People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research scientifique

University of 8 May 1945-Guelma-

Faculty of Mathematics, Computer Science and Science of Matter

Department of Computer Science

Master Thesis

Specialty : Computer Science

Option :

Science and information and communication

Theme

Détection des communautés par le PageRank

Presented by :

OURAHMANE Yousra

supervised by :

Dr.LOUAFI Wafa

Jury Members

Chairman : Dr.CHOHRA Chemseddine
Examinee : Dr.BENHAMDI Soulef

Jun 2023.

Remerciement

Avant tout, nous tenons à exprimer notre gratitude envers Dieu le Tout-Puissant qui nous a accordé la force et le courage nécessaires pour mener à bien ce modeste travail.
Je tiens à exprimer ma sincère gratitude au superviseur,**Dr. LOUAFI Wafa**, pour son soutien et son conseil précieux tout au long de l'écriture de ce mémoire. Vos conseils et votre expertise ont été inestimables pour la réussite de ce projet.
De plus, je tiens à exprimer ma gratitude sincère à **Dr. CHOHRA Chemseddine**, qui est le président du comité, pour son leadership et sa grande contribution à l'évaluation de ce mémoire. Votre expertise et votre dévouement à l'excellence académique ont été extrêmement bénéfiques. Enfin, je tiens à exprimer ma gratitude au **Dr. BENHAMDI Soulef**, examinateur, pour votre temps, votre expertise et vos commentaires utiles lors de l'examen de ce mémoire. Votre contribution a été essentielle à son amélioration.
Nous souhaitons également exprimer notre profond amour et notre reconnaissance infinie envers nos parents. Leur contribution, leur soutien inconditionnel et leur patience exemplaire ont été des piliers solides sur lesquels nous avons pu nous appuyer. Leur confiance en nos capacités et leur encouragement constant ont été une source d'inspiration indéniable.
Enfin, nous tenons à remercier chaleureusement tous les enseignants qui ont contribué à notre formation. Leur dévouement à transmettre leur savoir et leur investissement dans notre réussite ont été essentiels pour notre parcours académique. Nous sommes également reconnaissants envers toutes les personnes qui ont participé, de près ou de loin, à la réalisation de ce travail, que ce soit par leur soutien moral, leurs conseils avisés ou leur contribution matérielle.

Dédicace

Avant tout, je remercie mon Dieu qui m'a donné la force et la patience que je peux faire ce travail.

Je désire exprimer mon profond amour et ma gratitude aux plus chers : Mes parents, pour leurs prières, leur amour, leur soutien et leurs sacrifices pour moi. Vous êtes mon inspiration et ma force motrice.

À mes frères, [Nouh, Linda, Meriem, Abdelrahman] pour leurs prières, leur amour et leur soutien. Vous aimez les gens à mon cœur.

À mon amie Meriem, pour leur soutien moral et leurs encouragements qui ont réduit ma charge scolaire.

Enfin, je dédie ce message à toutes les personnes qui ont traversé mon chemin et contribué directement ou indirectement à mon parcours éducatif. Votre soutien et votre influence ne seront pas oubliés dans ma vie.

Résumé

La détection de communautés dans les réseaux est cruciale pour comprendre leur structure. Dans ce mémoire, une nouvelle approche de détection de communautés est proposée, qui utilise les pageranks des noeuds pour initialiser les noeuds centraux et effectuer un regroupement. Cette méthode est simple, efficace et facile à mettre en œuvre. Les résultats de cette approche ont été comparés avec quelques algorithmes de pointe en utilisant la modularité Q et la conductance comme mesures d'évaluation sur des réseaux synthétiques et réels. Les résultats obtenus sont acceptables.

Mots clés :

Détection des communautés, Pagerank, Regroupement, modularité, conductance.

Abstract

The detection of communities in networks plays a crucial role in understanding their structures. Applying pagerank to the task of community detection in complex networks can yield good results. In this thesis, we propose a new approach to community detection based on the most influential nodes, initialized using their pageranks, and then clustering them. Our approach is effective, simple, and easy to implement. We compared our algorithm with some state-of-the-art algorithms on synthetic and real networks, using evaluation measures such as Modularity Q and Conductance, and obtained acceptable results.

Mots clés :

Community detection, Pagerank, clustering, modularity, conductance.

Table des matières

Table des figures

Liste des tableaux

Introduction

Les réseaux complexes jouent un rôle vital dans de nombreux domaines, tels que les réseaux sociaux, les systèmes biologiques, les infrastructures de transport et même Internet lui-même. Ces réseaux sont constitués de nœuds (entités) reliés par des liens (relations), qui peuvent représenter des amitiés, des interactions, des dépendances ou tout autre type de connexion. Comprendre la structure et les motifs cachés de ces réseaux est une tâche cruciale pour de nombreuses applications, de la recommandation de contenu à l'analyse des épidémies.[3]
Dans notre étude, nous nous intéressons particulièrement à la détection de communauté dans les réseaux complexes. Une communauté peut être définie comme un groupe de nœuds qui ont de fortes interactions entre eux mais des interactions plus faibles avec des nœuds extérieurs à la communauté [7]. La détection communautaire permet de découvrir des structures sous-jacentes, d'identifier des groupes fonctionnels ou sociaux et de comprendre la dynamique collective d'un réseau[19].
Notre objectif principal était d'étudier l'utilisation du PageRank, un algorithme largement utilisé pour classer l'importance des nœuds dans un réseau[20], dans le contexte de la détection de communauté. Nous nous sommes appuyés sur des recherches existantes montrant que le PageRank peut être adapté pour détecter les structures communautaires dans un réseau en fonction de la hiérarchie d'importance des nœuds.
Dans notre étude, nous avons abordé plusieurs chapitres pour examiner en détail la détection des communautés par le PageRank.

Le premier chapitre est une introduction approfondie à la détection de communauté. Nous commençons par présenter une représentation simple des quatre communautés, ce qui facilite la compréhension des concepts clés. Ensuite, nous explorons différentes approches et algorithmes de détection de communauté, tels que la modularité, la propagation de

balises, la détection de modèles et les techniques de clustering. Pour évaluer la qualité de la détection, nous discutons également des méthodes d'évaluation, y compris les normes de qualité Internet, les normes de qualité externes et la stabilité de la communauté.

Le deuxième chapitre est spécifiquement axé sur l'algorithme PageRank, nous allons explorer en détail cet algorithme utilisé par les moteurs de recherche pour évaluer la pertinence des pages web. Nous commencerons par définir le PageRank et expliquerons sa formule de calcul. Ensuite, nous discuterons des propriétés du PageRank et des différentes approches qui ont été utilisées pour l'améliorer. Nous aborderons également les perspectives d'avenir du PageRank et son utilisation dans la détection de communautés. Enfin, nous comparerons le PageRank à d'autres algorithmes qui ont été proposés et appliqués avant lui.

Le troisième chapitre traite de la conception de notre système de détection communautaire. Nous décrivons en détail la problématique, les objectifs et les phases de notre travail. Nous présentons les différentes étapes du processus en détaillant notre approche proposée. Nous mettons également en évidence les fonctionnalités que nous avons développées et utilisées tout au long du projet et décrivons le programme que nous avons mis en œuvre. Des exemples d'illustrations spécifiques sont donnés pour faciliter la compréhension des concepts présentés.

Le dernier chapitre est consacré à l'environnement de travail de notre projet. Nous présentons les matériaux environnementaux que nous avons utilisés, la logique environnementale que nous avons adoptée, ainsi que la plate-forme et l'IDE (environnement de développement intégré) qui ont été utilisés. Nous mentionnons également les bibliothèques qui ont été utilisées pour mettre en œuvre notre système de détection de communauté. Enfin, nous présentons les résultats de nos expériences et analyses, en discutant des expériences réalisées sur des réseaux réels et des réseaux artificiels. Nous discutons également de la complexité de l'algorithme PageRank ainsi que des avantages et inconvénients de notre méthode de détection de communauté.

Chapitre 1

Détection des comunautés

1.1 Introduction

Ce chapitre se concentre sur la détection de communauté dans les réseaux complexes. Premièrement, nous définissons ce qu'est une communauté dans le contexte de réseaux complexes. Examinez les caractéristiques et les traits qui définissent une communauté, tels que : Densité de connexion interne et faibles connexions externes.

Nous considérons ensuite diverses approches et algorithmes pour détecter les communautés. Nous introduisons une méthode basée sur des modules visant à maximiser la mesure de la qualité intrinsèque pour identifier la structure de la communauté. De plus, sur la base de la propagation des étiquettes, nous expliquons comment attribuer des étiquettes aux nœuds en fonction de leur proximité et de leurs relations.

Nous décrivons également une méthode basée sur la reconnaissance de formes pour identifier des sous-graphes répétés d'éléments dans des réseaux. De plus, nous décrivons des méthodes basées sur des techniques de clustering qui regroupent les nœuds en fonction de la similarité structurelle ou attributaire.

Examinons maintenant différentes méthodes d'évaluation de la détection de communauté. Nous présentons une métrique de qualité interne qui mesure l'adhésion et la dissociation des communautés détectées, et une métrique de qualité externe qui compare la détection des communautés à des références. Enfin, nous discutons du concept de stabilité dans la découverte de la communauté. Nous présentons une méthode bootstrap qui évalue la cohérence des communautés détectées par un échantillonnage aléatoire répété du réseau.

Ensuite, nous décrivons une méthode de perturbation qui mesure la stabilité de la détection communautaire en perturbant le réseau et en comparant les résultats avant et après la perturbation.

Enfin, ce chapitre donne un aperçu des différentes approches et algorithmes utilisés pour détecter les communautés dans les réseaux complexes. Les méthodes d'évaluation et de mesure de la qualité et le concept de stabilité dans la découverte de la communauté sont également explorés. Cette compréhension approfondie de la détection communautaire est essentielle pour l'analyse et l'interprétation des structures organisationnelles et des dynamiques sociales dans divers domaines de recherche.

1.2 Définition d'une communauté

Une communauté peut être définie comme un sous-ensemble de noeuds ou individus dans un réseau qui sont étroitement connectés les uns avec les autres. Les membres une communauté partagent souvent des caractéristiques, des intérêts, des objectifs ou des affiliations similaires, ce qui les fait former un groupe distinct au sein du réseau plus large.[18]

Les communautés, dans tous les cas, sont formées par des liens forts au sein du groupe et des liens faibles avec des nœuds extérieures à la communauté.[7]

Les communautés peuvent varier en taille et en organisation. Certaines communautés sont hiérarchiques, est-à-dire il existe des sous-communautés plus petites au sein une communauté plus importante. Il existe également des types de communautés qui ne se chevauchent pas, où un nœud appartient à une seule communauté, et des types de communautés qui se chevauchent, où un nœud appartient à plusieurs communautés en même temps.[7]

Il est important de noter que la détection des communautés est pas toujours une tâche claire et objective car les perspectives et les définitions de ce est une communauté peuvent varier. est pourquoi il existe plusieurs approches et algorithmes pour détecter les communautés, chacun ayant ses propres avantages et limites.

1.2.1 Représentation simple de quatre communautés

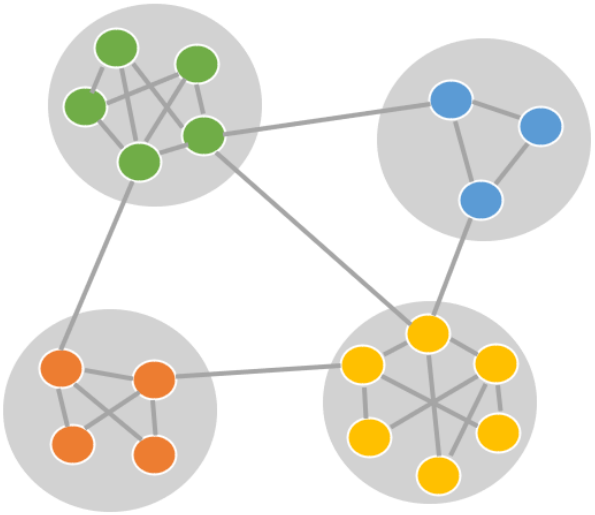

FIGURE 1.1 – Figure de quatre communautés simple

1.3 Les approches et les algorithmes de détection des communautés

Diverses méthodes et algorithmes ont été développés pour localiser ces communautés. Ces techniques recherchent les ensembles cohérents de nœuds qui forment des communautés distinctes en examinant leurs motifs, leurs propriétés topologiques et leurs caractéristiques.

Il existe plusieurs approches et algorithmes pour détecter les communautés dans les réseaux. Voici quelques-unes des méthodes les plus couramment utilisées :

1.3.1 Méthodes basées sur la modularité

Les méthodes basées sur la modularité sont largement utilisées pour détecter les communautés en réseaux complexes. La modularité mesure à quel point une partition d'un réseau en communautés est meilleure que ce à quoi on s'attendrait par hasard. Plus, a modularité is an élevée and la structure du Réseau est bien d'ouverture in communautés specifices.

— **Algorithme de Louvain**

L'algorithme de Louvain est l'un des algorithmes de détection de communauté les plus populaires basés sur la modularité. Maximisez la modularité en fusionnant et en déplaçant les nœuds entre les communautés à l'aide d'une approche itérative. L'algorithme augmente de manière itérative la modularité en attribuant d'abord chaque nœud à sa propre communauté, puis en fusionnant les communautés adjacentes, ce qui se traduit par une plus grande modularité globale. Ce processus est répété jusqu'à ce qu'une nouvelle fusion n'améliore pas la modularité.[2]

— **Méthode de recherche exhaustive**

Une méthode de recherche complète examine en profondeur toutes les partitions possibles du réseau pour trouver la partition avec la plus grande modularité. Cela signifie que toutes les combinaisons de nœuds de division en communautés sont évaluées. Cela peut être très coûteux en calcul dans les grands réseaux. Cependant, cette approche garantit que vous trouverez la partition optimale en termes de modularité.[17]

— **Méthode de recuit simulé**

La méthode de recuit simulé est une approche heuristique qui trouve la partition optimale par exploration probabiliste de l'espace des solutions. Inspiré des processus de lueur en physique. L'algorithme démarre à partir de la première partition et passe d'une partition à l'autre à l'aide de jointures de communauté mobiles et de scissions de communauté. La probabilité de passer à une nouvelle partition est déterminée par la différence de modularité entre la nouvelle partition et la partition actuelle, et le paramètre de température qui contrôle l'exploration de l'espace des solutions. Le processus itératif se poursuit jusqu'à ce que la température atteigne un seuil défini, permettant à l'algorithme d'explorer à la fois des solutions optimales et sous-optimales.[10]

Ces techniques basées sur des modules sont appréciées pour leur capacité à reconnaître des communautés de différentes tailles et formes et leur flexibilité lorsqu'elles sont appliquées à différents types de réseaux. Cependant, la maximisation de la modularité peut conduire à un fractionnement sous-optimal, et la résolution de la détection de communauté peut être

affectée par plusieurs facteurs tels que la taille du réseau et la résolution des communautés recherchées.[8]

1.3.2 Méthodes basées sur la propagation d'étiquettes

Les méthodes basées sur la propagation d'étiquettes sont des approches pour détecter les communautés au sein des réseaux. Deux de ces méthodes sont décrites ici.

— **Algorithme de propagation de l'étiquette (Label propagation algorithm)**

Algorithme de propagation des étiquettes : cet algorithme fonctionne en attribuant d'abord des étiquettes aux nœuds du réseau. Ces balises deviennent des identifiants ou des catégories qui représentent votre communauté. L'algorithme itère ensuite encore et encore, en passant des étiquettes entre les nœuds adjacents en fonction de la similarité des attributs ou des liens. Plus précisément, chaque nœud met à jour son étiquette en fonction des étiquettes des nœuds voisins en utilisant des règles de propagation. Ce processus de propagation est répété jusqu'à ce qu'un critère d'arrêt prédéfini soit satisfait, par exemple lorsque la majorité des nœuds ont convergé vers une étiquette stable. L'algorithme de propagation des balises est relativement rapide et efficace pour détecter les communautés sur les grands réseaux.[7]

— **Algorithme de propagation de la chaleur (Heat diffusion algorithm)**

Algorithme de diffusion thermique : Cet algorithme utilise le principe de diffusion thermique pour détecter les communautés dans le réseau. L'idée est que la distribution de chaleur à travers le réseau est affectée par la connectivité locale, et cette connectivité peut être utilisée pour identifier les communautés. L'algorithme commence par allouer une certaine quantité de chaleur à chaque nœud du réseau. La chaleur est ensuite propagée à plusieurs reprises entre les nœuds adjacents en utilisant des règles basées sur la connectivité du réseau. En particulier, la chaleur est plus facilement conductrice entre des nœuds fortement connectés et moins facilement entre des nœuds faiblement connectés. En observant la répartition de la chaleur après un certain nombre d'itérations, des communautés au sein du réseau

peuvent être identifiées. Les zones du réseau avec des concentrations de chaleur plus élevées peuvent desservir différentes communautés. L'algorithme de propagation de la chaleur est également relativement rapide et peut être utilisé pour la détection de communauté sur de grands réseaux.[12]

Ces deux méthodes basées sur la propagation d'étiquettes offrent des approches alternatives pour détecter les communautés dans les réseaux, basées sur des principes différents des méthodes basées sur les modules.

1.3.3 Méthodes basées sur la détection de motifs

Les méthodes basées sur des modèles sont des méthodes permettant de découvrir des communautés dans des réseaux. Deux de ces méthodes sont décrites ici.

— **Algorithme de détection de cliques (Clique percolation algorithm)**

Cet algorithme se concentre sur la recherche et l'analyse des cliques, qui sont des sous-graphes complets dans lesquels chaque nœud est directement connecté à tous les autres nœuds de la clique. L'algorithme analyse le réseau à la recherche de cliques et recherche les cliques qui se chevauchent, c'est-à-dire les nœuds qui existent dans plus d'une clique. Le chevauchement de ces cliques peut indiquer des zones de chevauchement entre les communautés. En identifiant les factions et leurs intersections, l'algorithme peut identifier les communautés qui existent au sein du réseau.[21]

— **Algorithme de détection de motifs structuraux (Structural motif detection algorithm)**

cette méthode se concentre sur l'identification des modèles structurels caractéristiques dans les réseaux qui sont souvent connectés aux communautés. Un modèle de structure est un modèle répétitif ou une configuration spécifique de connexions dans un réseau. Cet algorithme recherche ces modèles sur le Web et les utilise comme signatures pour identifier les communautés. Les motifs de texture incluent des anneaux, des étoiles, des bords de lignes, etc. En identifiant ces modèles caractéristiques, les algorithmes peuvent identifier les communautés qui présentent ces

modèles de connectivité particuliers.[13]

Ces deux approches, basées sur la reconnaissance de formes, offrent des alternatives pour reconnaître les communautés au sein des réseaux. Les communautés sont utiles lorsqu'elles sont liées à des structures spécifiques ou à des schémas répétitifs au sein du réseau.

1.3.4 Méthodes basées sur les techniques de clustering

— **Algorithme de clustering spectral**]

L'algorithme de clustering spectral est une méthode de détection de communauté basée sur la décomposition spectrale de cartes. L'idée principale est d'utiliser les vecteurs propres de la matrice d'adjacence d'un graphe ou de la matrice laplacienne pour représenter les nœuds dans un espace de dimensionnalité réduit. Ces vecteurs de caractéristiques sont utilisés pour regrouper les nœuds en communautés. L'algorithme calcule d'abord le vecteur propre correspondant à la plus petite valeur propre du diagramme. Ces vecteurs de caractéristiques sont ensuite utilisés pour projeter les nœuds dans un nouvel espace où les nœuds appartenant à la même communauté sont regroupés de manière compacte. Dans ce domaine, des techniques de clustering classiques telles que K-means peuvent être utilisées pour maintenir les communautés.[14]

— **Algorithme de k-moyennes (k-means algorithm)**]

L'algorithme K-Means, également connu sous le nom de K-Means, est une technique de clustering largement utilisée. Regroupe les nœuds en fonction de la similarité des attributs. L'algorithme commence par définir le nombre de clusters k et initialise aléatoirement les centres de ces clusters. Chaque nœud est ensuite affecté au centre de cluster le plus proche en termes de similarité d'attribut. Une fois tous les nœuds affectés aux clusters, les centres de cluster sont mis à jour en faisant la moyenne des attributs des nœuds appartenant à chaque cluster. Ce processus d'attribution et de mise à jour des centres de cluster est répété jusqu'à ce que les centres de cluster convergent et que l'inertie intra-cluster, qui mesure la propagation des nœuds dans chaque cluster, soit minimisée.[1]

En bref, les algorithmes de clustering spectral reposent sur la décomposition spectrale

d'un graphe en nœuds de cluster, tandis que les algorithmes K-means exploitent les similitudes d'attributs avec les nœuds de cluster. Les deux approches sont efficaces pour reconnaître les communautés au sein des réseaux et offrent des points de vue différents sur la représentation et les mesures de similarité.[24]

Notez que chaque méthode présente des avantages, des limites et des domaines d'application spécifiques. Le choix de l'approche ou de l'algorithme dépend des caractéristiques du réseau, des objectifs de détection de la communauté et des limites de calcul. Certaines méthodes conviennent aux grands réseaux, tandis que d'autres conviennent à l'identification de communautés ayant des caractéristiques spécifiques.

1.4 Méthodes d'évaluation de détection communauté

L'évaluation de la détection communautaire est un processus fondamental pour évaluer l'efficacité des algorithmes de détection communautaire dans les réseaux complexes. Voici quelques façons courantes d'évaluer la détection de communauté.

1.4.1 Métriques de qualité interne

— **Modularité**

La modularité mesure l'écart de la structure de la communauté détectée par rapport à la distribution aléatoire du réseau. Comparez le nombre de sauts dans la communauté avec le nombre de sauts attendus dans un réseau aléatoire. Une valeur de modularité élevée indique que le réseau est segmenté avec plus de précision, les nœuds étant fortement connectés au sein des communautés et faiblement connectés entre les communautés. La modularité est calculée à l'aide de diverses formules, y compris la formule originale de Newman-Girvan et les formules développées par Clauset, Newman et Moore (CNM).[17]

Voici la formule de modularité pour un graphe non orient :

$$Q = \frac{1}{2m} \sum\_ \left(A\_ij - \frac{k_i k_j}{2m} \right)$$

où :

— **Q** est la modularité du graphe.

— **A__ij** représente la présence ou l'absence de l'arête entre les nœuds i et j (1 si l'arête existe, 0 sinon).

— **k__i** est le degré du nœud i (le nombre d'arêtes connectées à ce nœud).

— **m** est le nombre total d'arêtes dans le graphe.

— **Conductance**

La conductance évalue la séparation entre les communautés détectées en mesurant la proportion d'arêtes qui relient les nœuds à l'extérieur de leur communauté par rapport au total des arêtes du nœud. Une valeur faible de conductance indique une meilleure délimitation des communautés, avec une forte concentration des arêtes à l'intérieur de chaque communauté et peu d'arêtes reliant les communautés. La conductance peut être calculée pour chaque communauté individuelle ou pour l'ensemble du réseau.[22]

$$\text{Conductance}(C) = \frac{\text{Nombre d'arêtes coupant la communauté } C}{\text{Minimum}(\text{Volume}(C), \text{Volume}(\overline{C}))}$$

où :

— **Conductance(C) :** représente la conductance de la communauté .

— **Le nombre d'arêtes coupant la communauté (C) :** est la somme des arêtes ayant une extrémité dans C et l'autre en dehors de C.

— **Volume(C) :** est le nombre total d'arêtes ayant les deux extrémités dans C.

— $\overline{C}$ représente le complément de la communauté C dans le graphe.

— **Volume**($\overline{C}$) est le nombre total d'arêtes ayant les deux extrémités en dehors de C.

— **Minimum**(**Volume**(C), **Volume**($\overline{C}$)) **:** est le minimum entre le volume de la communauté C et le volume de son complément $\overline{C}$.

— **Coefficient de regroupement**

Le coefficient de regroupement mesure le degré de regroupement des nœuds au sein d'une communauté en évaluant la densité des connexions locales dans chaque communauté. Celui-ci est calculé en comparant le nombre de triangles observés dans

une communauté donnée (ensemble de trois nœuds connectés) avec le nombre de triangles attendus dans un réseau aléatoire. Un coefficient de cluster élevé indique une structure plus dense au sein des communautés avec de nombreuses connexions locales. Les coefficients de regroupement peuvent également être calculés pour l'ensemble du réseau ou pour chaque communauté individuelle.[25]

Ces métriques de qualité internes indiquent la structure et la cohésion des communautés détectées. Ils aident à évaluer la qualité intrinsèque de la perception de la communauté sans avoir besoin d'une analyse comparative. Cependant, il est important de noter que le choix de la meilleure métrique de qualité interne dépend du contexte spécifique de votre application et de l'objectif de votre analyse.

1.4.2 Métriques de qualité externe

Des métriques de qualité externes sont utilisées pour évaluer les performances des algorithmes de détection de communauté en comparant les fragments détectés à des fragments de référence connus. Ces métriques mesurent la similarité des partitions détectées avec une partition de référence et aident à évaluer la précision et la qualité de la détection de la communauté.

— **Indice de similarité**

L'indice de similarité est une métrique couramment utilisée pour évaluer la similarité entre les partitions perçues et les partitions de référence. Calculez le pourcentage de nœuds répartis uniformément sur les deux partitions. Plus cet indice est élevé, plus la partition reconnue est similaire à la partition de référence. [25]

— **Score de précision et de rappel**

Les scores de précision et de rappel sont une autre mesure de qualité externe qui évalue les performances de reconnaissance de la communauté. Comparez les partitions reconnues avec les partitions de référence pour garantir l'exactitude et la reconnaissance. La précision mesure la proportion de nœuds correctement identifiés au sein d'une communauté au sein de la partition reconnue, et le rappel mesure la proportion de nœuds correctement identifiés au sein d'une communauté au sein de la partition de référence. Des scores de précision et de rappel élevés indiquent une

reconnaissance communautaire précise. [26]

— **Mesures de recouvrement**

Des mesures de récupération telles que l'indice Jaccard et l'indice Rand sont également utilisées pour comparer la similitude entre deux partitions. L'indice de Jaccard est calculé en divisant le nombre de paires d'éléments affectés de manière identique dans les deux partitions par le nombre total de paires d'éléments affectés de manière différente ou similaire dans les deux partitions. L'indice Rand est calculé en calculant le nombre de paires d'éléments affectées de la même manière et le nombre de paires d'éléments affectées de manière différente divisé par le nombre total de paires d'éléments dans les deux partitions. Mesure la similarité globale entre deux partitions. [16]

Ces mesures de qualité externes vous permettent d'évaluer objectivement les performances de votre algorithme de détection de communauté en le comparant à des scores de référence connus. Ils sont utiles pour mesurer la précision, le rappel et la similarité entre les partitions et permettent une évaluation quantitative de la qualité de la reconnaissance communautaire.

1.4.3 Stabilité

La stabilité est une mesure de la robustesse de la détection de communauté qui évalue la cohérence des résultats obtenus sur plusieurs chemins ou perturbations du réseau. Deux méthodes d'évaluation de la stabilité couramment utilisées sont la méthode bootstrap et la méthode de perturbation.

— **Méthode de bootstrap**

Les méthodes bootstrap consistent à effectuer un échantillonnage aléatoire répété du réseau. A chaque itération, une sous-section aléatoire du réseau est sélectionnée et une détection de communauté est effectuée sur cette sous-section. Répéter ce processus encore et encore crée plusieurs divisions de la communauté. La stabilité est ensuite évaluée en mesurant la cohérence des communautés trouvées à travers différentes itérations. Si les communautés détectées sont similaires dans la plupart des itérations, cela indique une bonne stabilité de la détection des communautés. [23]

— **Méthode de perturbation**

Les méthodes de perturbation le réseau en supprimant de manière aléatoire des nœuds ou des bords. Chaque fois qu'une interruption se produit, la détection de communauté est exécutée sur le réseau interrompu et les résultats sont comparés avec les résultats avant l'interruption. Si les divisions de la communauté sont similaires avant et après l'interruption, cela indique une bonne stabilité de la détection de la communauté face aux interruptions du réseau. [9]

Les deux méthodes peuvent être utilisées pour évaluer la stabilité de la détection communautaire en mesurant la cohérence des résultats dans différentes conditions. Une stabilité élevée indique une détection de communauté robuste qui est moins sensible aux changements aléatoires et aux pannes de réseau.

En résumé, Il est important de noter que différentes métriques peuvent être appropriées en fonction du contexte spécifique de l'application et des caractéristiques du réseau étudié. Par conséquent, il est recommandé d'utiliser plusieurs métriques pour obtenir une évaluation plus complète et robuste de la détection de communautés.

1.5 Conclusion

En conclusion, ce chapitre a fourni un aperçu complet des différentes facettes de la détection des communautés. Il a abordé la définition des communautés, présenté des approches et des algorithmes de détection, discuté des méthodes d'évaluation et de mesure de la qualité, et exploré le concept de stabilité. Ces connaissances sont essentielles pour comprendre les structures complexes et les interactions au sein des réseaux, ouvrant ainsi de nouvelles perspectives pour de nombreuses disciplines telles que la sociologie, la biologie, l'informatique et bien d'autres.

Chapitre 2

PageRank

2.1 Introduction

Ce chapitre examine le concept de PageRank, une mesure utilisée pour évaluer l'importance des pages Web. Nous définirons le PageRank avant d'expliquer la formule utilisée pour le calculer. Ensuite, nous parlerons des caractéristiques du PageRank et des différentes méthodes qui peuvent être utilisées pour l'augmenter. Enfin, nous examinerons quelques travaux de recherche sur la relation entre le PageRank et la détection de communautés. De plus, nous comparerons ces travaux afin de mieux comprendre les différentes méthodes utilisées.

2.2 Définition PageRank

Le PageRank et la structure du réseau sont des concepts étroitement liés dans le contexte de la découverte communautaire. Le PageRank est un algorithme développé par Larry Page et Sergey Brin, les fondateurs de Google, pour juger de l'importance des pages Web en fonction de leur structure de liens. Il a été conçu à l'origine pour classer les pages dans les résultats de recherche en fonction de leur pertinence et de leur popularité. [4]

Le calcul du PageRank est basé sur le principe qu'une page est considérée comme importante s'il y a des liens vers elle à partir d'autres pages tout aussi importantes. La note PageRank d'une page est donc déterminée par la quantité et la qualité des liens entrants qui y mènent. [4]

Dans le cadre de la détection de communauté, le PageRank peut être utilisé pour identifier les nœuds importants ou centraux d'un réseau. En fait, les nœuds avec un PageRank élevé sont souvent les plus connectés et les plus influents au sein d'un réseau. Ils jouent un rôle important dans la structure globale du réseau. [4]

En termes de détection de communauté, la structure du réseau est prise en compte pour déterminer les relations entre nœuds et groupes de nœuds. Les communautés sont généralement formées lorsque les nœuds sont fortement connectés les uns aux autres au sein d'un groupe tout en ayant des connexions plus faibles avec les nœuds extérieurs au groupe. [4]

PageRank peut être utilisé pour mesurer l'importance relative des nœuds dans un réseau, ce qui peut être utile pour identifier les nœuds centraux dans les communautés. Les nœuds avec un PageRank élevé au sein d'une communauté peuvent être considérés comme des points d'entrée importants pour découvrir la structure de la communauté. [4]

Par conséquent, en utilisant le PageRank et l'analyse de la structure du réseau, il est possible de mettre en évidence les nœuds les plus importants et les communautés potentielles du réseau. Cette approche permet de prendre en compte l'influence des connexions entre nœuds et d'identifier des regroupements cohérents au sein du réseau. [4]

Cependant, il convient de noter que PageRank seul ne peut pas assurer une détection précise de la communauté. Il est souvent utilisé en combinaison avec d'autres méthodes et approches pour améliorer les résultats de détection. L'analyse de la structure du réseau est essentielle pour comprendre les communautés et identifier les nœuds centraux, mais d'autres facteurs tels que les attributs des nœuds et les modèles de connectivité doivent être pris en compte pour obtenir une détection plus complète et plus précise.

2.3 Formule de calcul du PageRank

Dans Un graphe non orienté est une structure mathématique utilisée pour représenter des relations symétriques entre des entités. Les arêtes ne sont pas directionnelles, ce qui signifie qu'elles relient les nœuds de manière bidirectionnelle. Ces graphes sont largement utilisés pour modéliser des amitiés dans les réseaux sociaux, des connexions dans les réseaux de transport, et d'autres relations où la direction n'a pas d'importance. Ils sont étudiés dans la théorie des graphes et l'analyse de réseaux, et offrent une base pour l'analyse et la

modélisation des interactions entre les entités. La formule PageRank est basée sur un algorithme itératif développé par les fondateurs de Google, Sergey Brin et Larry Page. La formule de calcule le PageRank d'un nœud dans un graphique non orienté est :

$$PR(A) = \frac{1-d}{N} + d \sum_{B \in Neighbors(A)} \frac{PR(B)}{deg(B)}$$

où :

- **PR(A) :** est le score de PageRank du nœud A.
- **d :** est un facteur d'amortissement (typiquement défini à 0,85).
- **N :** est le nombre total de nœuds dans le graphe.
- **Neighbors(A) :** représente l'ensemble des nœuds voisins de A.
- **PR(B) :** est le score de PageRank du nœud B.
- **deg(B) :** est le degré du nœud B (c'est-à-dire le nombre d'arêtes connectées à B).

Dans un graphe non orienté, les nœuds voisins de A sont simplement les nœuds connectés à A par une arête.

2.4 Propriétés de pagerank

PageRank présente de nombreux avantages, dont les suivants :

- **Pertinence des résultats :**Le PageRank permet de classer les pages web en fonction de leur pertinence en fonction de la structure des liens entrants. Cela permet aux utilisateurs d'obtenir des résultats de recherche plus pertinents et d'obtenir les informations qu'ils recherchent plus facilement.
- **Fiabilité des résultats :**Le PageRank évalue la qualité des liens entrants d'une page pour évaluer sa fiabilité. Par conséquent, les pages qui contiennent des liens provenant de sources reconnues auront un classement plus élevé. Cela réduit la présence de spam ou de résultats de recherche de mauvaise qualité.
- **Égalité et neutralité :**Le PageRank est basé sur des algorithmes objectifs qui mesurent la popularité et la qualité des pages web en fonction des liens entrants. De cette façon, vous évitez les préjugés subjectifs et vous assurez une évaluation juste et neutre du site.

— **Évolutivité :**PageRank est conçu pour fonctionner à grande échelle, ce qui signifie qu'il peut être appliqué à un grand nombre de pages web. Cela classe efficacement des milliards de pages et fournit des résultats de recherche cohérents et pertinents, même si le Web continue de se développer.

— **Utilisation dans d'autres domaines :**Le concept de PageRank a été développé pour inclure l'analyse des réseaux sociaux, la recommandation de produits et la détection de communautés. Cela démontre que le PageRank est polyvalent et applicable dans divers contextes.

2.5 PageRank et détection de communautés

Dans cette partie, nous allons présenter quelques travaux qui utilisent le PageRank dans la détection des communautés.

Dans l'article "Detecting communities in social networks using the PageRank algorithm with different damping factors" publié dans PloS One en 2017 [28], les auteurs ont proposé des modifications à l'algorithme PageRank en introduisant différents facteurs d'amortissement. Ces facteurs contrôlent la probabilité que les utilisateurs restent dans le réseau plutôt que de le quitter. Ils ont utilisé une approche itérative pour calculer le score PageRank de chaque nœud du réseau social, où les nœuds avec les scores les plus élevés sont considérés comme centraux et peuvent être utilisés pour identifier les communautés

L'efficacité de leur algorithme a été évaluée en utilisant plusieurs ensembles de données de médias sociaux réels, et les résultats ont montré qu'il était efficace pour détecter les communautés de médias sociaux. Cependant, il y a quelques inconvénients à leur proposition. Pour commencer, l'algorithme PageRank et sa version modifiée peuvent être affectés par la taille du réseau et la présence de noeuds isolés. De plus, le choix des facteurs d'amortissement influence l'interprétation subjective des résultats. Enfin, l'algorithme peut être coûteux pour les grands réseaux sociaux en termes de temps de calcul.

L'algorithme décrit dans l'article de Chen, Zhao et Cai (2009) intitulé "Une nouvelle méthode de détection de communautés dans des réseaux complexes en utilisant l'algorithme PageRank" utilise une approche basée sur l'algorithme PageRank pour détecter des communautés dans des réseaux complexes. L'algorithme se compose de plusieurs étapes. Tout

d'abord, il construit une matrice de transition à partir du réseau complexe donné. Ensuite, il calcule le score PageRank pour chaque nœud du réseau en appliquant l'algorithme PageRank à cette matrice. En utilisant une technique de seuillage, l'algorithme détecte les communautés en attribuant un seuil à chaque nœud en fonction de son score de PageRank. Les nœuds dont le score de PageRank dépasse leur seuil sont considérés comme membres potentiels de la communauté. Les nœuds qui se chevauchent sont ensuite agrégés pour former des communautés. Ce processus est itéré en mettant à jour les seuils des nœuds et en répétant les étapes de détection des communautés jusqu'à ce qu'aucun nouveau nœud ne soit ajouté à une communauté. Les résultats de l'algorithme sont précis car il identifie les nœuds centraux et regroupe les nœuds similaires dans des communautés. Cependant, il présente certaines limites, notamment l'impact des paramètres de seuillage sur les résultats de détection des communautés, son efficacité réduite pour détecter des communautés de petite taille ou des communautés qui se chevauchent fortement, et son coût en termes de temps de calcul pour les grands réseaux sociaux. (Chen, Zhao, Cai, 2009) [5].

Mustafa Hajij, Eyad Said et Robert [11] ont utilisé le vecteur PageRank comme mesure de centralité pour identifier les centroides initiaux dans un graphe, afin de les utiliser comme nombre prédéfini k dans l'algorithme de clustering k-means. Leur étude a montré que cet algorithme est efficace et présente plusieurs avantages. Tout d'abord, le vecteur PageRank peut être défini pour des graphes dirigés et non dirigés. De plus, étant conçu pour être calculé sur des graphes massifs, il offre une vitesse de traitement accrue. En outre, cet algorithme peut facilement être généralisé aux espaces métriques, ce qui le rend applicable à d'autres domaines. Enfin, il s'agit d'un algorithme simple et facile à utiliser.

Cependant, il y a quelques inconvénients à prendre en compte avec cet algorithme. En premier lieu, il est sensible aux paramètres. Les résultats de détection des communautés peuvent être affectés par les paramètres inappropriés de l'algorithme PageRank et de l'algorithme de regroupement k-means. De plus, l'algorithme peut avoir du mal à identifier les communautés de petite taille car elles peuvent avoir des scores PageRank faibles et être moins visibles dans la structure du réseau dans son ensemble. Finalement, l'algorithme de regroupement k-means a des limites, telles que sa sensibilité à l'initialisation des centres de cluster et sa convergence vers des valeurs stables.

Analyse des traveaux

- **Article 1**
  - Résultats : Efficace pour détecter les communautés de médias sociaux.
  - Complexité : Coûteux pour les grands réseaux sociaux en termes de temps de calcul.
  - Qualité : Les facteurs d'amortissement peuvent influencer l'interprétation subjective des résultats.
  - Avantages : Propose des modifications à l'algorithme PageRank pour améliorer la détection des communautés.
  - Inconvénients : Peut être affecté par la taille du réseau et la présence de nœuds isolés. Le choix des facteurs d'amortissement peut influencer l'interprétation subjective des résultats.
- **Article 2**
  - Résultats : Précis dans l'identification des nœuds centraux et la formation de communautés.
  - Complexité : Efficacité réduite pour détecter des communautés de petite taille ou des communautés qui se chevauchent fortement.
  - Qualité : Impact des paramètres de seuillage sur les résultats de détection des communautés.
  - Avantages : Utilise une approche basée sur l'algorithme PageRank pour détecter les communautés. Identifie les nœuds centraux et regroupe les nœuds similaires dans des communautés.
  - Inconvénients : Impact des paramètres de seuillage sur les résultats de détection des communautés. Efficacité réduite pour détecter des communautés de petite taille ou des communautés qui se chevauchent fortement.
- **Article 3**
  - Résultats : Efficace pour identifier les centroides initiaux dans un graphe.
  - Complexité : Simple et facile à utiliser.
  - Qualité : Sensible aux paramètres de l'algorithme PageRank et de l'algorithme de regroupement k-means.
  - Avantages : L'utilisation du vecteur PageRank pour identifier les centroides initiaux est adaptée aux graphes dirigés et non dirigés. Offre une vitesse de

traitement accrue. Peut être généralisé aux espaces métriques.

— Inconvénients : Sensible aux paramètres de l'algorithme PageRank et de l'algorithme de regroupement k-means. Difficulté à identifier les communautés de petite taille. Limites de l'algorithme de regroupement k-means.

Article	Temps d'exécution	Forces	Faiblesses
Article 1	Coûteux pour les grands réseaux sociaux en termes de temps de calcul	Efficace pour détecter les communautés de médias sociaux	Sensible à la taille du réseau et à la présence de nœuds isolés, interprétation subjective des résultats
Article 2	Coût en termes de temps de calcul pour les grands réseaux sociaux	Précis dans l'identification des nœuds centraux et la formation de communautés	Impact des paramètres de seuillage sur les résultats, efficacité réduite pour les petites communautés ou les communautés fortement chevauchantes
Article 3	Offre une vitesse de traitement accrue	Utilisation du vecteur PageRank pour identifier les centroides initiaux dans l'algorithme de clustering k-means	Sensible aux paramètres, difficulté à détecter les petites communautés, sensibilité à l'initialisation des centres de cluster dans l'algorithme de clustering k-means

TABLE 2.1 – Comparaison entre les traveaux

2.6 Conclusion

En conclusion, la détection de communautés à l'aide du PageRank est une méthode simple mais efficace pour identifier les communautés dans un réseau. Elle peut être utilisée en complément d'autres méthodes de détection de communautés pour améliorer la précision des résultats.

Sur la base des connaissances acquises dans ce chapitre, nous sommes en mesure de proposer une nouvelle méthode innovante pour traiter notre problème spécifique de détection de communautés. Cette méthode prendra en compte les avantages du PageRank, tout en surmontant ses limitations, elle sera présentée en détail dans les deux chapitres suivants., intitulé "Conception et Implémentation du Modèle Proposé".

Chapitre 3

Conception du système

3.1 Introduction

Dans ce chapitre, nous exposons une approche de détection de communautés disjointes dans les réseaux sociaux en utilisant une classification non supervisée des graphes basée sur l'identification des nœuds centraux ayant un PageRank élevé. Pour cela, nous utilisons les nœuds les plus influents comme points de regroupement pour leurs voisins. Bien que les premières communautés identifiées puissent être chevauchantes, nous proposons une phase d'élimination du chevauchement pour détecter des communautés disjointes. Cette méthode est applicable aux réseaux non orientés et non pondérés. Le chapitre est organisé en plusieurs sections : nous exposons les objectifs de notre système, détaillons son architecture, exposons les étapes de conception et présentons les algorithmes proposés. Enfin, nous concluons ce chapitre.

3.2 Problématique

La détection des communautés dans les réseaux complexes est une tâche cruciale dans de nombreux domaines, mais elle reste un défi en raison de la taille et de la complexité des réseaux, ainsi que de la diversité des structures topologiques des communautés. Il existe de nombreux algorithmes pour la détection des communautés, chacun avec ses avantages et ses limites. Comment choisir le meilleur algorithme pour la détection des communautés dans un réseau donné ? Comment évaluer la qualité de la détection des communautés ?

Dans ce contexte, nous proposons une approche pour la détection de communautés dans les réseaux complexes qui se base sur le PageRank et le regroupement pour réduire la complexité. Notre approche consiste à identifier les nœuds centraux du réseau, en utilisant l'algorithme PageRank, et à les utiliser comme points de regroupement pour leurs voisins. Cette méthode permet de réduire considérablement le nombre de nœuds à considérer pour la détection de communautés, ce qui réduit la complexité de la tâche. Cependant, cette approche a également ses limites et ses performances doivent être comparées à celles d'autres méthodes de détection de communautés dans les réseaux complexes.

3.3 L'objectif

Dans ce travail, nous proposons un système de détection de communautés disjointes dans les réseaux sociaux et complexes en utilisant le regroupement des nœuds selon leur PageRank. Cette approche vise à atteindre plusieurs objectifs :

- La détection efficace et rapide des communautés.
- La réduction du nombre d'itérations dans le processus de regroupement.
- Une méthode simple et facile à implémenter
- La sélection automatique du nombre de communautés.
- Applicabilité aux grands réseaux
- Réduction de la complexité.

3.4 Architucture du système

Notre approche comporte cinq parties principale :

- **Phase 01 :** cette étape se déroule en deux étapes :
  - Collecte de données
  - Construction du réseau
- **Phase 02 :** cette étape contient :
  - Calculez le score PageRank de chaque nœud du graphe
  - Mettre en ordre décroissant.
- **Phase 03 :** cette étape contient :

— Regroupement des nœuds ont les plus grands pagerank avec leurs voisins les plus proches jusqu'à ce que chaque nœud soit inclus dans au moins une communauté.

— **Phase 04 :** cette étape contient :

— Elimination de chevauchement des nœuds

— **Phase 05 :** cette étape contient :

— Identification des communautés disjointes finales

La figure 3.1 ci-dessous représentent le schéma général de notre approche

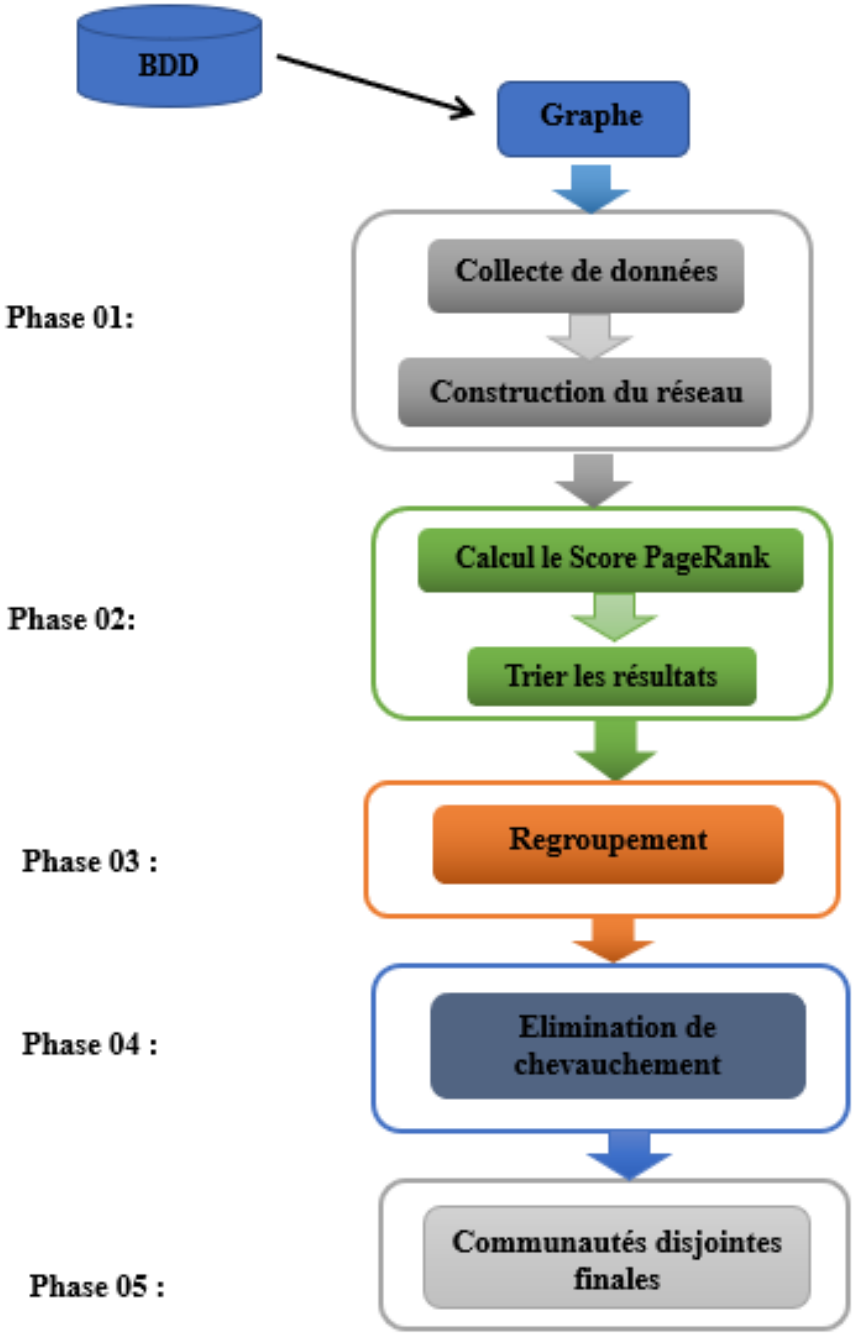

FIGURE 3.1 – Illustration de notre approche.

3.5 Conception détaillée de l'approche proposée

L'algorithme que nous avons proposé vise à détecter les communautés utilisant le système PageRank et le regroupement.
Notre algorithme se déroule en cinq phases :

3.5.1 Première phase

L'analyse de réseau et la détection de communautés sont des approches puissantes pour comprendre les structures et les relations complexes qui existent dans divers systèmes. Avant que ces méthodes puissent être appliquées, il est nécessaire de collecter des données pertinentes et de construire un réseau à partir de ces données. Cette étape du processus de détection de communauté se concentre sur la collecte de données réseau et la construction d'un graphique, où nous avons deux types des réseaux : collecter des données réseau réelles à partir d'un fichier ou créer un réseau artificiel.

- **Collecte de données d'un réseau réel :**
  À ce stade, sélectionnez le fichier contenant les données réelles du réseau. Après avoir sélectionné le fichier, lisez-le pour construire le réseau. Autrement dit, il lit les arêtes d'un fichier, les ajoute au diagramme et crée automatiquement les nœuds nécessaires.
- **Génération d'un réseau artificie (LFR Benchmark)**
  Dans cette partie, l'utilisateur doit spécifier le nombre de nœuds (n) et le paramètre epsilon (mu). Avec ces informations, la fonction crée un graphique vide. Les arêtes sont ensuite ajoutées au graphique selon une valeur de probabilité aléatoire (mu). Cela permet aux connexions entre les nœuds d'être générées de manière aléatoire et d'ajuster le degré de connexion réseau générée. Le réseau est ensuite stocké là où il sera utilisé dans les prochaines étapes du processus de détection de communauté.

En résumé, cette phase consiste à collecter les données du réseau, que ce soit à partir d'un fichier contenant les arêtes du réseau réel ou en générant un réseau artificiel avec un nombre spécifié de nœuds et un paramètre d'échantillonnage. Une fois les données collectées, il sera utilisé dans les étapes ultérieures du processus de détection des communautés

3.5.2 Deuxième phase

Le calcul du PageRank est la troisième étape du processus de détection de communautés à l'aide de l'algorithme PageRank. Cette étape consiste à calculer le PageRank pour chaque nœud dans le réseau, ce qui permet de déterminer l'importance relative de chaque nœud dans le réseau

- **Calcul du PageRank**

  Après avoir collecté les données et construit le réseau, l'étape suivante consiste à calculer le score PageRank pour chaque nœud du réseau. Le PageRank est un algorithme utilisé pour mesurer l'importance relative des nœuds dans un graphe en fonction de la structure du réseau. Il attribue un score plus élevé aux nœuds associés à un grand nombre de nœuds importants.
- **classer les noeuds du réseau par ordre décroissant du pagerank**

  Ensuite, les nœuds sont triés par ordre décroissant selon le score PageRank. Cette étape fournit une liste ordonnée des nœuds du réseau, du plus important au moins important en termes de score PageRank.

Le calcul du PageRank est essentiel pour la détection de la communauté car il permet d'identifier les nœuds clés qui ont un fort impact sur la structure du réseau. Ces nœuds peuvent jouer un rôle important dans la formation des communautés, et leur identification facilite le processus de regroupement des nœuds en communautés cohésives.

Brièvement, la phase de calcul du PageRank consiste à attribuer un score à chaque nœud du réseau en fonction de son importance relative. Cela permet d'identifier et d'utiliser les nœuds clés dans les étapes ultérieures du processus de découverte de la communauté.

3.5.3 Troisième phase

Une fois que les nœuds ont été triés par ordre décroissant de leur PageRank, le processus de regroupement commence en utilisant le premier nœud de la liste triée.

- Ce premier nœud est mis en communauté avec tous ses voisins
- Pour chaque nœud suivant dans la liste triée, s'il n'appartient pas déjà à une communauté existante, il est ajouté à une nouvelle communauté avec ses voisins directs.

— Si le nœud suivant appartient déjà à une communauté existante, il est ignoré et le processus passe au nœud suivant.
— Ce processus de clustering se poursuit jusqu'à ce que tous les nœuds du réseau aient été au moins dans une communauté.
— créant ainsi des communautés chevauchantes et connectées dans le réseau.

En résumé, la phase de regroupement consiste à parcourir les nœuds du réseau, en commençant par les nœuds les plus importants en termes de score PageRank, et à les assigner à des communautés. Les nœuds qui n'appartiennent à aucune communauté existante sont utilisés comme points de départ pour créer de nouvelles communautés. Les voisins directs des nœuds sont également inclus dans les communautés pour capturer les structures de voisinage.

3.5.4 Quatrième phase

Comment traiter les nœuds qui se chevauchent ?

a phase de gestion des nœuds chevauchants est une étape importante dans la détection des communautés dans le réseau. Cette phase vise à attribuer de manière cohérente les nœuds qui appartiennent à plusieurs communautés à une seule communauté, afin de réduire le chevauchement et d'améliorer la précision de détection des communautés.

Pour gérer les nœuds chevauchants, nous avons suivi les étapes suivantes :

1. Identification des nœuds chevauchants :
   Nous avons créé un ensemble distinct pour stocker les nœuds qui se chevauchent, c'est-à-dire ceux qui appartiennent à plusieurs communautés.
2. Recherche des communautés auxquelles appartiennent les nœuds chevauchants :
   Nous avons examiné la liste des communautés existantes pour déterminer dans quelles communautés se trouve le nœud chevauchant.
3. Sélection de la communauté cible :
   Si le nœud chevauchant appartient à plusieurs communautés, nous avons spécifié la communauté cible à laquelle le nœud sera attribué de manière persistante. Pour cela, nous avons comparé le nombre de voisins partagés par un nœud avec chaque communauté dans laquelle il se trouve. La communauté avec le plus grand nombre de voisins communs avec le nœud a été choisie comme communauté cible. Dans

le cas où plusieurs communautés ont le même nombre de voisins partagés avec le nœud, nous avons effectué une autre vérification pour choisir la communauté cible. Nous avons comparé le nombre total de nœuds dans chaque communauté et sélectionné celle qui avait le moins de nœuds parmi les communautés avec le même nombre de voisins partagés.

4. Attribution du nœud à la communauté cible :

   Une fois la communauté cible déterminée, le nœud a été retiré des autres communautés auxquelles il appartenait. Cela garantit que le nœud chevauchant est uniquement attribué à la communauté cible et élimine son appartenance à d'autres communautés. Pour faciliter la compréhension des résultats de la détection des communautés, nous avons affiché les nœuds chevauchants et les communautés identifiées. Chaque communauté a été numérotée et les nœuds qui en font partie ont été affichés. Cela permet à l'utilisateur de voir la composition de chaque communauté.

En résumé, la gestion des nœuds chevauchants est une étape cruciale pour la détection précise des communautés dans le réseau. En attribuant systématiquement les nœuds chevauchants à une seule communauté, nous réduisons le chevauchement entre les communautés et améliorons la précision de la détection des communautés.

3.5.5 Ciquième phase

Après avoir parcouru toutes les étapes précédentes, nous arrivons à la phase de création des "communautés finales". À ce stade, nous présentons les nœuds regroupés en communautés identifiées. Cette phase implique la présentation des communautés avec leurs membres, pour donner une vue d'ensemble des groupes cohérents et connectés dans le réseau.

3.6 Algorithmes du système

Pour exécuter cet algorithme, nous devons préparer les données d'entrée suivantes : **Entrées** :

— Un graphe G contenant les données du réseau.

Une fois les données d'entrée préparées, nous pouvons exécuter la fonction que nous avons implémentée. C'est ce que nous devons apporter et ce qui sortira

Entrées :

— Nous nous assurons que la variable globale G est définie avec les données du graphe.

Sortie :

— Si le graphe G est vide, il affichera "Le graphe est vide." et retournera.

— Sinon, il calculera le PageRank pour tous les nœuds dans G.

— Il organisera les nœuds descendants par score PageRank.

— Il effectuera le regroupement des nœuds en communautés.

— Il supprimera les nœuds chevauchants qui appartiennent à plusieurs communautés en utilisant des critères spécifiques.

— Il supprimera les sous-communautés.

— Il affichera les nœuds supprimés qui chevauchent des communautés.

— Il affichera les communautés finales.

Les fonctions :

1. `realnet()` : Chargement d'un graphique à partir d'un fichier edge. Son travail consiste à charger un fichier de bord spécifié par l'utilisateur et à créer un objet graphique à l'aide du module NetworkX.
2. `gen()` : Génération de graphe aléatoire. Cette fonction crée un graphe aléatoire en spécifiant le nombre de nœuds et un paramètre epsilon (mu) qui contrôle la probabilité d'ajouter une arête entre deux nœuds.
3. `generate_graph(n, mu)` : Génération de graphes à l'aide du modèle LFR (Lancichinetti-Fortunato-Radicchi). Cette fonction crée un graphe synthétique à l'aide du modèle de référence LFR spécifiant le nombre de nœuds, les paramètres de distribution des degrés (tau1 et tau2), le paramètre epsilon (mu) qui contrôle la probabilité d'ajout d'arêtes et d'autres paramètres.
4. `app()` : Application principale qui détecte les communautés dans le graphe. Cette fonction utilise l'algorithme de détection de communautés basé sur le PageRank pour identifier les communautés du graphe.
5. `graphcolor(l)` : Coloration des nœuds du graphe selon les communautés détectées. Cette fonction attribue différentes couleurs aux nœuds appartenant à différentes communautés et affiche un graphique avec les couleurs correspondantes.
6. `affiche_graph()` : Affichage graphique. Cette fonction affichera le graphique sans coloration ni manipulation particulière.

7. `Page_rankg()` : Calcul des scores PageRank pour les nœuds de graphique. Cette fonction utilise l'algorithme PageRank pour calculer un score PageRank pour chaque nœud du graphique. Renvoie une liste de nœuds triés par ordre décroissant en fonction de leur score PageRank.

La première phase

Les fonctions :

- **realnet() :** Collecte de données d'un réseau réel
- **gen() :** Génération d'un réseau artificie

— **affiche**$_g raph()$: $Affichagegraphiqueduréseau$

Algorithme 1 : Reseau_real

---

1. Début :
2. Lire 'txt'.txt
3. Retour G
4. Fin.

Algorithme 2 : Reseau_Artificiel

Entree : n,mu

1. Début :
2. Generer_LFR(n,mu)
3. Retour G
4. Fin.

La deuxième phase

Les fonctions :

— **Page_rankg()** cette fonction calcule le PageRank pour chaque page d'un graphe G et renvoie une liste des pages classées par ordre décroissant de leur importance selon le PageRank.

Algorithme 3 : Calculate_PageRank(G ; Graphe) : ListeDeNoeuds

---

Entree :

pageranks : DictionnaireDeReels

sorted_nodes : ListeDeNoeuds

1. Début :

2. *pageranks* := AppelerFonctionPageRank(G)

3. *sorted_nodes* := TrierNoeudsParPageRank(pageranks)

4. Fin.

La troisième phase

Les fonctions :

— **app()** Application principale qui détecte les communautés dans le graphe. Cette fonction utilise l'algorithme de détection de communautés basé sur le PageRank

---

Algorithme 4 : TrouverCommunautes

Entree : sorted_nodes

Début :

1.communautés := []

2.Pour chaque noeud DANS sorted_nodes

3. communauté_trouvée := FAUX

4. Pour chaque communauté DANS communautés

5. Si noeud DANS communauté Alors

6. communauté_trouvée := VRAI

7. SORTIR

8. Fin Si

9. Fin Pour

10. Si NON communauté_trouvée Alors

11. communauté := ENSEMBLE[noeud]

12. Pour chaque voisin DANS G.voisins(noeud) Faire

13. communauté := communauté **UNION** ENSEMBLE[voisin]

14. Fin Pour

15. communautés.ajouter(communauté)

16. Fin Si

17.Fin Pour

18.Pour chaque noeud DANS sorted_nodes Faire

19. communauté_trouvée := FAUX

20. Pour chaque communauté DANS communautés Faire

21. Si noeud DANS communauté Faire

22. communauté_trouvée := VRAI

23. SORTIR

24. Fin Si

25. Fin Pour

26.Fin Pour

27.RETURN communautés

28.Fin.

La quatrième phase

Algorithme 5 : SupprimerNoeudsChevauchants

Entree : overlapping_nodes_set, communities_with_node, max_neighbors, target_community, neighbors, same_neighbors_communities, min_nodes

Début :

1. overlapping_nodes_set := ENSEMBLE(overlapping_nodes_list)
2. POUR CHAQUE node DANS overlapping_nodes_set FAIRE
3. communities_with_node := [c POUR CHAQUE c DANS communities SI node DANS c]
4. SI LONGUEUR(communities_with_node) > 1 ALORS
5. POUR CHAQUE community DANS communities_with_node FAIRE
6. neighbors := $LONGUEUR(community \cap (G.voisins(node)))$
7. SI neighbors > max_neighbors ALORS
8. max_neighbors := neighbors
9. target_community := community
8. FIN SI
9. FIN POUR
10. same_neighbors_communities := [community POUR CHAQUE community DANS
11. communities_with_node SI LONGUEUR(community.INTERSECTION(G.voisins(no = max_neighbors]
12. SI LONGUEUR(same_neighbors_communities) > 1 ALORS
13. POUR CHAQUE community DANS same_neighbors_communities FAIRE
14. SI LONGUEUR(community) < min_nodes ALORS
15. min_nodes := LONGUEUR(community)
16. target_community := community
17. FIN SI
18. FIN POUR
19. FIN SI
20. FIN SI

21. FIN POUR

22. RETURN communautes

23. Fin.

La ciquième phase

Les fonctions :

— graphcolor(l) : Il s'agit d'une représentation générale du graphique des communautés détectées dans un graphe.

Algorithme 6 : Affiche_graphe

---

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

3.7 Exemple illustratif

Pour montrer l'efficacité de notre approche, on utilise l'exemple suivant :

Notre exemple est un graphe avec 14 nœuds et 19 arrêtes, ce graphe est non pondéré et non orienté. Tous les schémas qu'on va utiliser au cours de ces exemples sont de notre application. Notre exemple se déroule en cinq phases.

3.7.1 La première phase

Dans cette phase, on va importer la base de donnée de réseau, et on va le transformer en graphe.

3.7.2 La deuxième phase

Dans cette phase, notre algorithme calcule le Pagerank des noeuds.

Le tableau 3.1 illustre les résultats de cette étape.

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Pagerank*10^3	90	36	60	83	70	93	82	75	75	70	70	70	59	59

TABLE 3.1 – Valeurs des pagrank des noeuds

Ensuite, Puis il ordonne tous les nœuds du réseau par ordre décroissant de Pagerank, dans le cas où il trouve 2 nœuds avec la même valeur de Pagerank, il classe ces nœuds par ordre croissant de leur numéro (choisi la première).

Les résultats de cette étape : ['6', '1', '4', '7', '8', '9', '11', '12', '5', '10', '3', '13', '14', '2']

3.7.3 La troisième phase

> Nous regroupons le nœud 6 avec ses voisins : $[6, 5, 10, 8, 9]$.

> Ensuite, nous travaillons avec le deuxième nœud de la liste (le nœud 1) : $[1, 2, 3, 4]$.

> Nous ignorons le nœud 4 car il existe déjà, puis nous travaillons avec le nœud 7 : $[7, 9, 14, 13]$.

> Nous ignorons les nœuds 8 et 9 car ils existent déjà.

> Enfin, nous travaillons avec le dernier groupe de nœuds : $[11, 5, 12, 10]$.

> Nous nous arrêtons ici.

> Nous avons obtenu quatre communautés, dont trois sont imbriquées et une communauté disjointe. $[[6, 5, 10, 8, 9], [1, 2, 3, 4], [7, 9, 14, 13], [11, 5, 12, 10]]$,

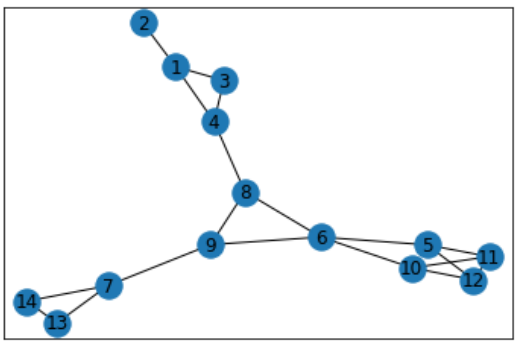

FIGURE 3.2 – Représentation graphique du graphe.

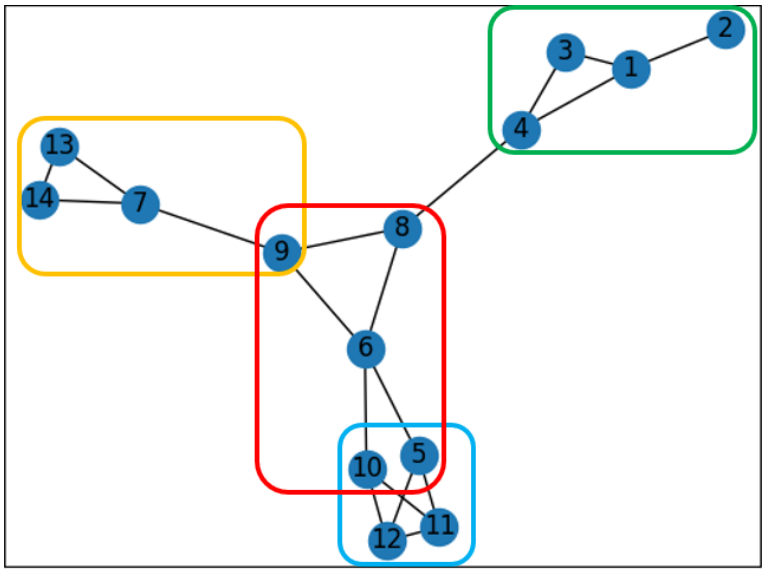

FIGURE 3.3 – Représentation graphique du graphe.

3.7.4 La quatrième phase

Dans cette phase, on va éliminer le chevauchement, les noeuds chevauchentes sont : ['9', '5', '10']. Pour éliminer des chevauchements Nous ajouterons les deux nœuds 5 et 10 à la communauté bleue car ils ont chacun deux voisins dans cette communauté, tandis que la communauté rouge n'a qu'un seul voisin. Quant au nœud 9, il sera ajouté à la communauté rouge car il contient deux voisins, mais le jaune en contient un

3.7.5 La ciquième phase

Dans cette phase, 3.4 montre un affichage graphique des communautés détectées.

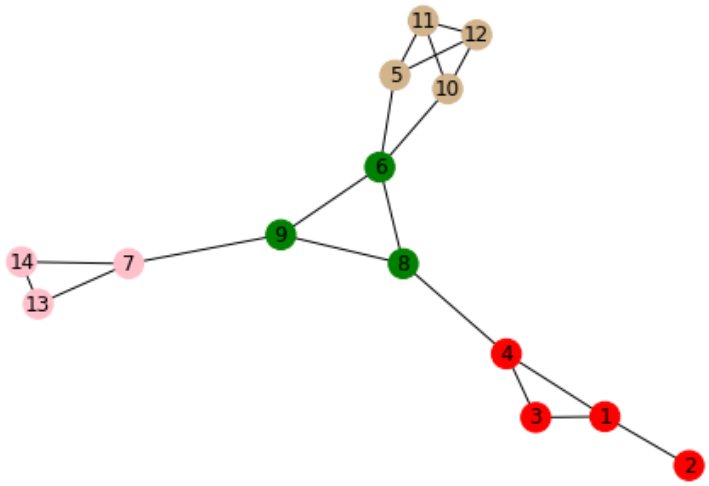

FIGURE 3.4 – Représentation graphique des communautés détectées.

3.8 Conclusion

En conclusion, le chapitre offre une vue d'ensemble du contexte, de l'objectif et des étapes des travaux réalisés pour la détection de communautés à l'aide de l'algorithme PageRank. Il fournit une base solide pour la compréhension de l'approche et prépare le terrain pour le chapitre suivant qui approfondiront les résultats et l'analyse des expérimentations.

Chapitre 4

Implémentation

4.1 Introduction

Dans ce chapitre, nous présentons les outils et les langages utilisés pour implémenter la méthode proposée dans le chapitre précédent. Par la suite, nous montrons les résultats obtenus après la comparaison de notre méthode avec quelques algorithmes de pointe.

4.2 Environnement de travail

4.2.1 Environnement matériel

Toutes nos installations et nos tests seront réalisés sur un un micro-ordinateur qui possède les caractéristiques suivantes :

> Processeur : Intel(R) Core(TM) i5-5300U CPU @ 2.30GHz ;

> Mémoire Installée (RAM) : 8,00 Go ;

> Type de système : système d'exploitation windows10 64 bits.

Notre application a été développée sous un système d'exploitation Windows 10 de 64 bits où on a utilisé le langage de programmation python 3.10 sous Anaconda.

4.2.2 Environnement logiciel

Nous avons utilisé le langage de programmation Python la version . Python est un langage de programmation interprété, interactif, orienté objet et de haut niveau

* **Python est interprété :** Python est traité au moment de l'exécution par l'interpréteur. Nous n'avons pas besoin de compiler notre programme avant de l'exécuter. Ceci est similaire à PERL et PHP.
* **Python est interactif :** Nous pouvons réellement nous installer à une invite Python et interagir directement avec l'interpréteur pour écrire nos programmes.
* **Python est orienté objet :** Python prend en charge le style ou la technique de programmation orienté objet qui encapsule le code dans des objets.

L'installation de python est gratuite et facile, il suffit de le télécharger depuis le site : `https://www.python.org/downloads/`

4.2.3 Plateforme et IDE

Nous avons utilisé la Plateforme Anaconda car c'est un distributeur libre et open source du langage de programmation Python appliqué au développement d'applications dédiées à la science de données et à l'apprentissage automatique (traitement de données à grande échelle, analyse prédictive, calcul scientifique)

Pour téléchargez et installez la 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 á partir du site Web d'Anaconda : `https://www.anaconda.com/products/distribution`

Nous avons utilisé l'environnement de développement (IDE) Spyder. C'est un environnement de développement pour Python libre et multiplateforme (Windows, Mac OS, GNU/Linux) qui contient nombreuses bibliothèques d'usage scientifique : Matplotlib, NumPy, SciPy et IPython.

4.2.4 Bibliothèques utilisées

Une bibliothèque est un ensemble de fonctions prédéfinies. Celles-ci sont regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire. Python est un langage de programmation très riche avec ses bibliothèques.

On a utilisé plusieurs paquets (bibliothèques) dans ce travail qui sont :

> `networkx` : une bibliothèque Python pour la manipulation, la création et l'étude de la structure, des dynamiques et des fonctions des réseaux complexes.

> `numpy` : une bibliothèque Python qui prend en charge les tableaux multidimensionnels et les fonctions mathématiques de haut niveau pour les manipuler.

> `matplotlib` : une bibliothèque de traçage en 2D en Python qui produit des figures de qualité de publication dans une variété de formats imprimés et interactifs.

> `sklearn.metrics.normalized_mutual_info_score` : une fonction de la bibliothèque scikit-learn qui calcule la mesure d'information mutuelle normalisée (NMI) entre deux ensembles de balises.

4.2.5 Présentation du système

Dans cette section, nous présenterons quelques modules pour notre application. L'image nous montre l'interface principale de notre application, où se trouvent de nombreux boutons que nous avons créés pour importer ou écrire la base de données que nous allons traiter, et des boutons pour les algorithmes que l'utilisateur peut implémenter.

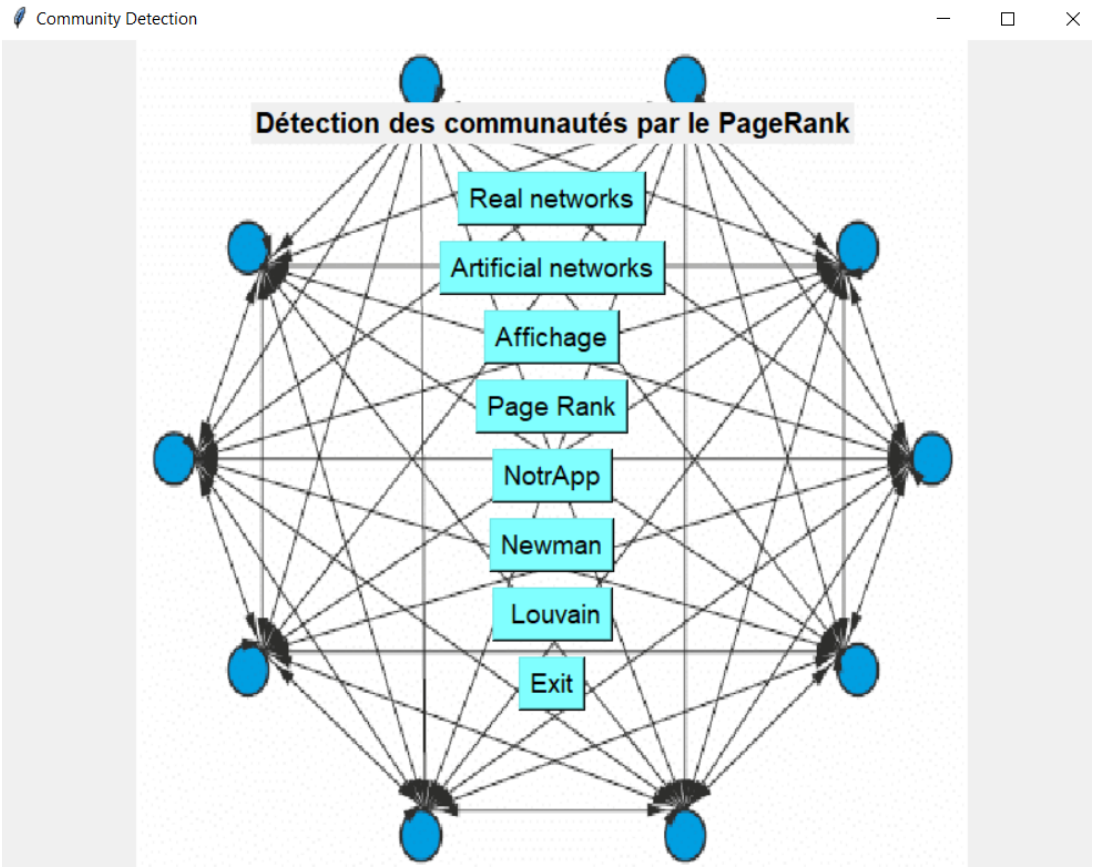

FIGURE 4.1 – Interface générale de l'application

Méthode	Nb_communautés	Modularité	Conductance moyenne
Notre Approche	6	0.5100	0.2113
Newman	6	0.5379	0.1892
Louvain	5	0.5402	0.1874

TABLE 4.1 – Mésure d'évaluation de dauphins

4.3 Résultats expérimentaux et analyse

4.3.1 Expériences sur les réseaux du monde réel

Pour vérifier l'efficacité de notre algorithme, nous avons utilisé cinq réseaux du monde réel (dauphins de Lusseau, karaté, Livres politiques, Football américain et Facebook).

Les dauphins de Lusseau

Lusseau et al. (2003) ont mené une étude scientifique appelée le réseau des dauphins de Lusseau qui an examiné les interactions sociales entre les dauphins d'une population spécifique. Les relations entre les membres de cette population de dauphins sont représentées par ce réseau social, qui permet d'étudier les schémas de connectivité sociale au sein du groupe.[15]

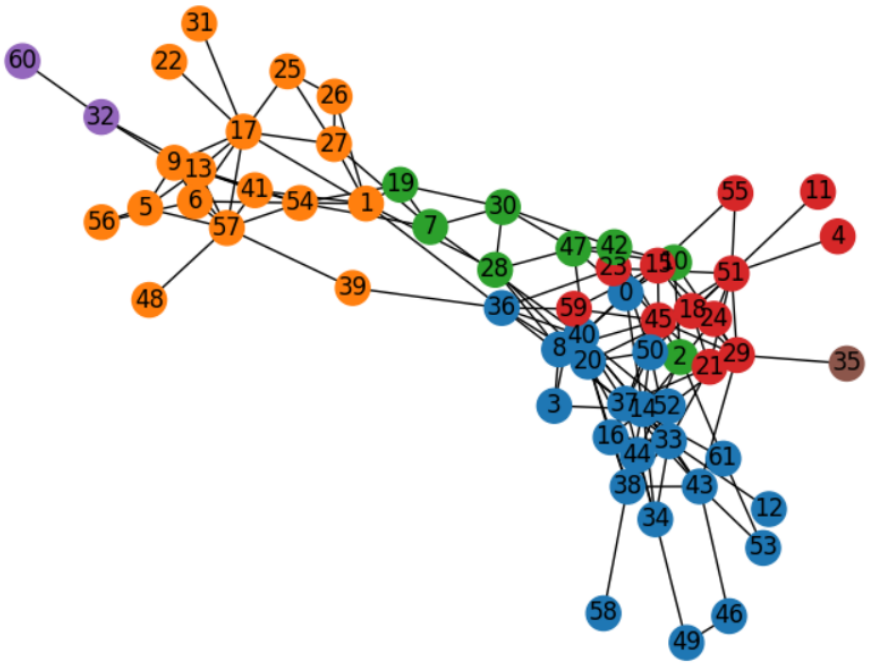

FIGURE 4.2 – dauphins de Lusseau

Karate

Le domaine de la détection de communautés étudie fréquemment le réseau social du club de karaté de Zachary. Son nom vient de l'étude de 1977 de Wayne W. Zachary qui an examiné les interactions sociales entre les membres d'un club de karaté universitaire.Il est composé de 34 noeuds et de 78 arêtes [27]. En appliquant l'algorithme que nous avons proposé à ce réseau, nous avons obtenu deux communautés.

Méthode	Nb_communautés	Modularité	Conductance moyenne
Notre Approche	2	0.4213	0.0686
Newman	4	0.4418	0.2131
Louvain	4	0.4435	0.2130

TABLE 4.2 – Mésure d'évaluation de karaty

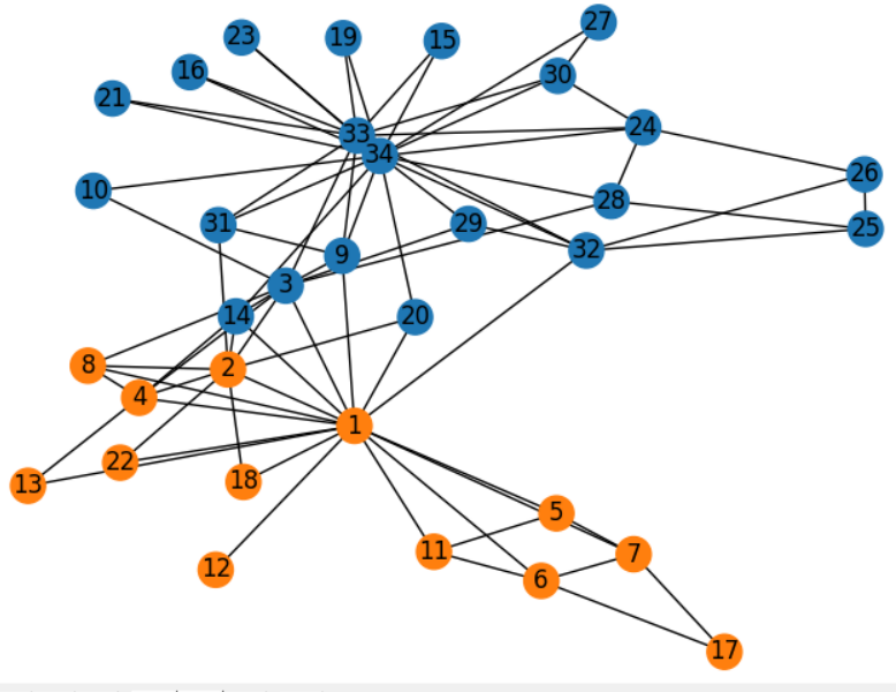

FIGURE 4.3 – Karate

Les livres politiques

Le réseau des livres politiques fait référence à un réseau construit à partir des liens entre les livres politiques, tels que les ouvrages traitant de sujets politiques, sociaux ou économiques. Dans ce réseau, chaque livre est représenté par un nœud, et les liens entre les livres sont établis en fonction de différentes relations, telles que les références bibliographiques, les citations ou les co-occurrences dans des bibliographies.

Méthode	Nb_communautés	Modularité	Conductance moyenne
Notre Approche	4	0.5282	0.1260
Newman	5	0.5405	0.1807
Louvain	5	0.5405	0.1806

TABLE 4.3 – Mésure d'évaluation de livres politiques

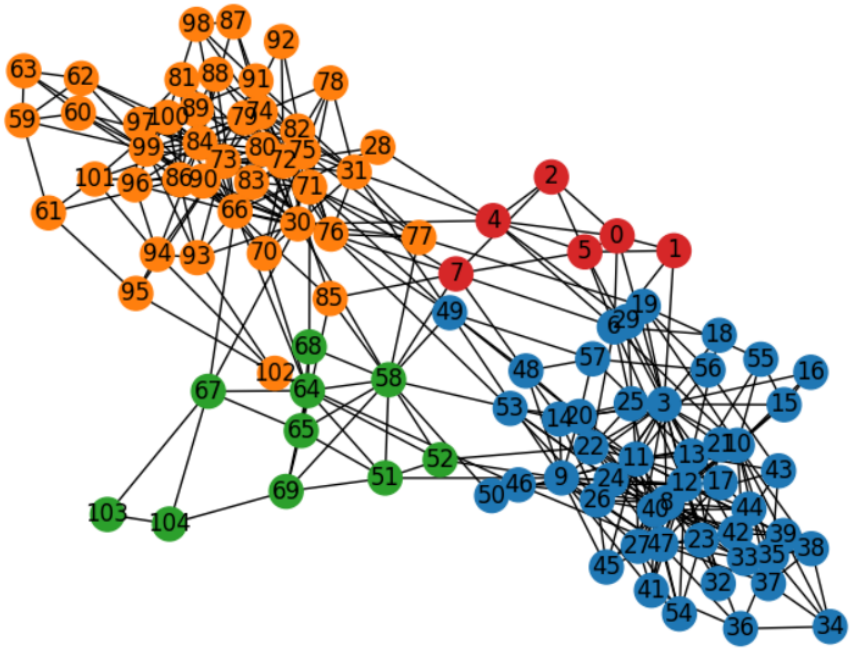

FIGURE 4.4 – Les livres politiques

Football américain

Le terme "réseau du football américain" fait référence à un réseau social composé de connexions entre les joueurs, les entraîneurs, les équipes et autres parties prenantes du football américain. Dans ce réseau, chaque acteur est représenté par un noeud, et les liens entre les nœuds peuvent représenter des interactions comme les relations de travail, les collaborations, les amitiés ou les affinités professionnelles.

Méthode	Nb_communautés	Modularité	Conductance moyenne
Notre Approche	6	0.5585	0.1700
Newman	10	0.6074	0.1721
Louvain	10	0.6142	0.1641

TABLE 4.4 – Valeurs des mésures d'évaluations des réseaux.

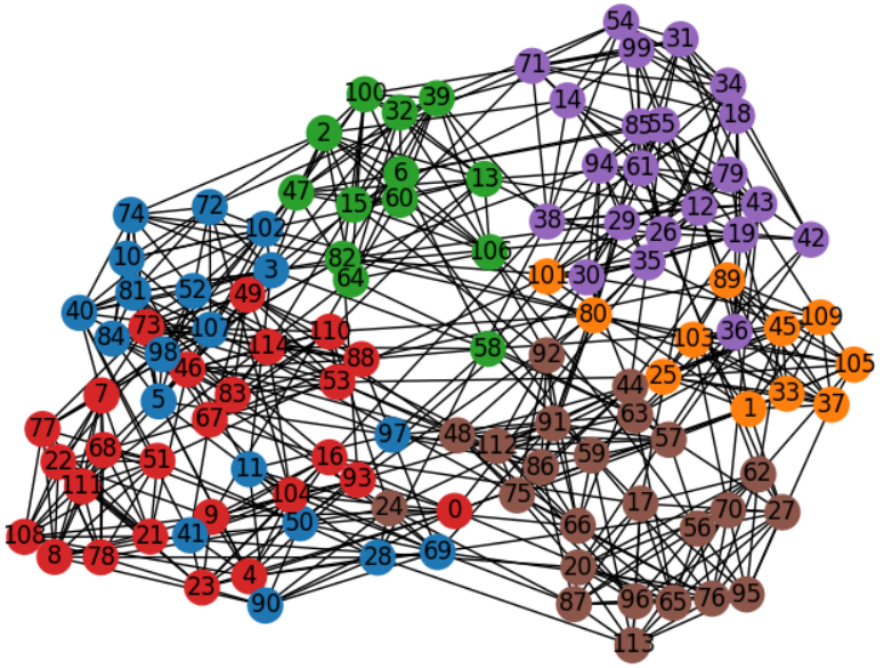

FIGURE 4.5 – Football américain

Facebook

Le réseau Facebook est un réseau social massif qui connecte des millions de personnes à travers le monde. Chaque utilisateur est représenté par un nœud dans le réseau, et les liens entre les nœuds sont établis lorsque deux utilisateurs deviennent amis sur la plateforme. Ces liens d'amitié créent un réseau complexe de relations et d'interactions entre les utilisateurs.

Méthode	Nb_communautés	Modularité	Conductance moyenne
Notre Approche	6	0.6048	0.0416
Newman	16	0.8250	0.0331
Louvain	14	0.8247	0,0267

TABLE 4.5 – Mésure d'évaluation de facbook

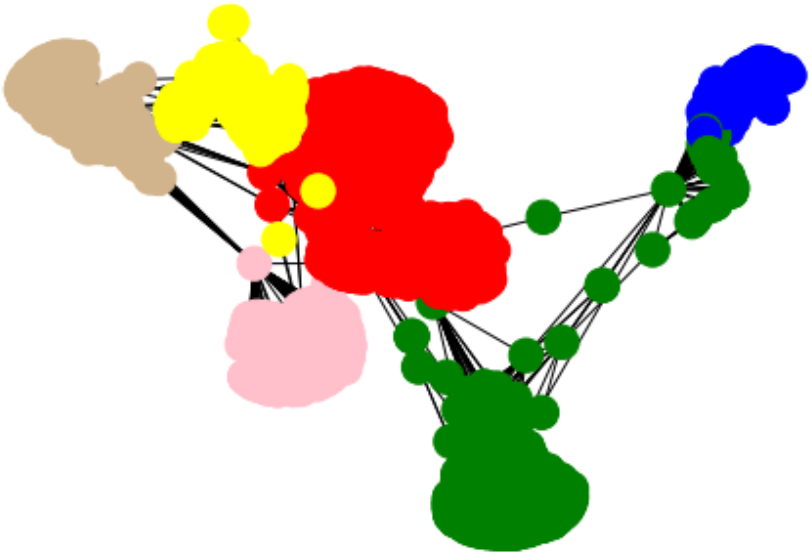

FIGURE 4.6 – Facebook

4.3.2 Comparaison des performances des algorithmes

	Louvain	Newman	LPA	F.Greedy	Notre Approche
Karate	0.4435	0.4418	0.132	0.38	0.4213
Dolphins	0.5402	0.5379	0.47	0.49	0.5100
Football	0.6142	0.6074	0.52	0.57	0.5585
politiques_books	0.5405	0.5405	0.48	0.50	0.5282

TABLE 4.6 – Valeurs de modularités des differents réseaux.

Ce tableau montre les résultats de quatre ensembles de données différents : Karate, Dolphins, Football et politiques_books. Une colonne représente chaque méthode, et chaque cellule du tableau contient les valeurs qui indiquent les scores obtenus par chaque méthode pour chaque ensemble de données.

Les outils d'analyse de réseaux utilisés sont Louvain, Newman, LPA, F.Greedy et Notre Approche.

En analysant les résultats, on peut constater que la méthode Louvain obtient le score le plus élevé pour l'ensemble de données Karate avec 0,4435, suivie de près par la méthode Newman avec 0,4418. Notre méthode obtient un score de 0.4213, ce qui la place au milieu.

La méthode Louvain et la méthode Newman obtiennent des scores très similaires pour l'ensemble de données des Dolphins, avec 0.5402 et 0.5379. Notre méthode reçoit un score légèrement plus élevé de 0.5100.

La méthode Louvain obtient le score le plus élevé pour l'ensemble de données Football avec 0,6142, suivie de près par la méthode Newman avec 0,6074. Notre méthode reçoit un score inférieur de 0.5585.

Enfin, la méthode Louvain et la méthode Newman obtiennent des scores identiques avec 0.5405 pour l'ensemble de données politiques_books. Le score de notre méthode est légèrement inférieur à 0.5282.

En résumé, les résultats de ce tableau montrent que, en fonction de l'ensemble de données, les différentes méthodes d'analyse de réseaux fonctionnent différemment. Dans la plupart des cas, la méthode Louvain et la méthode Newman semblent être les plus efficaces, tandis que notre approche se situe généralement dans la moyenne des résultats.

4.3.3 Réseau artificial

Dans le réseau artificiel, le nombre de nœuds est demandé à l'utilisateur, ainsi que la valeur epsilon, qui contrôle la densité des bords du graphe généré. Une valeur epsilon plus élevée augmentera la probabilité d'ajouter une arête, ce qui entraînera une augmentation de la densité du graphe. Inversement, une faible valeur d'epsilon se traduira par une probabilité plus faible d'ajouter une arête, ce qui se traduira par un graphique clairsemé. Après cela, le graphique créé sera affiché avec ses informations statistiques.

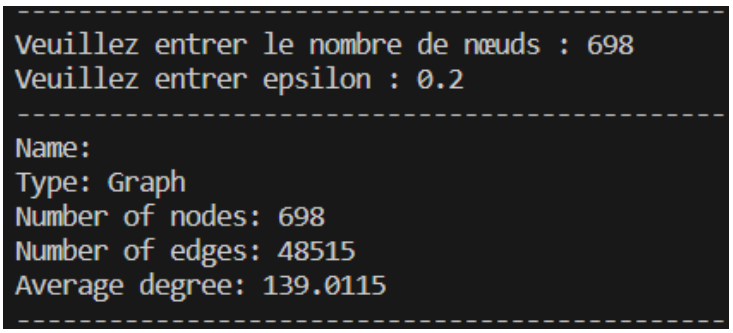

FIGURE 4.7 – Réseau artificiel

Méthode	Notre_Approche	Newman	Louvain
communautés	24	9	8
Modularité	0.025	0.0650	0.0637
Conductance_moyenne	0.7757	0.7284	0.688

TABLE 4.7 – Mésure d'évaluation d'un réseau réél

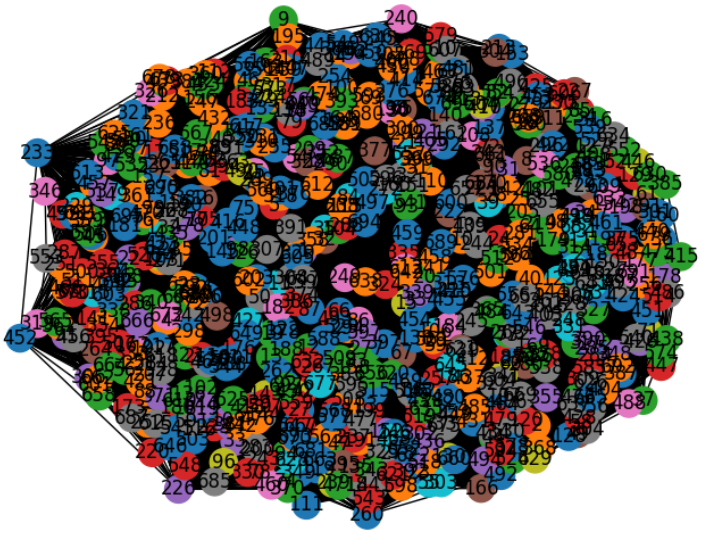

FIGURE 4.8 – Réseau artificiel

4.4 Complexité

Le rapport de complexité général dépend principalement du nombre de nœuds (n), du nombre d'arêtes (E) et du nombre de communautés (C) dans le graphe. Voici un aperçu de la complexité de chaque étape :

Calcul du PageRank pour tous les nœuds : La complexité dépend de l'algorithme PageRank utilisé, mais généralement, elle est de l'ordre de O(n^2) ou O(n^3).

Recherche et création de communautés : Cette étape implique des boucles imbriquées pour parcourir les nœuds et les communautés. La complexité est généralement de l'ordre de O(n^2 C).

Suppression des nœuds chevauchants : Cette étape implique également des boucles imbriquées pour parcourir les nœuds et les communautés afin de supprimer les nœuds chevauchants. La complexité est généralement de l'ordre de O(n^2 C).

Suppression des sous-communautés : Cette étape implique la vérification de la

connectivité des sous-graphes formés par les communautés. La complexité dépend du nombre de sous-communautés et de la taille de chaque sous-communauté.

En résumé, la complexité globale de l'algorithme peut être approximativement estimée à O(n^2 C), où n est le nombre de nœuds du graphe et C est le nombre de communautés.

Le tableau suivant nous montre une comparaison de la complexité de notre algorithme avec d'autres algorithmes de détection de communauté.[6]

Méthode	Complexité
Notre méthode	$O(n^2c)$
Walktrap	$O(n^4)$
Clauset et Newman	$O(n^2 d \log n)$
Girvan et Newman	$O(n^5)$
Fortunato	$O(n^7)$
Louvain	$O(mn^3)$

TABLE 4.8 – Comparaison des complexités

4.5 Les avantages de notre approche

> **Simplicité d'utilisation :** notre approche est simple à utiliser, ce qui la rend accessible aux utilisateurs novices ou ceux qui souhaitent une solution rapide et facile sans avoir à régler de nombreux paramètres.

> **Moins de chevauchement des communautés :** Les communautés détectées présentent moins de chevauchement par rapport aux autres approches. Cela signifie que les nœuds appartiennent principalement à une seule communauté, ce qui peut être utile pour des analyses plus ciblées sur des groupes spécifiques.

> **Interprétation claire des résultats :** Étant donné que les communautés détectées par sont plus disjointes, il est généralement plus facile de les interpréter et de comprendre les relations entre les nœuds. Cela peut faciliter l'analyse des réseaux complexes et la compréhension des structures de communautés.

> **Performances computationnelles :** En raison de sa simplicité, peut être plus rapide en termes de temps de calcul par rapport à d'autres méthodes plus com-

plexes. Cela peut être avantageux lors de l'analyse de grands réseaux où les performances computationnelles sont essentielles.

> **Complexité :** Notre approche offre une complexité réduite par rapport à d'autres approches.

4.6 Conclusion

Dans ce chapitre, nous avons présenté l'environnement de travail utilisé pour mettre en œuvre notre méthode de détection de communauté. Nous avons décrit l'environnement matériel ainsi que les logiciels et bibliothèques utilisés. Nous avons également présenté la plate-forme et l'IDE utilisés pour le développement.

Nous avons ensuite révélé les résultats expérimentaux obtenus en appliquant notre méthode à des réseaux réels et à un réseau artificiel. Nous avons analysé ces résultats pour évaluer les performances de notre algorithme de détection de communauté.

Nous avons également discuté de la complexité de notre méthode et mis en évidence les aspects temporels qui peuvent affecter les performances de l'algorithme, en particulier pour les grands graphes.

En conclusion, notre méthode présente des avantages significatifs pour la détection de communauté dans les graphes, mais il est important de considérer les inconvénients mentionnés et d'évaluer leur impact en fonction du contexte spécifique de l'application.

Conclusion Générale

En conclusion, notre projet sur la détection des communautés de réseaux à l'aide de l'algorithme PageRank est une exploration approfondie d'une méthode prometteuse pour analyser et comprendre la structure communautaire des réseaux. À travers nos quatre chapitres interconnectés, nous avons couvert divers aspects de la détection de communauté en mettant l'accent sur l'application du PageRank.

Dans notre premier chapitre, nous avons introduit les concepts de base de la détection de communauté et exploré les différentes approches et algorithmes utilisés sur le terrain. Nous avons également discuté des méthodes d'évaluation de la détection communautaire qui nous permettent de quantifier la qualité de nos résultats.

Le deuxième chapitre s'est concentré sur l'algorithme PageRank a été largement étudié et amélioré au fil des années, avec différentes approches et techniques proposées. Il présente des propriétés intéressantes, telles que la résistance aux manipulations et la capacité à détecter des communautés dans un réseau. Cependant, il existe également d'autres algorithmes qui ont été développés et appliqués avant le PageRank, et qui méritent d'être comparés. Dans l'ensemble, le PageRank reste un outil puissant pour l'analyse des réseaux et la recherche d'informations sur le web. Son utilisation continue d'évoluer et de se développer, offrant de nouvelles perspectives et opportunités dans le domaine de la recherche d'informations en ligne.

Dans le troisième chapitre, nous avons présenté la conception de notre système de détection communautaire, décrit la problématique, les objectifs et les phases de notre travail. Nous avons également fourni des exemples concrets et illustratifs pour faciliter la compréhension des concepts présentés.

Enfin, le dernier chapitre a été consacré à l'environnement de travail de notre projet, il décrit les outils, bibliothèques et résultats obtenus grâce à nos expérimentations et analyses sur des réseaux réels et artificiels.

Dans l'ensemble, notre projet offre une perspective approfondie sur l'utilisation du PageRank pour la détection de communauté et fournit à la fois une base théorique solide et des applications pratiques. Nous espérons que ce travail contribuera à la compréhension et au développement de la détection communautaire dans les réseaux et ouvrira de nouvelles perspectives de recherche et d'applications dans divers domaines.

Bibliographie

[1] K. Ahn and Y. Choi. A comprehensive study on k-means clustering algorithm. *Journal of Intelligent Information Systems*, 43(3) :409–428, 2014.

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics : Theory and Experiment*, 2008(10) :P10008, 2008.

[3] Stefano Boccaletti, Vito Latora, Yamir Moreno, Marián Chavez, and Doo-Hee Hwang. Complex networks : Structure and dynamics. *Physics Reports*, 424(4-5) :175–308, 2006.

[4] Paolo Boldi and Sebastiano Vigna. The web as a graph : Measurements, models, and methods. pages 545–554, 2004.

[5] Jierui Chen, Zhi-Dan Zhao, and Liangliang Cai. A novel method for community detection in complex networks using pagerank algorithm. *Physica A : Statistical Mechanics and its Applications*, 388(14) :2986–2992, 2009.

[6] ACHRAF ELAGGOUNE. Détection des communautés dans les réseaux sociaux. 2020.

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

[8] Santo Fortunato and Darko Hric. Community detection in networks : A user guide. 2016.

[9] J. Garcia and E. Martinez. Evaluating community detection stability in perturbed networks. *Physical Review E*, 90(4) :042805, 2014.

[10] Lixian Guo, Yuanshun Wang, Yu Sun, and Yan Wang. Simulated annealing algorithm for community detection based on modularity optimization. pages 480–486, 2019.

[11] Mustafa Hajij, Eyad Said, and Robert Todd. Pagerank and the k-means clustering algorithm. *arXiv preprint arXiv :2005.04774*, 2020.

[12] Xiao-Yong He, Sheng-Jun Deng, Ding-Quan Wang, and Jin-Hua Hu. Detecting community structures via heat diffusion processes. *Journal of Statistical Mechanics : Theory and Experiment*, 2011(08) :P08007, 2011.

[13] Y. Huang, X. Liu, and C. Zhang. Detecting structural motifs in complex networks. *Physica A : Statistical Mechanics and its Applications*, 460 :279–289, 2016.

[14] X. Jin, J. Han, and Y. Cai. Clustering large attributed graphs : A spectral embedding and refinement approach. *IEEE Transactions on Knowledge and Data Engineering*, 26(8) :1873–1888, 2014.

[15] David Lusseau and et al. The emergence of unshared consensus decisions in bottlenose dolphins. *Behavioral Ecology and Sociobiology*, 54(4) :396–405, 2003.

[16] Marina Meilă. Comparing clusterings by the variation of information. pages 173–187, 2007.

[17] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23) :8577–8582, 2006.

[18] M. E. J. Newman. Networks : An introduction. 2010.

[19] Mark EJ Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 8(1) :25–31, 2012.

[20] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking : Bringing order to the web. *Stanford Digital Library Project*, 1999.

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

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

[23] J. Smith and A. Johnson. Stability in community detection using bootstrap method. *Journal of Network Science*, 10(2) :123–137, 2019.

[24] Q. Wang, J. Zhang, J. Wu, and Z. Liu. Spectral clustering versus k-means : A comparative study. *Pattern Recognition Letters*, 72 :39–47, 2016.

[25] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684) :440–442, 1998.

[26] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 33(3) :681–705, 2012.

[27] Wayne W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4) :452–473, 1977.

[28] Wang Zhi-Xiao, Li Ze-chao, Ding Xiao-fang, and Tang Jin-hui. Overlapping community detection based on node location analysis. *Knowledge-Based Systems*, 105 :225–235, 2016.