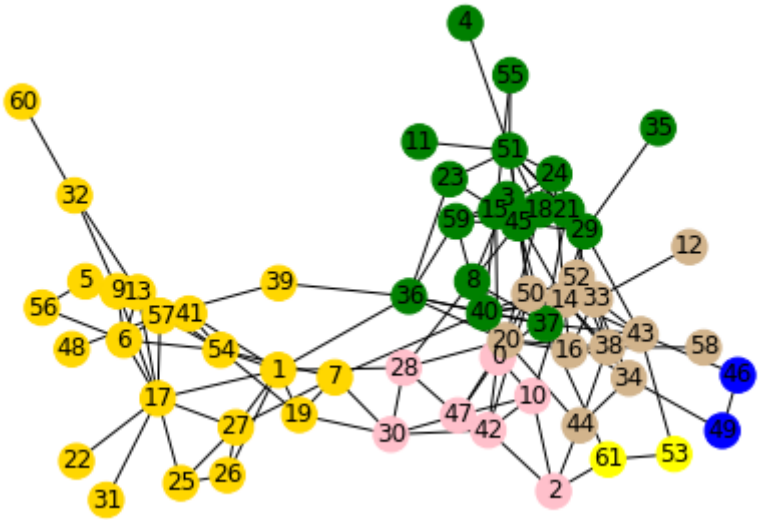

FIGURE 4.17 – Représentation graphique trouvée par l'algorithme de label-propagation pour le réseau de dauphins

Méthodes	Nombre de communautés	Modularité (Q)
Notre méthode	3	0.20
Louvain	5	0.22
Fast-Greedy	4	0.08
Label-propagation	6	0.01

TABLE 4.2 – Résultat d'exécution des algorithmes sur le réseau de dauphins.

Livres sur la politique américaine

Une troisième comparaison a été réalisée par le Political Books Network. Ce réseau se compose de 105 livres et 441 liens.

Les figures ci-dessus nous montre le résultat d'exécution de notre methode et les algorithmes connus sur le réseau livres poulitique :

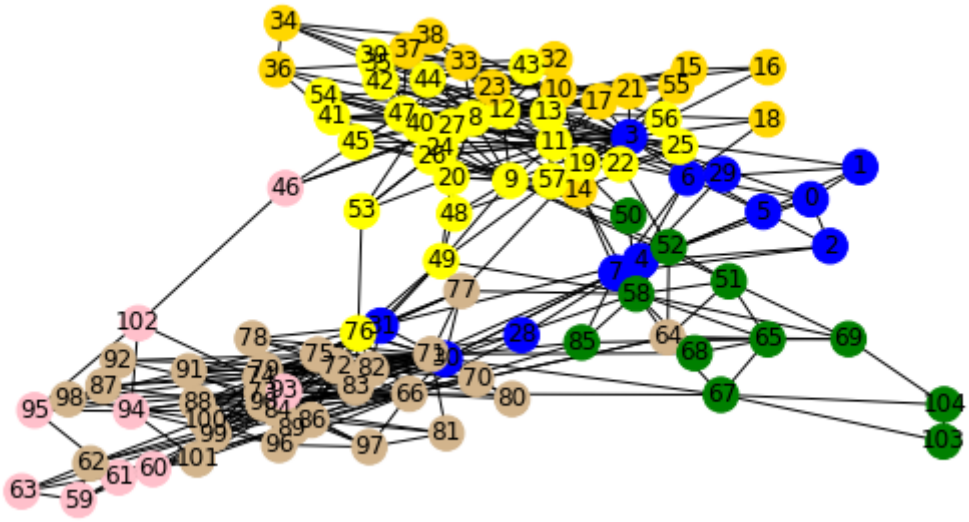

FIGURE 4.18 – Représentation graphique trouvée par notre methode pour le réseau livres poulitique

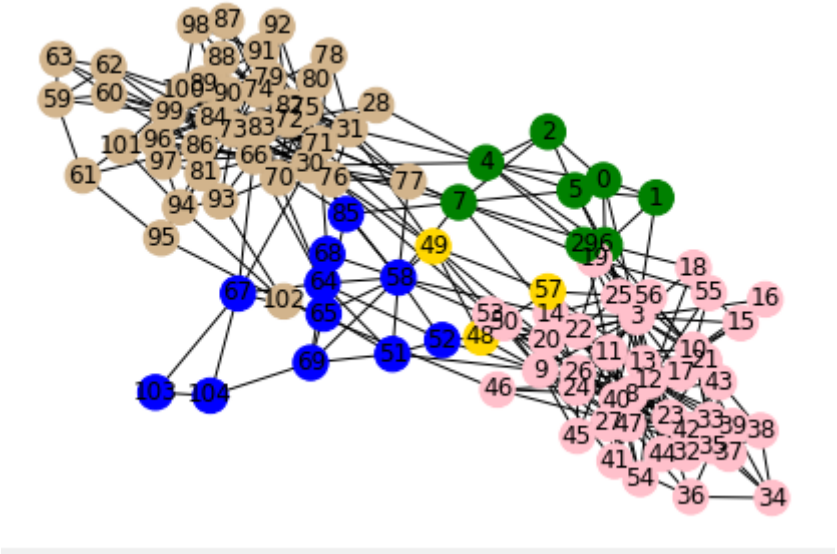

FIGURE 4.19 – Représentation graphique trouvée par l'algorithme de louvin pour le réseau livres poulitique

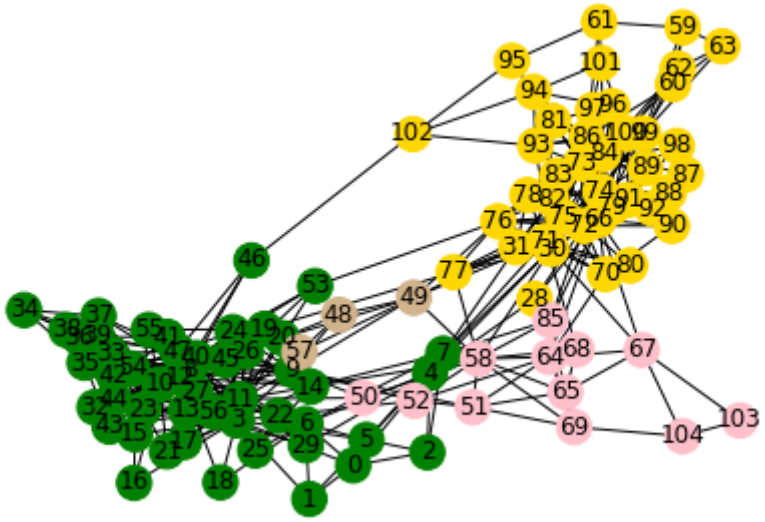

FIGURE 4.20 – Représentation graphique trouvée par l'algorithme de Fast-Gerdy pour le réseau livres poulitique

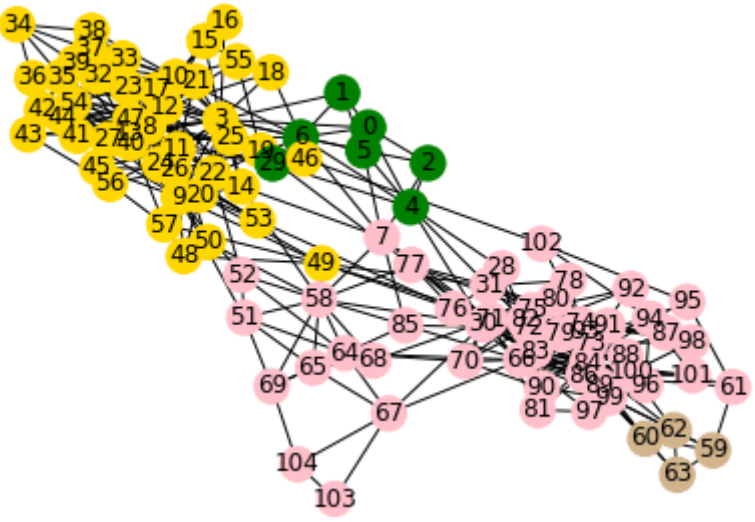

FIGURE 4.21 – Représentation graphique trouvée par l'algorithme de label-propagation pour le réseau livres poulitique

Méthodes	Nombre de communautés	Modularité (Q)
Notre méthode	6	0.23
Louvain	5	0.44
Fast-Greedy	4	0.08
Label-propagation	4	0.04

TABLE 4.3 – Résultat d'exécution des algorithmes sur le réseau livres poulitique.

Football

une autre comparaison significative a été réalisée en examinant un réseau réel distinct, à savoir le réseau de jeux de football américain [34]. Ce réseau présente une structure tout à fait différente de celle du réseau de livres politiques. Il est composé de douze communautés distinctes, comptant 115 nœuds interconnectés par un total de 613 liens

Les figures ci-dessus nous montre le résultat d'exécution de notre methode et les algorithmes connus sur le réseau de Football

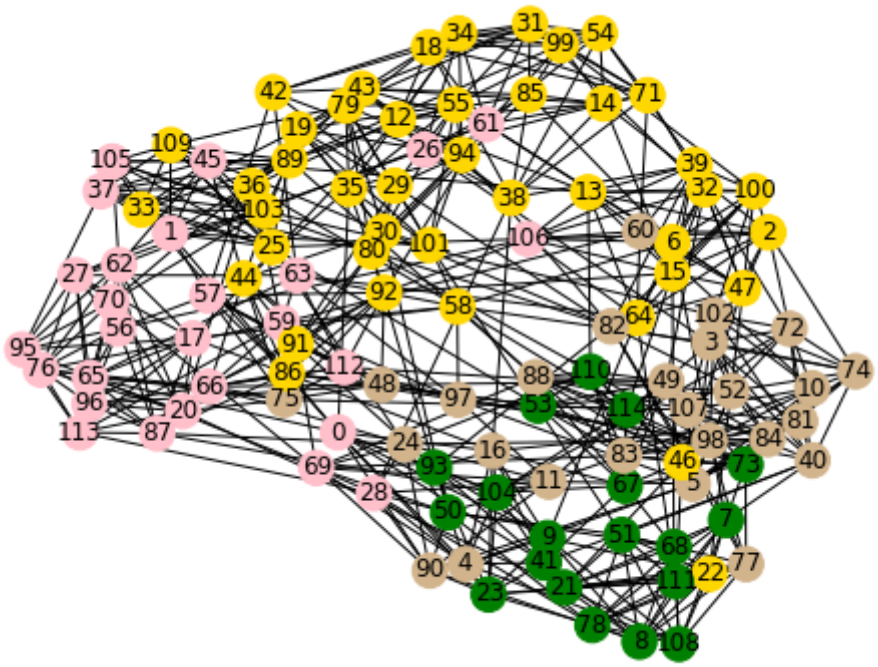

FIGURE 4.22 – Représentation graphique trouvée par notre methode pour le réseau de Football

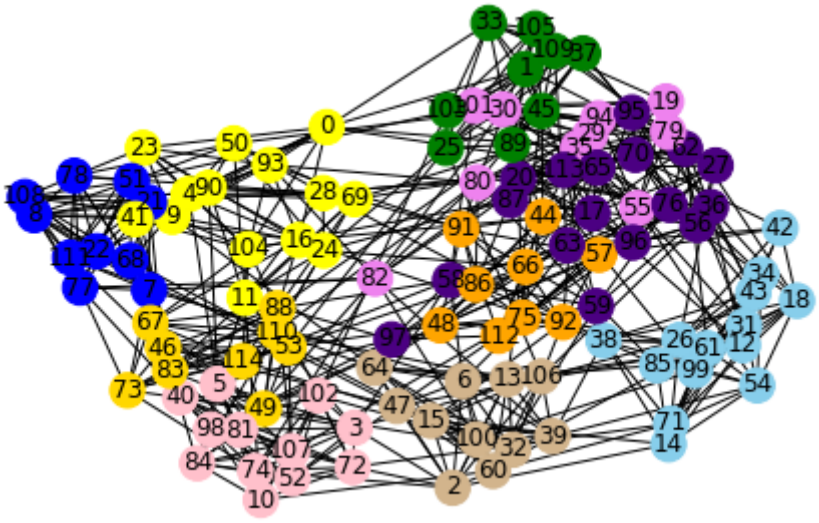

FIGURE 4.23 – Représentation graphique trouvée par l'algorithme de louvin pour le réseau de Football

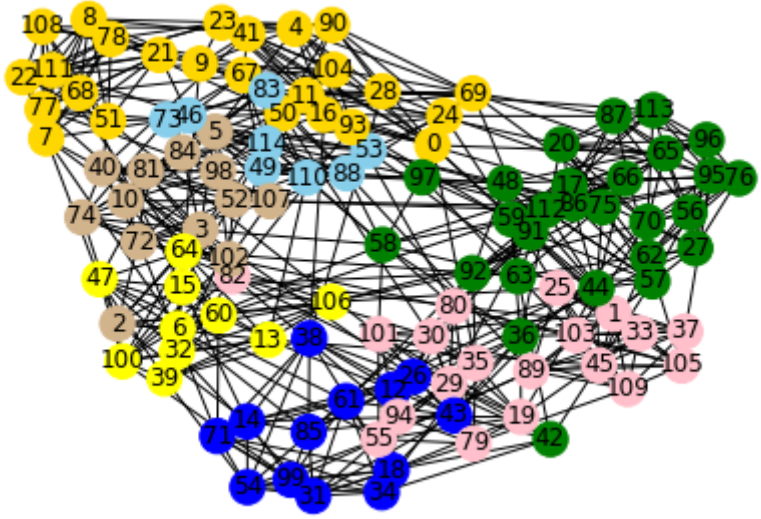

FIGURE 4.24 – Représentation graphique trouvée par l'algorithme de Fast-Gerdy pour le réseau de Zachary

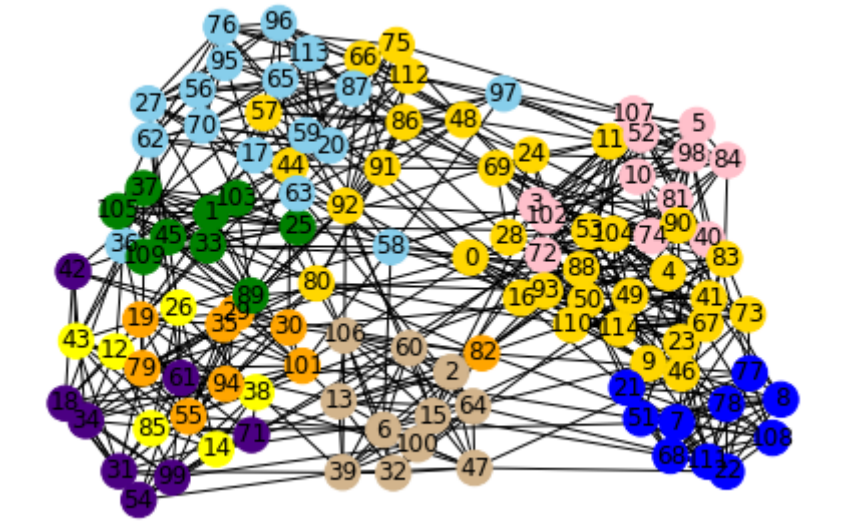

FIGURE 4.25 – Représentation graphique trouvée par l'algorithme de label-propagation pour le réseau de Football

Méthodes	Nombre de communautés	Modularité (Q)
Notre méthode	4	0.15
Louvain	9	0.15
Fast-Greedy	7	0.10
Label-propagation	9	0.08

TABLE 4.4 – Résultat d'exécution des algorithmes sur le réseau de Football.

Dans cette recherche , Nous avons proposé une nouvelle méthode de détection de communautés basée sur la densité avec une itération et une fusion progressive des communautés en fonction de seuils de similarité que permet pricisé améliore la précision des résultats de détection de la communauté.en plus que La modularité utilise comme mesure de la qualité des communautés détectées que donner des resultat mieux que plusieurs méthodes connus sur le monde réel.

Les résultats d'exécution de notre algorithme restent toujours stables.

Trouver un équilibre entre la détection de petites communautés spécifiques et la détection de communautés plus larges.

Notre est relativement simple à comprendre et à mettre en œuvre. Elle ne nécessite pas une compréhension approfondie de méthodes complexes de détection de communautés, ce qui peut la rendre accessible aux personnes qui débutent dans ce domaine.

En ce qui concerne les performances en termes de temps d'exécution, notre approche s'est

avérée très efficace. En effet, elle reste rapide même avec les réseaux réels que nous avons utilisés, affichant des temps d'exécution de 0,0126 s pour le réseau Karaty, 0,07 s pour le réseau Dauphin, 0,25 s pour Poolbooks et 0,51 s pour Football. Ainsi, quelle que soit la taille du réseau, notre approche demeure rapide et efficace.

4.4 Coclusion

Dans ce chapitre, nous avons commencé par présenter l'environnement matériel et logiciel que nous avons utilisé dans notre travail, suivi d'une présentation détaillée de notre système. Ensuite, nous avons exposé les conclusions de notre recherche en comparant notre approche et en analysant les données collectées.

Conclusion Générale

L'objectif principal de ce mémoire était d'identifier des groupes distincts au sein de réseaux.

Le premier chapitre s'est concentré sur la clarification des concepts liés à la détection de ces groupes, ainsi que sur l'examen de certaines notions fondamentales de la théorie des graphes. De plus, nous avons examiné différentes méthodes de mesure pour évaluer le processus de détection de ces communautés.

Dans le deuxième chapitre, nous avons présenté une vue d'ensemble des concepts fondamentaux liés à l'influence et à l'importance des nœuds au sein des réseaux. En nous appuyant sur l'identification des nœuds significatifs et sur la tâche de détection des communautés qui y est associée, nous avons élaboré en détail les mesures de densité que nous utilisons pour évaluer la quantité de nœuds impliquée dans notre approche.

Après avoir introduit les notions clés de notre recherche, le troisième chapitre a présenté une nouvelle méthode de détection de communautés. Cette approche se fonde sur l'évaluation de la densité locale avec une itération et une fusion progressive des communautés en fonction de seuils de similarité enfin des communautés présentant des similitudes.

Dans le dernier chapitre, nous avons mis en œuvre l'algorithme que nous avons proposé et effectué des tests sur des réseaux réels. Nos résultats ont démontré l'efficacité de notre méthode pour détecter des communautés.

Lorsque nous avons comparé les résultats de notre approche à ceux d'autres méthodes, notre algorithme s'est avéré performant, fournissant des résultats de haute qualité.

De plus, il est caractérisé par sa simplicité et sa facilité de compréhension, tout en maintenant une bonne performance lorsqu'il est appliqué à des réseaux du monde réel. Les communautés identifiées restent stables même lors de plusieurs exécutions sur le même

réseau, ce qui souligne la robustesse de notre méthode. Enfin, la mise en œuvre de la méthode que nous avons proposée s'avère être très facile à réaliser.

Les communautés trouvées par notre méthode restent stables dans plusieurs exécutions sur le même réseau et la mise en œuvre de la méthode proposée est très facile.

Perspective

Dans le cadre des extensions potentielles de cette recherche, il serait intéressant d'explorer des variantes de notre méthode qui tiennent compte de la dynamique des réseaux, en particulier dans les réseaux sociaux en constante évolution. Les communautés peuvent changer avec le temps, et adapter notre méthode à cette dynamique serait une étape importante.

Bien que notre méthode soit efficace, l'optimisation de sa vitesse d'exécution pourrait être un objectif pour les futurs chercheurs. Cela serait particulièrement bénéfique dans le contexte de l'analyse en temps réel des réseaux en évolution.

En explorant comment l'apprentissage automatique peut être intégré à notre méthode, nous pourrions améliorer la précision de la détection des communautés et la capacité à gérer des données bruitées ou incomplètes.

Nous encourageons vivement les praticiens à explorer des applications de notre méthode dans des domaines tels que la détection de groupes d'utilisateurs similaires dans les réseaux sociaux, la surveillance de la sécurité des réseaux, ou même la recommandation de produits dans le commerce électronique.

Bibliographie

[1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics : Theory and Experiment*, vol. 2008, p. P10008, oct 2008.

[2] Neo4j, "Graph algorithms in neo4j : Label propagation," *Neo4j Blog*, Date de publication inconnue.

[3] N. Boucerredj, "Détection des communautés par une méthode d'apprentissage automatique," 2022.

[4] M. Talbi, *Une nouvelle approche de détection de communautés dans les réseaux sociaux*. PhD thesis, Université du Québec en Outaouais, 2013.

[5] N. Kasoro, S. Kasereka, E. Mayogha, T. Ho, and J. Kinganga, "Percomcv : A hybrid approach of community detection in social networks," vol. 151, 05 2019.

[6] "Graph of the dolphin network." `https://www.researchgate.net/figure/Graph-of-the-dolphin-network-where-the-red-nodes-are-selected-according-to-Algori fig1_324859794`, 2022. Visited on 10/06/2023.

[7] A. de l'article, "Finding central vertices and community structure via extended density peaks-based clustering," *Nom du journal*, Année de publication. Consulté le 8 Sep, 2023.

[8] A. de la figure, "Détection de communautés dans les grands graphes d'interactions (multiplexes) : état de l'art - scientific figure," Année de publication. Consulté le 8 Sep, 2023.

[9] "Network of the books about us politics created by krebs with 105 vertices, 441 edges." Available from ResearchGate, Accessed Year. Accessed on 8th September 2023.

[10] N. Kheloufi and Y. Leslous, *Outils de graphes pour l'analyse des réseaux sociaux*. PhD thesis, UMMTO, 2012.

[11] S. Fortunato and D. Hric, "Community detection in networks : A user guide," *Physics reports*, vol. 659, pp. 1–44, 2016.

[12] W. Louafi and F. Titouna, "Pcmeans : Community detection using localpagerank, clustering, and k-means," 2023.

[13] R. Kanawati, "Détection de communautés dans les grands graphes d'interactions (multiplexes) : état de l'art," 2013.

[14] W.-T. Wang, Y.-L. Wu, C.-Y. Tang, and M.-K. Hor, "Adaptive density-based spatial clustering of applications with noise (dbscan) according to data," in *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 1, pp. 445–451, IEEE, 2015.

[15] S. Wasserman and K. Faust, "Social network analysis : Methods and applications," 1994.

[16] C. Berge, *Theorie des graphes et ses applications : 2e ed.* Dunod, 1967.

[17] L. N. Elhouda, "Analyse des réseaux sociaux et détection de communautés," Master's thesis, Université 8 Mai 1945 - Guelma, 2019.

[18] J. Creusefond, *Caractériser et détecter les communautés dans les réseaux sociaux*. PhD thesis, Normandie Université, 2017.

[19] C. Bichot and N. Durand, "Partitionnement de graphe," *Lavoisier, Paris*, 2010.

[20] L. Beauguitte, "L'analyse des graphes bipartis," 2013.

[21] D. W. McMillan and D. M. Chavis, "Sense of community : A definition and theory," *Journal of community psychology*, vol. 14, no. 1, pp. 6–23, 1986.

[22] L. Despalatović, T. Vojković, and D. Vukicević, "Community structure in networks : Girvan-newman algorithm improvement," in *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 997–1002, 2014.

[23] M. Newman, "The function and structure of complex networks," *SIAM Rev.*, vol. 35, no. 2, pp. 167–256, 2003.

[24] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[25] J. Darlay, N. Brauner, and J. Moncel, "Partition en sous graphes denses pour la détection de communautés," in *ROADEF 2010, 11ème Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision France (2010)*, 2010.

[26] D. P. Joshi and K. K. Sutaria, "A comprehensive analysis of social network mining," *connections*, vol. 11, p. 15.

[27] M. Canu, *Détection de communautés orientée sommet pour des réseaux mobiles opportunistes sociaux*. PhD thesis, Université Pierre et Marie Curie-Paris VI, 2017.

[28] S. Gregory, "Finding overlapping communities in networks by label propagation," *New journal of Physics*, vol. 12, no. 10, p. 103018, 2010.

[29] Z. Xiaohu, S. Wenjun, W. Chongjun, and X. Junyuan, "Improved algorithm based on girvan-newman algorithm for community detection," *Journal of Frontiers of Computer Science & Technology*, vol. 4, no. 12, p. 1101, 2010.

[30] M. Zahiri, J. Mohammadzadeh, and S. Harifi, "An improved girvan–newman community detection algorithm using trust-based centrality," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2021.

[31] J. Ramsey, M. Glymour, R. Sanchez-Romero, and C. Glymour, "A million variables and more : the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images," *International journal of data science and analytics*, vol. 3, pp. 121–129, 2017.

[32] G. Palla, A.-L. Barabási, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.

[33] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations : can geographic isolation explain this unique trait ?," *Behavioral Ecology and Sociobiology*, vol. 54, pp. 396–405, 2003.

[34] J. Park and M. E. Newman, "A network-based ranking system for us college football," *Journal of Statistical Mechanics : Theory and Experiment*, vol. 2005, no. 10, p. P10014, 2005.

[35] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of anthropological research*, vol. 33, no. 4, pp. 452–473, 1977.

[36] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[37] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.

[38] Z. Yang, J. I. Perotti, and C. J. Tessone, "Hierarchical benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 96, no. 5, p. 052311, 2017.

[39] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo, "A comparison of extrinsic clustering evaluation metrics based on formal constraints," *Information retrieval*, vol. 12, pp. 461–486, 2009.

[40] E. Orsini, "Détection de communautés dans les flots de liens par optimisation de la modularité," in *6ème Conférence Modèles et Analyses Réseau : Approches Mathématiques et Informatiques (MARAMI 2015)*, 2015.

[41] A. F. McDaid, D. Greene, and N. Hurley, "Normalized mutual information to evaluate overlapping community finding algorithms," *arXiv preprint arXiv :1110.2515*, 2011.

[42] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846–850, 1971.

[43] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1978.

[44] S. Aral, L. Muchnik, and A. Sundararajan, "Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks," *Proceedings of the National Academy of Sciences*, vol. 106, no. 51, pp. 21544–21549, 2009.

[45] M. S. Granovetter, "The strength of weak ties [j]," *American journal of sociology*, vol. 78, no. 6, pp. 1360–1380, 1973.

[46] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, no. 4, pp. 581–603, 1966.

[47] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking : Bringing order to the web.," tech. rep., Stanford infolab, 1999.

[48] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world'networks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[49] A. Barabási, "Network science. cambridge university press, cambridge," 2016.

[50] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks : Structure and dynamics," *Physics reports*, vol. 424, no. 4-5, pp. 175–308, 2006.

[51] U. Brandes, "A faster algorithm for betweenness centrality," *Journal of mathematical sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[52] E. Britton, C. Rogerson, S. Mehta, Y. Li, X. Li, O. consortium, R. C. Fitzgerald, Y. S. Ang, and A. D. Sharrocks, "Open chromatin profiling identifies ap1 as a transcriptional regulator in oesophageal adenocarcinoma," *PLoS genetics*, vol. 13, no. 8, p. e1006879, 2017.

[53] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical review E*, vol. 76, no. 3, p. 036106, 2007.

[54] B. Albert, "Albert, réka-barabási, albert-lászló," *Statistical mechanics of complex networks*, pp. 47–97, 2002.

[55] K. Deturck, "Détection d'influenceurs dans des médias sociaux (influencer detection in social medias)," in *Actes de la Conférence TALN. Volume 2-Démonstrations, articles des Rencontres Jeunes Chercheurs, ateliers DeFT*, pp. 117–130, 2018.

[56] N. Tatti and A. Gionis, "Density-friendly graph decomposition," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 1089–1099, 2015.

[57] B. Daille and E. Morin, "Actes de la 10ème conférence sur le traitement automatique des langues naturelles. articles longs," in *Actes de la 10ème conférence sur le Traitement Automatique des Langues Naturelles. Articles longs*, 2003.

[58] D. Whitehead, W. R. N. Edwards, and P. G. Jarvis, "Conducting sapwood area, foliage area, and permeability in mature trees of picea sitchensis and pinus contorta," *Canadian journal of forest research*, vol. 14, no. 6, pp. 940–947, 1984.

[59] M. Boullé, "Estimation de la densité d'arcs dans les graphes de grande taille : une alternative à la détection de clusters.," in *EGC*, pp. 353–364, 2011.

[60] M. Barjon, A. Casteigts, S. Chaumette, C. Johnen, and Y. M. Neggaz, "Un algorithme de test pour la connexit\'e temporelle des graphes dynamiques de faible densit\'e," *arXiv preprint arXiv :1405.0170*, 2014.

[61] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics : Ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60,

1999.

[62] P. Batra, "Comparative study of density based clustering algorithms," 2018.

[63] Python, "Networkx documentation." Consulté le 9 septembre 2023.

[64] "Python . tkinter python."

[65] "Python . matplotlib python."

[66] "Python . community python."

[67] "Python . scikit-learn python."

[68] F. Parés, D. G. Gasulla, A. Vilalta, J. Moreno, E. Ayguadé, J. Labarta, U. Cortés, and T. Suzumura, "Fluid communities : A competitive, scalable and diverse community detection algorithm," in *Complex Networks & Their Applications VI : Proceedings of Complex Networks 2017 (The Sixth International Conference on Complex Networks and Their Applications)*, pp. 229–240, Springer, 2018.