N° Ordre : .....

UNIVERSITÉ MOHAMED BOUDIAF - M'SILA
FACULTÉ DES MATHEMATIQUES ET DE L'INFOMATIQUE
Département d'informatique

THÈSE

Présentée en vue de l'obtention du diplôme de

DOCTEUR EN SCIENCES

Filière : Informatique
Spécialité : Informatique

Par :
SAIFI ABDELHAMID

Thème

Fouille d'interactions dans les réseaux complexes

Soutenue le : 16 / 01 / 2024

Devant le jury composé de :

Pr. BOURAHLA Mustapha	**Professeur**	**Université de M'Sila**	**Président**
Pr. NOUIOUA Farid	**Professeur**	**Université de Bordj Bou Arréridj**	**Rapporteur**
Pr. AKROUF Samir	**Professeur**	**Univ. de M'sila**	**Co-Rapporteur**
Pr. BOUAMAMA Salim	**Professeur**	**Université de Sétif 1**	**Examinateur**
Dr. MOUHOUB Nasser Eddine	**MCA**	**Université de M'Sila**	**Examinateur**
Dr. ATTIA Abdelouahab	**MCA**	**Université de Bordj Bou Arréridj**	**Examinateur**

Année universitaire : 2022-2023

Remerciements

Je voudrais profiter de cet espace pour exprimer toutes mes gratitudes et mes reconnaissances à mon encadrant Pr. NOUIOUA Farid, et mon co-encadrant Pr. AKHROUF Samir qui n'ont pas cessé de m'encourager et de me guider vers la bonne voie tout au long de la réalisation de ce travail.

Je tiens à exprimer mes sincères remerciements aux membres du jury d'avoir accepté de juger ce travail.

Je salue l'ensemble des professeurs et étudiants du département informatique.

Je remercie, enfin, toute personne ayant aidé de près ou de loin dans la réalisation de ce travail.

SAIFI Abdelhamid

Dédicaces

À la mémoire
De mes chers père et mère.

À mon épouse,
A mes enfants Aya, Ahmed, Brahim et Youcef

Et à mes sœurs et frères.

SAIFI Abdelhamid

Table des matières

Chapitre -II- Prédiction de liens dans les réseaux complexe : Etat de l'art

Chapitre -III- Une nouvelle approche rapide pour la prédiction de liens

Chapitre –IV- Résultats expérimentaux

Liste des figures

Liste des tableaux

Liste d'abréviations

AA	Adamic Adar index
ADP	Adaptive degree penalization
ADV	Activity-Driven Networks
BUP	Blogs of Unusual Size and Popularity
CAR	Collective Adaptive Random walks
CCLP	Clustering Coefficient for Link Prediction
CDM	Collaboration and Diversity in Macroscope
CEG	C. elegans connectome electrical and gap-junctional network
CGS	Collaboration Graphs with Self-citations
CN	Common Neighbors
CN+PA	Common neighbor plus preferential attachment
CN2D	Common Neighbor to Degree
EML	Email networks of the Enron Corporation
FBK	Facebook Like Network
FSW	Functional Similarity Weight
GRQ	Generalized Reciprocity Quotient
HDI	Hub depressed index
HMT	Hollywood Movie-Tie-In
HPI	Hub promoted index
HTC	Highly Cited Researchers and Papers in Top 20 Science and Social Science Journals
IA	The Individual Attraction
INF	Infectious disease spread through contact networks in Flanders
JI	Jaccard index
KNH	Korean National Assembly Hansard
LDG	Leskovec, Datasets, and Galileo
LHNI	Leicht-Holme-Newman index
LIT	Local InTeracting Score
LMPD	local major path degree
LNB	Local Naïve Bayes
LP	Link Prediction
MI	Mutual Information
NLC	Node and Link Clustering
NSC	Networks of Scientific Collaboration
PA	Preferential Attachment Index
PGP	Pretty Good Privacy
RA	Resource allocation index
RA-CNI	Resource Allocation Based on Common Neighbor Interactions
RP	Relative-path
Sam	Samad Similarity
SI	Salton index
SO	Sorensen index
TRA	Triangle Structure Index
UAL	United Airlines Network
ZWL	Zavrsnik, Wirsing, and Lautenbach networks

Résumé

Ces dernières années, les réseaux complexes tels que Facebook, Twitter, etc. ont connu une évolution considérable. Ces bases de données sont énormes et leur utilisation est souvent très gourmande en temps. Dans ce travail, nous présentons un calcul optimal en fouille de graphes pour la prédiction de liens, afin de réduire le temps d'exécution. Dans cette perspective, nous proposons une nouvelle approche qui exploite les composantes connexes d'un réseau au lieu d'opérer sur l'intégralité du réseau. Nous montrons que grâce à cette décomposition, les résultats de tous les algorithmes de prédiction de liens, utilisant les mesures de similarité basées sur le voisinage et le chemin, sont obtenus avec beaucoup moins de calculs et donc dans un délai beaucoup plus court. Nous montrons que ce gain dépend de la répartition des nœuds dans les composantes et peut être capté par les mesures de Gini et de variance. Nous proposons ensuite une architecture parallèle du processus de prédiction de lien basée sur la décomposition en composantes connexes. Pour valider cette architecture, nous avons mené une étude expérimentale sur une large gamme d'ensembles de données. Les résultats obtenus confirment clairement l'efficacité de l'exploitation de la décomposition du réseau en composantes connexes dans la prédiction de lien.

Mots Clés : Prédiction de lien, Réseau social, Traitement parallèle, fouille de graphes, · Informations locales, Fouille d'interactions, Réseaux complexes, composantes connexes, théorie des graphes.

Abstract

Social networks such as Facebook, Twitter, etc. have dramatically increased in recent years. These databases are huge and their use is time consuming. In this work, we present an optimal calculation in graph mining for link prediction to reduce the runtime. For that purpose, we propose a novel approach that operates on the connected components of a network instead of the whole network. We show that thanks to this decomposition, the results of all link prediction algorithms using local and path-based similarity measures scan be achieved with much less amount of computations and hence within much shorter runtime. We show that this gain depends on the distribution of nodes in components and may be captured by the Gini and the variance measures. Then, we propose a parallel architecture of the link prediction process based on the connected components decomposition. To validate this architecture, we have carried out an experimental study on a wide range of well-known datasets. The obtained results clearly confirm the efficiency of exploiting the decomposition of the network into connected components in link prediction.

Keywords: Link prediction, Social network, Parallel computing, Graph mining, Local information, Interaction mining, Complex networks, connected components, graph theory.

.

ملخص

في السنوات الأخيرة ، ظهرت شبكات معقدة مثل Facebook و Twitter وما إلى ذلك وقد تطورت بشكل كبير .قواعد بياناتها ضخمة وتستغرق وقتًا طويلاً في الاستخدام. في هذا العمل ، نقدم حسابًا مثاليًا في التنقيب في الشبكات للتنبؤ بالروابط ، من أجل تقليل وقت التنفيذ . لتحقيق هذه الغاية ، نقترح نهجًا جديدًا يعتمد على المكونات المتصلة لشبكة بدلاً من الشبكة بأكملها .نوضح أنه بفضل هذا التحليل ، يتم تحقيق نتائج جميع خوارزميات التنبؤ بالارتباط باستخدام مقياس التشابه المستند إلى الجوار أوالمسار بحساب أقل بكثير وبالتالي في وقت أقصر بكثير. نوضح أن هذا الكسب يعتمد على توزيع العقد في المكونات ويمكن التقاطه بواسطة معاملات Gini ومقاييس التباين. و من ثم نقترح بنية متوازية لعملية التنبؤ بالارتباط بناءً على التحليل إلى مكونات متصلة .للتحقق من صحة هذه البنية، أجرينا دراسة تجريبية على مجموعة واسعة من مجموعات البيانات. تؤكد النتائج التي تم الحصول عليها بوضوح على كفاءة استغلال تحليل الشبكة إلى مكونات متصلة في توقع الارتباط.

كلمات مفتاحية : توقع الارتباط ، الشبكة الاجتماعية ، الحوسبة المتوازية ، التنقيب في الشبكات ، المعلومات المحلية ، التنقيب التفاعلي ، الشبكات المعقدة ، المكونات المتصلة ، نظرية الرسم البياني.

Introduction Générale

Les réseaux complexes sont des phonèmes qui nous permettent de modéliser divers systèmes du monde réel. Les réseaux sociaux, de transports, de technologies, d'informations et même de biologie sont considérés comme des réseaux complexes. Ces réseaux ont une topologie de nature extrêmement compliquée, irrégulière, dynamique, auto-organisée et évolutive dans le temps.

Un réseau complexe est un réseau d'interactions entre entités dont les nœuds représentant des objets interconnectés par des liens. Ces interactions peuvent être très variées, comme des liens d'amitié ou de parenté, des activités professionnelles ou personnelles communes, ou encore de partage de mêmes opinions, de publications, ou des liens de citations, de voisinages et de pages web, etc.

Les réseaux complexes peuvent être représentés sous forme de graphes avec des acteurs comme des nœuds et des arêtes représentant tout type d'interaction. La représentation des réseaux complexes sous forme de graphe a facilité leur analyse. Cette représentation a permi de calculer et d'extraire plusieurs informations et propriétés caractérisant les nœuds, les liens et le réseau complet.

Plusieurs efforts ont été faits pour comprendre l'évolution des réseaux complexes en termes de nœuds, de liens et d'interaction, leurs relations avec les topologies et le fonctionnement, ainsi que les caractéristiques du réseau.

L'exploration de fouille d'interactions dans des réseaux complexes a de plus en plus attiré l'attention de plusieurs chercheurs et elle est devenue le sujet de nombreuses branches de la science.

Le domaine de recherche le plus important de fouille d'interactions dans les réseaux complexes s'intéresse aux liens entre nœuds et consiste à étudier la prédiction de nouveaux liens. La prédiction consiste à détecter les liens qui manquaient dans le passé mais qui apparaîtront potentiellement comme de nouveaux liens dans le futur. La prédiction nous permet aussi de détecter les liens manquants ou qui ont disparu dans un réseau à partir les liens existants.

1. Motivations

Les réseaux complexes tels que Facebook, Twitter, etc. ont considérablement augmenté ces dernières années. Ces bases de données, qui enregistrent les nœuds et les

interactions entre ces nœuds en termes de liens, sont de tailles énormes et leurs utilisations consomme souvent beaucoup de temps et d'espace de mémoire.

Afin d'appliquer et évaluer les performances d'un algorithme de prédiction de liens, nous devons calculer tous les scores pour tous les liens qui manquaient dans le passé, car ces liens manquants seront de nouveaux liens potentiels dans le futur.

Ainsi, si on considère réseau très volumineux on se rend compte que le nombre de liens inexistants qui doivent être évalués peut-être gigantesque. Par exemple, dans twitter-datasets (Reza and Huan, 2009) qui possède une base de données ayant 11.316.811 nœuds et 85.331.846 liens, le nombre de scores calculés pour les liens inexistants est supérieur à 64 trillions !

Il est à noter que le calcul de scores pour les liens inexistants peut prendre un temps considérable qui dépend de la complexité de la méthode utilisée et de l'information nécessaire pour les calculs, et qui peut être mesuré par jours. De plus, l'espace de mémoire nécessaire pour stocker et traiter les informations nécessaires est souvent très grand également à cause du nombre énorme de nœuds et de liens, et des nombreuses calculs et résultats intermédiaires.

Notre objectif dans ce travail est de trouver un moyen optimal en fouille de graphes pour réduire au maximum le nombre d'informations traitées et ainsi la quantité de calcul nécessaire. Le but étant de minimiser le temps d'exécution et l'espace mémoire nécessaires pour effectuer la tâche de prédiction de liens.

2. Contributions

Dans cette perspective, le but de ce travail est d'étudier, présenter et évaluer les différentes approches pour prédire les liens dans les réseaux complexes, que ce soit les approches basées sur le calcul de voisinage ou celles basées sur le calcul de chemins entre les nœuds. Ensuite, sur la base de cette évaluation, nous proposons une nouvelle approche suffisamment souple, rapide et efficace pour réaliser la prédiction de liens.

Notre approche est basée sur la décomposition du graphe qui représente notre réseau complexe en sous-graphes représentant ces composantes connexes.

Cette approche, nous permet d'atteindre notre premier objectif qui est de diminuer le nombre d'opérations de calcul. En effet, au lieu d'effectuer les calculs sur le graphe complet, notre approche suggère d'effectuer les calculs seulement sur les scores des liens intra-composantes, puisque les scores inter-composantes, sont toujours nuls. Ceci nous permet bien évidemment de réduire significativement le temps d'exécution et l'espace mémoire.

Pour valider et évaluer notre première contribution, nous avons utilisé dans notre étude expérimental 22 bases de données. De plus, une étude comparative est accompagnée à cette

expérimentation dans laquelle notre algorithme est comparé avec l'algorithme classique en utilisant les mêmes mesures de similarités.

Notre deuxième objectif est d'exploiter la décomposition de notre graphe en composantes connexes indépendantes pour paralléliser le reste du calcul, afin de gagner un temps additionnel. Dans cette partie, nous proposons quatre algorithmes dérivés de la décomposition du graphe. De plus, nous proposons une architecture parallèle pour la tâche de prédiction de liens dans un réseau complexe.

Il est à noter que cette approche est valable pour toutes les méthodes de similarité basées sur les informations locales et les méthodes de similarité basées sur les informations globales. Grâce à la répartition logique des calculs sur de nombreux processus parallèles, notre approche offre un gain important en temps d'exécution, en particulier pour les réseaux complexes de grande taille nécessitant beaucoup de calculs.

3. Organisation de la thèse

La présente thèse est structurée en quatre chapitres :

Le premier chapitre est une introduction au sujet de la fouille d'interactions dans les réseaux complexes. Nous exposons dans ce chapitre les caractéristiques des réseaux complexes, et nous illustrons leur présentation sous forme de graphes. Nous terminons le chapitre par une vue générale sur la tâche de prédiction de liens.

Le deuxième chapitre constitue un état de l'art sur la prédiction des liens. Nous examinons en détail les approches qui s'appuient sur des informations locales ainsi que celles qui se basent sur des informations globales. Une fois ces approches présentées en détail, nous les mettons en perspective pour mieux comprendre leur mise en œuvre.

Le troisième chapitre est consacré à la présentation du cadre théorique de notre approche et servira à décrire les différentes phases de celle-ci. Nous y présentons tous les éléments qui nous permettent de concevoir notre solution et aboutir à nos objectifs.

Le quatrième chapitre présente les prototypes des tests que nous avons réalisés pour tester notre architecture proposée. Ce chapitre expose aussi une série d'expérimentations que nous avons réalisées, afin de valider notre approche.

Nous concluons ce travail par une vision synthétique sur le travail que nous avons fait. Nous discutons de l'expérience que nous avons acquise ainsi que les perspectives de travaux futurs que cette expérience nous a ouverte.

CHAPITRE –I-

Réseaux complexes : Concepts de base

Chapitre -I -

Réseaux complexes : Concepts de base

I.1. Introduction

Le travail en groupe est devenu un phénomène très important dans la vie actuelle. Il nécessite plus de collaboration, partage de ressources et de connaissances, coopération, communication, et interaction entre les individués. Chaque groupe constitue un environnement de travail sous forme d'un réseau d'interaction.

Le réseau constitué possède une architecture qui n'est pas forcement uniforme. Sa topologie peut être extrêmement compliquée, dynamique avec auto-organisation, distribution irrégulière, et non-triviale, ce qui justifie l'appellation de réseau complexe.

Plusieurs systèmes naturels ou artificiels, tels que l'interaction entre gènes, protéines et autres molécules, le réseau électrique ou les réseaux de transports, sont des réseaux complexes et peuvent être modélisées par un graphe.

Dans ce chapitre, nous présentons les concepts principaux d'un réseau complexe, ainsi que les différents types de réseaux, leurs caractéristiques et leur représentation sous forme d'un graphe. Nous évoquons également les domaines d'application en mettant l'accent plus particulièrement sur la prédiction de liens.

I.2. Définition

Un réseau est un ensemble de nœuds et de liens. Il est considéré comme un ensemble d'éléments en interactions mutuelles. Le nombre d'éléments ou de nœuds déterminent la taille d'un réseau. Dans un contexte de modélisation, la présence d'un lien entre deux nœuds indique une interaction entre les deux éléments correspondants du système.

Un réseau complexe (Estrada, 2012) est une représentation schématique d'un système. Il est constitué de nœuds, qui représentent les entités du système et de paires de nœuds reliés par des liens, qui représentent un type particulier de l'interconnexion entre ces entités.

A titre d'exemple, dans un système de voies ferrées, les entités à connecter sont les gares. Ils sont reliés les uns aux autres par des lignes de chemin de fer. En général, les entités peuvent représenter des stations, des individus, des ordinateurs, des villes, des opérations, et ainsi de suite, et les connexions peuvent représenter des lignes de chemin de fer, des routes, des câbles, des relations humaines, des hyperliens, etc.

I.3. Représentation Graphique

Les graphes sont utilisés pour représenter de nombreuses applications réelles qui se modélisent par des réseaux complexes. Les réseaux peuvent correspondre à des trajets dans une ville, un réseau téléphonique ou un réseau de circuits entre autres.

Les graphes sont également utilisés pour représenter les réseaux sociaux comme ***linkedIn***, ***Facebook***. Par exemple, dans ***Facebook***, chaque personne est représentée par un sommet (ou nœud). Chaque nœud est une structure qui contient des informations telles que l'identifiant de la personne, le nom, le sexe et les paramètres régionaux.

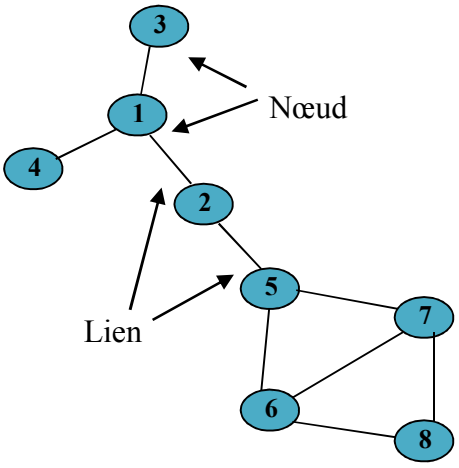
Un graphe est une structure de données composée des deux composants suivants :

- Un ensemble fini de sommets également appelés nœuds.
- Un ensemble fini de paires de la forme (***x, y***) appelés liens. Une paire de la forme (***x, y***) indique qu'il existe un lien du nœud ***x*** au nœud ***y***. Les liens peuvent contenir aussi des valeurs qui représentent des poids ou des coûts.
- Dans le cas d'un graphe orienté, la paire est ordonnée, i.e., (***x, y***) n'est pas la même que *(y,* ***x***).

Formellement, un graphe ***G*** (non orienté) est défini par ***G = (V, E),*** où ***V*** est un ensemble nœuds et ***E*** est un ensemble de liens tel que $\boldsymbol{E \subseteq V \times V}$, Les nœuds $\boldsymbol{v_i}$ et $\boldsymbol{v_j}$ sont liés si $\boldsymbol{e = (v_i, v_j) \in E}$.

Exemple

La Figure I.1. Montre un exemple d'un graphe non orienté avec 8 nœuds et 9 arrêtes.



G=(V,E) où
|V| = 8 /le nombre de nœuds
|E| = 9 /le nombre de liens
V = {1, 2, 3, 4, 5, 6, 7, 8}
E = {e_1(1,2), e_2(1,3),
e_3(1,4), e_3(2,5),
e_5(5,6), e_6(5,7),
e_7(6,7), e_8(6,8),
e_9(7,8)}

Figure I.1 - Graphe simple non orienté avec 8 nœuds et 9 liens.

En théorie des graphes, on trouve plusieurs types de graphes qui diffèrent sur l'interprétation des relations entre les nœuds. Parmi ces types, les plus courants sont :

Graphe non orienté. Graphe ayant des liens sans direction. Ce type de graphes est utilisé pour représenter des liens symétriques.

Graphe orienté. Également appelé digraphe, est un graphe ayant des liens avec direction. A titre d'exemple, cette notion est pertinente dans la situation où une personne ***P1*** connaît une autre personne ***P2***, mais le contraire n'est pas vrai.

Graphe pondéré. Graphe ayant des poids, i.e., des valeurs réelles associées aux liens. Ces poids peuvent représenter une variété de concepts tels que le coût de connexion, la longueur, la capacité, la similarité, la distance, etc., qui dépendent de l'utilisation spécifique de ce graphe.

Graphe régulier (ou uniforme). Graphe dans lequel tous les sommets ont le même nombre de voisins.

Graphe complet. Graphe dans lequel n'importe quelle paire de nœuds est connectée.

Graphe connexe. Graphe dans lequel chaque paire de nœuds distincts dans ce graphe sont connectés, sinon, il est dit non connexe.

Une **composante connexe** est un sous-ensemble $C = \{v_1, \ldots, v_k\} \subseteq V$ tel que, pour tout $v_i, v_j \in C$, il existe un chemin dans G de v_i à v_j.

Un nœud qui n'est pas connecté à aucun autre nœud, est dit **isolé**. Il forme une composante connexe à lui tout seul.

Ainsi, un graphe **connexe** possède une unique composante connexe.

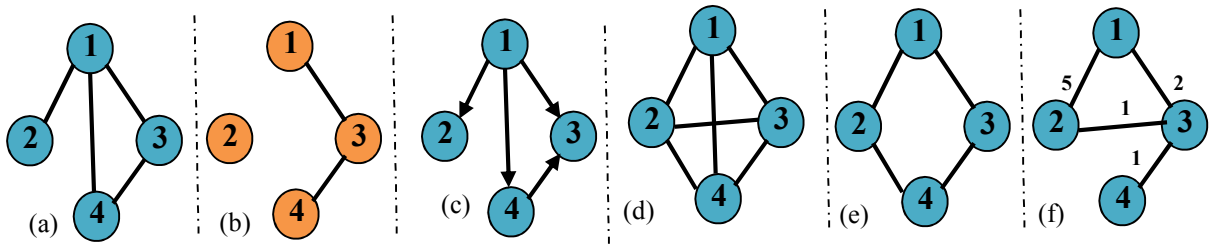


Figure I.2- un réseau complexe présenté par (a) Graphe Connexe (b) Graphe non Connexe (c) Graphe orienté (d) Graphe complet (e) Graphe régulier (f) pondéré

I.4. Représentation algorithmique

Le choix de la structure de représentation du graphe est spécifique à la situation et dépend totalement du type d'opérations à effectuer et de la facilité d'utilisation. Les trois structures suivantes sont les plus couramment utilisées pour stocker un graphe.

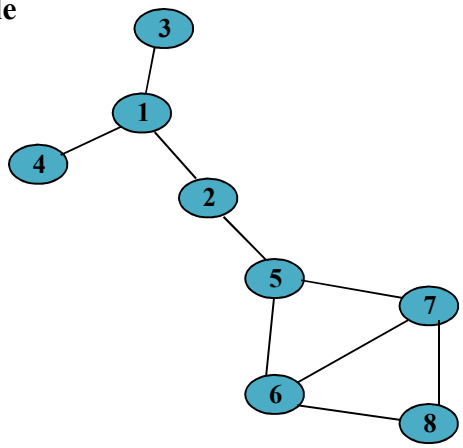
I.4.1. Matrice d'adjacence

On appelle matrice d'adjacence du graphe G ayant n nœuds numérotés de 1 à n, un tableau M à n lignes et n colonnes, où le coefficient $M_{i,j}$ situé à la i ème ligne et j ème colonne est défini par :

$$\forall i \in \{1,..,n\}, \forall j \in \{1,..,n\}, M_{i,j} = \begin{cases} 1 & si\ (i,j) \in E \\ 0 & si\ non \end{cases} \qquad \text{(I-1)}$$

Cette représentation n'est pas toujours efficace en termes de stockage

Exemple



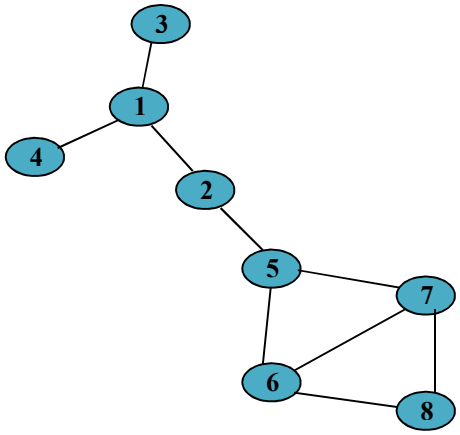
	1	2	3	4	5	6	7	8
1	0	1	1	1	0	0	0	0
2	1	0	0	0	1	0	0	0
3	1	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0
5	0	1	0	0	0	1	1	0
6	0	0	0	0	1	0	1	1
7	0	0	0	0	1	1	0	1
8	0	0	0	0	0	1	1	0

Figure I.3 - Un graphe non orienté et sa représentation en matrice d'adjacence

I.4.2. Liste d'adjacence

On appelle listes d'adjacences du graphe ***G*** un tableau A_{dj} de ***n*** listes, la $i^{ème}$ liste contenant la liste des nœuds adjacents au nœud.

Exemple



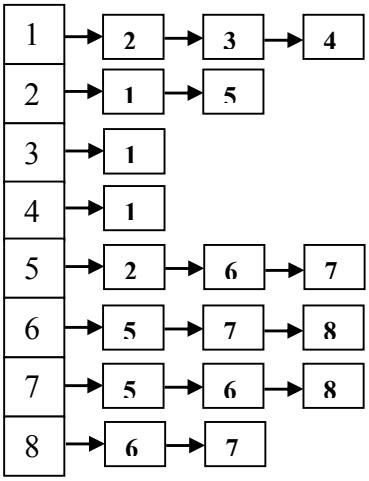
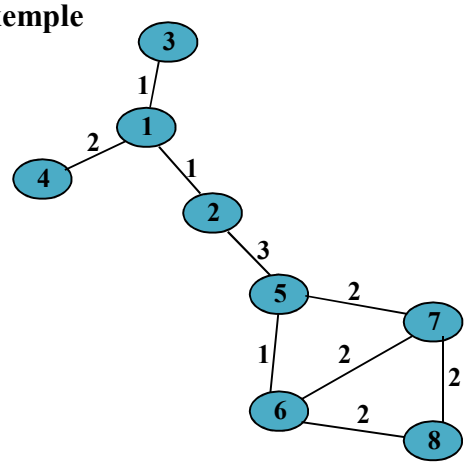


Figure I.4 – Graphe simple non orienté et sa représentation en listes d'adjacences

I.4.3. Matrice de liste des liens

Les listes de liens sont apparues comme une solution au problème de taille dans les matrices d'adjacence. Dans ce type de représentations, il s'agit de lister seulement les liens existants dans le graphe. Cette structure est préférée dans les réseaux de grande taille. Elle est directement stockée dans des fichiers permanents.

Exemple



de	a	valeur
1	2	1
1	3	1
1	4	2
2	5	3
5	6	1
5	7	2
6	7	2
6	8	4
7	8	2

Figure I.5 - Graphe simple non orienté et sa représentation en matrice de liste des liens

I.5. Mesures et propriétés

Les réseaux ont généralement des attributs qui peuvent être mesurés afin d'analyser leurs propriétés et caractéristiques.

I.5.1. Voisinage d'un nœud

On dit que deux nœuds d'un graphe non-orienté sont **voisins** ou **adjacents** s'ils sont reliés par un lien. Le **voisinage** d'un nœud x, noté $\Gamma(x)$, peut désigner l'ensemble de ses nœuds voisins.

Exemple (Graphe Figure 1.1)

$$\Gamma(1) = \{2,3,4\}, \Gamma(2) = \{1,5\} \quad , \Gamma(3) = \{1\} \quad , \Gamma(4) = \{1\}$$
$$\Gamma(5) = \{2,6,7\}, \Gamma(6) = \{5,7,8\} \quad , \Gamma(7) = \{5,6,8\}, \Gamma(8) = \{6,7\}$$

I.5.2. Degré d'un nœud

Pour un graphe non-orienté, on appelle degré d'un nœud x, noté $k(x)$ le nombre de liens dont le nœud est une extrémité ou bien le nombre de leurs voisins.

Exemple (Graphe Figure 1.1)

$k(1) = |\Gamma(1)| = 3$, $k(2) = |\Gamma(2)| = 2$, $k(3) = |\Gamma(3)| = 1$, $k(4) = |\Gamma(4)| = 1$,
$k(5) = |\Gamma(5)| = 3$, $k(6) = |\Gamma(6)| = 3$, $k(7) = |\Gamma(7)| = 3$, $k(8) = |\Gamma(8)| = 2$,

I.5.3. Chemin

Un chemin de longueur k entre un nœud x et un nœud y d'un graphe, noté $paths_{x,y}^{k}$ est une séquence ordonnée de nœuds ($v_0, v_1, \ldots, v_k$) tels que $x = v_0$, $y = v_k$, et $(v_{i-1}, v_i) \in E$ pour $i = 1, .., k$.

- La longueur d'un chemin est le nombre de liens dans un graphe non pondéré, et la somme des poids des liens dans un graphe pondéré,
- La distance entre x et **y** (noté, $d_{x,y}$) est le minimum des longueurs sur tous les chemins.
- Un plus court chemin entre **x** et **y** (noté $shortest\_path_{x,y}$) est un chemin dont la longueur est égale à $d_{x,y}$.

Exemple (Graphe Figure 1.1)

La distance $d_{1,6}$ entre les nœuds 1 et 6 :

$$paths_{1,6}^{3} = \{(1,2,5,6)\} \qquad |paths_{1,6}^{3}| = 1$$
$$paths_{1,6}^{4} = \{(1,2,5,7,6)\} \qquad |paths_{1,6}^{4}| = 1$$
$$paths_{1,6}^{5} = \{(1,2,5,7,8,6)\} \qquad |paths_{1,6}^{4}| = 1$$
$$d_{1,6} = shortest\_path_{1,6} = \min(3, 4, 5) = 3$$

La distance $d_{5,8}$ entre les nœuds 5 et 8 :

$$paths_{5,8}^{3} = \{(5,6,7,8)\} \qquad |paths_{5,8}^{2}| = 1$$

$$paths_{5,8}^{2} = \{(5,6,8),(5,7,8)\} \qquad |paths_{5,8}^{2}| = 2$$
$$d_{5,8} = shortest\_path_{5,8} = \min(3, 2) = 2$$

I.5.4. La densité d'un graphe

La densité ***D*** d'un graphe est définie par le rapport entre le nombre de liens effectifs divisé par le nombre de liens possibles. Dans le cas d'un graphe non orienté simple ***G = (V,E)*** la densité est le rapport :

$$\boldsymbol{D = \frac{2|E|}{|V|.([V|-1)}} \qquad \text{(I-2)}$$

Exemple

La densité de notre graphe de la Figure I.1 est : D = 0.32 / |E| = 9 , |V| = 8

I.5.5. Le coefficient de regroupement (clustering coefficient)

Le **coefficient de regroupement** (Clustering coefficient) est une mesure de la connectivité d'un graphe **non orienté**(Watts and Strogatz, 1998)**.**

Ce coefficient est en rapport avec la notion de transitivité dans le graphe. L'idée de transitivité peut être traduite de façon simple par le fait que les amis de nos amis sont souvent nos amis.

Une forte transitivité dans le graphe se traduit par le fait que, du point de vue topologique, on trouve beaucoup de triangles. Dans un graphe, on trouve des coefficients locaux (associés) à chaque nœud du graphe, et un coefficient global, qui est la moyenne arithmétique des coefficients locaux.

Le coefficient de ***clustering*** local d'un nœud ***i*** est défini comme :

$$C_i = \frac{|\boldsymbol{traingles\ de}\ \text{nœud}_i\ |}{\binom{\boldsymbol{d_i}}{\boldsymbol{2}}} \qquad \text{(I-3)}$$

Le coefficient de ***clustering*** global est défini comme :

$$C_{(G)} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \frac{C_i}{\boldsymbol{n}} \qquad \text{(I-4)}$$

Où $\boldsymbol{d_i}$ est le degré du nœud ***i*** et ***n*** le nombre de nœuds du graphe.

Exemple

Les valeurs des différents types de ***clustering*** du graphe de la Figure I.1

$C_1 = 0$, $C_2 = 0$, $C_3 = 0$, $C_4 = 0$,

$C_5 = 1/3$, $C_6 = 2/3$, $C_7 = 2/3$, $C_8 = 1$

$C_G = 8/3$

I.5.6. Les mesures de centralité

Une mesure de centralité est une fonction qui donne une valeur pour chaque nœud pour quantifier son importance, c'est-à-dire que plus la valeur est élevée, plus l'élément est important.

Par exemple, la centralité est un élément important de toute tentative visant à déterminer les personnes les plus importantes dans un réseau social comme les hommes et les femmes les plus influents sur ***Twitter***. Nous présentons ici les mesures de centralité les plus populaires :

- Le degré de centralité est la somme de tous les liens qui arrivent à un nœud, il est similaire au degré d'un nœud, et il est définie comme suit :

$$C_D(i) = \frac{d_i}{n-1} \tag{I-5}$$

Où d_i est le degré du nœud i et n le nombre de nœuds du graphe
Le degré de centralisation **(Freeman, 1978)** du graphe G étant défini comme suit :

$$C_D = \frac{\sum_{i=1}^{n}[d(u^*) - d(i)]}{(n-1)(n-2)} \tag{I-6}$$

Où ***d(u*)*** est le degré du nœud maximal du graphe.

- Centralité de proximité **(Closeness)** : la centralité de proximité d'un nœud est la proximité d'un sommet par rapport aux autres nœuds du graphe. Elle est définie comme suit **(Sabidussi, 1966)** :

$$C_c(i) = \frac{1}{\sum_{j=1}^{n} d_{i,j}} \tag{I-7}$$

Où $d_{i,j}$ est la distance entre le nœud i et j, n le nombre de nœuds du graphe

- Centralité d'intermédiarité **(Betweeness)** : la centralité d'intermédiarité d'un nœud peut être considérée comme la probabilité qu'une information transmise entre deux nœuds passe par ce nœud intermédiaire **(Brandes, 2001)**

$$C_B(i) = \sum_{J=1}^{n}\sum_{k=1}^{n}\frac{\sigma_{j,k}(i)}{\sigma_{j,k}} \tag{I-8}$$

Où $\sigma_{j,k}(i)$ est le nombre total des plus courts chemins entre les nœuds j et k qui passent par le nœud i, et $\sigma_{j,k}$ est le nombre total des plus courts chemins entre les nœuds j et k.

- Centralité des vecteurs propres (**Eigenvector centrality**) : est une mesure de l'influence d'un nœud dans un réseau. Un score de vecteur propre élevé signifie qu'un nœud est connecté à de nombreux nœuds qui ont eux-mêmes des scores élevés. Elle permet de mesure la centralité d'un nœud dans un réseau. Elle est utilisée pour identifier les nœuds les plus importants dans un réseau, en supposant que les nœuds les plus importants sont ceux qui sont connectés à d'autres nœuds importants **(Zaki and Meira, 2014)**. Elle peut être défini comme suit :

$$C_{Ev_i}(t) = \sum_{J=1}^{n} A_{ij} C_{Ev_j}(t-1) \qquad \text{(I-9)}$$

Avec la centralité au temps $t = 0$ étant $C_{Ev_j}(0) = 1$

Exemple

Les valeurs des différents types de centralité du graphe de la Figure I.1 sont résumées dans le tableau suivant

Tableau I.1. Les valeurs des différents types de centralité

I	1	2	3	4	5	6	7	8
$C_D(i)$	0.42	0.28	0.14	0.14	0.42	0.42	0.42	0.28
$C_c(i)$	0.46	0.53	0.33	0.33	0.53	0.43	0.43	0.33
$C_B(i)$	0.52	0.57	0.00	0.00	0.57	0.11	0.11	0.00
$C_{Ev}(i)$	0.12	0.22	0.04	0.04	0.48	0.52	0.52	0.39

I.5.7. le diamètre

Le diamètre d'un graphe G = (V, E), noté *Diam(G)* est le maximum des distances entre les nœuds de ce graphe :

$$Diam(G) = Max(d(V_i, V_j)/V_i, V_j \in V) \qquad \text{(I-10)}$$

Exemple

Le diamètre du graphe de la Figure I.1 est : Diam(G) = 5.

I.6.Type de réseaux complexes :

Plusieurs systèmes dans de nombreux domaines, tels que la physique, la biologie, les télécommunications, l'informatique, la sociologie, l'épidémiologie et autres, peuvent être représentés par des réseaux complexes.

La grande diversité de réseaux complexes rend difficile leur classification selon leurs propriétés communes. Cependant, dans (Newman, 2003) on trouve une tentative d'une telle classification en quatre grandes catégories de réseaux : réseaux sociaux, réseaux d'information, réseaux technologiques et réseaux biologiques.

I.6.1. Les réseaux sociaux

Un réseau social est un ensemble de personnes, de groupes de personnes ou même des organisations, des États-nations, des sites web ou des publications scientifiques avec des contacts ou des interactions spécifiques entre eux (Scott, 2017). Ces interactions peuvent être de natures très variées, tel que les modèles d'amitiés ou de parentés entre individus, les relations d'affaires entre entreprises, les mariages mixtes entre familles, ou encore le partage de mêmes opinions, les collaborations dans les publications scientifiques, etc.

L'expression "réseau social" dans l'usage habituel renvoie généralement à celle de médias sociaux, qui sont des applications web qui permettent la création et la publication de contenus générés par l'utilisateur tel que :

- Les réseaux sociaux en lignes (Facebook, Twitter, MySpace…etc.).
- Les réseaux de télécommunications, les réseaux de communication par email.
- Les réseaux de chat Messenger (Skype, Google Talk, MSN Messenger…etc.).
- Les réseaux en ligne de partage du contenu média (Youtube, Flickr, …etc.).

Figure I.6 - Graphe représentant les métadonnées de milliers de documents d'archives, documentant le réseau social de centaines d'acteurs de la Société des Nations (Grandjean, 2014)

I.6.2. Les réseaux d'information

Un exemple de réseau d'information est le réseau de citations entre les articles scientifiques. Dans la plupart des articles, nous citons des travaux antérieurs d'autres personnes sur des sujets connexes. Ces citations forment un réseau dont les nœuds sont les articles et un lien orienté de l'article A vers l'article B indique que A cite B.

Un autre exemple très important de réseau d'information est le World Wide Web, qui est un réseau de pages Web contenant des informations, reliées entre elles par des liens

hypertextes d'une page à une autre. Une page ne sera trouvée que si une autre page pointe vers elle. Ainsi, les pages forment les nœuds et les liens hypertextes forment des liens orientés.

Il existe d'autre types de réseaux tel que :

- Les réseaux peer-to-peer, qui sont des réseaux virtuels d'ordinateurs permettant le partage de fichiers entre utilisateurs d'ordinateurs sur des réseaux locaux ou étendus.
- Les réseaux linguistiques. Dans ces réseaux, les nœuds sont des mots et les liens représentent des relations entre les mots comme une cooccurrence significative dans les textes.

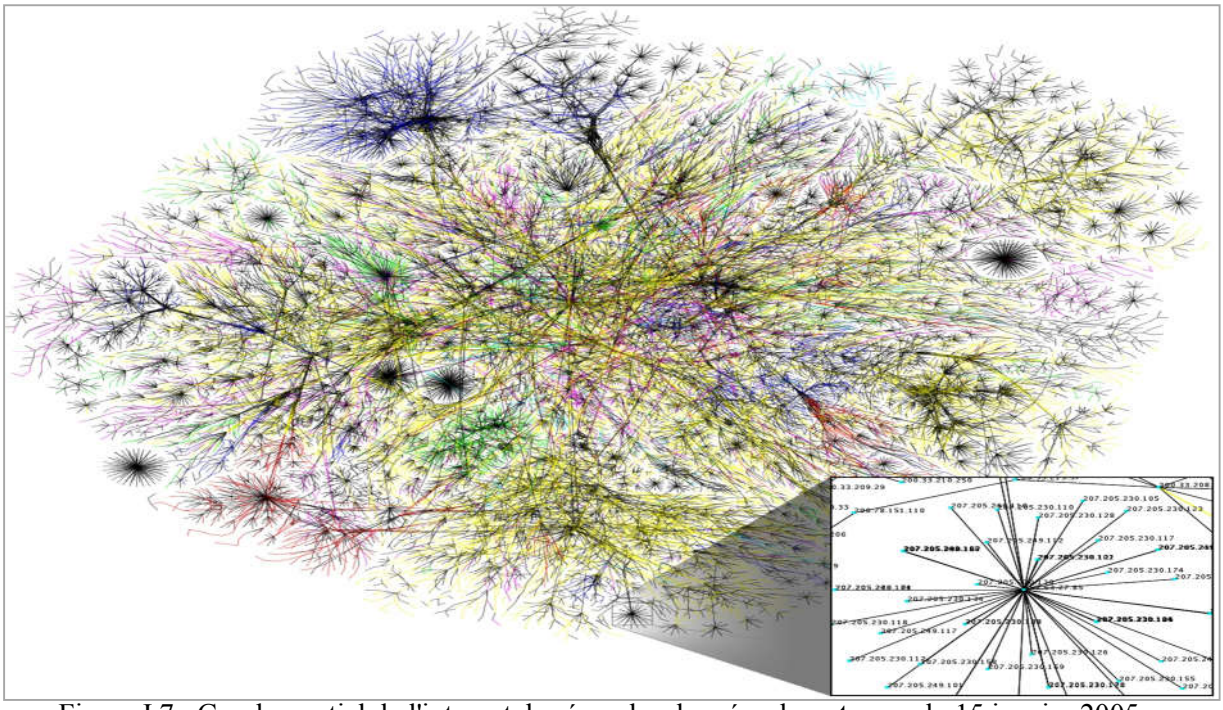


Figure I.7 : Graphe partiel de l'internet, basé sur les données de **opte.org** du 15 janvier 2005 (https://en.wikipedia.org/wiki/File:Internet_map_1024.jpg)

I.6.3. Les réseaux technologiques

Les réseaux technologiques sont les réseaux d'infrastructures physiques qui se sont développés et constituent l'épine dorsale des sociétés technologiques modernes. Il s'agit notamment d'Internet, du réseau téléphonique, des réseaux électriques, des réseaux de transport et des réseaux de livraison et de distribution (Newman, 2010).

Internet est le réseau mondial de connexions de données physiques entre les ordinateurs et les appareils de connexions. La représentation réseau la plus simple d'Internet est celui dans laquelle les nœuds du réseau représentent des ordinateurs et d'autres appareils, et les liens représentent des connexions physiques entre eux, tels que les lignes de fibre optique.

Le réseau téléphonique est le réseau de lignes fixes et sans fil qui transmet les appels téléphoniques. Il est l'un des plus anciens réseaux de communication encore en usage.

Un travail modéré a été effectué sur la structure et la fonction des réseaux de transport tels que les routes aériennes et les réseaux routiers et ferroviaires. Dans la plupart des réseaux étudiés, les nœuds représentent des lieux géographiques et les liens sont les routes entre eux.

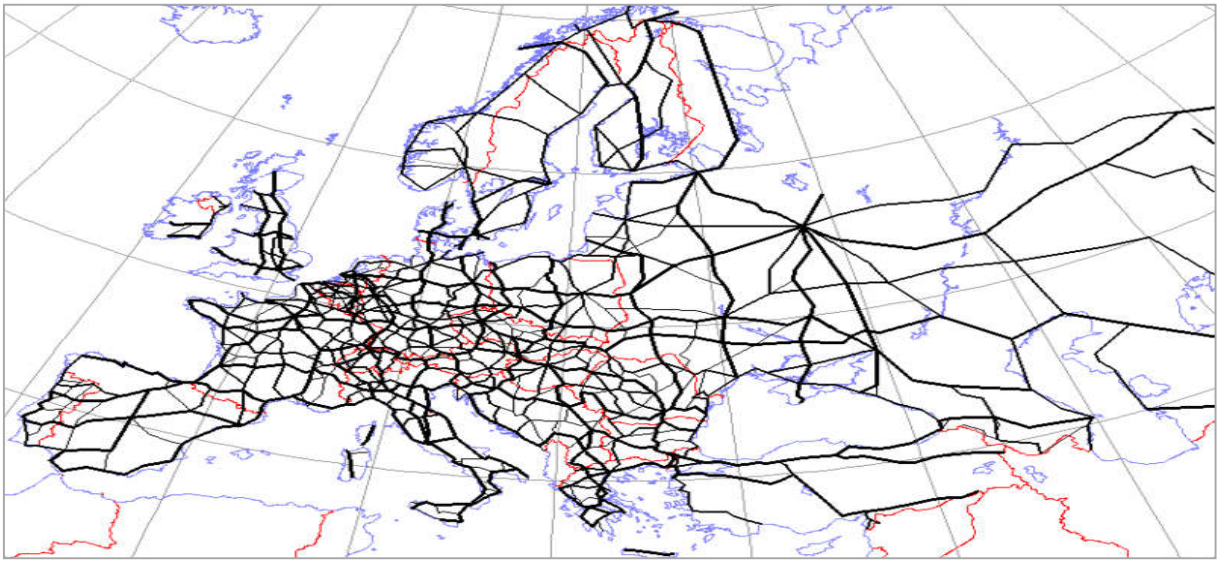


Figure I.8 : Réseau routier. Le réseau routier international en Europe.
(http://en.wikipedia.org/wiki/Image: International E Road Network.pgn.)

Un réseau électrique, dans ce contexte, est le réseau de haute tension de lignes de transmission qui assurent le transport d'énergie électrique sur de longues distances et entre les villes. Les nœuds d'un réseau électrique correspondent à des centrales et sous-stations de commutation, et les liens correspondent aux lignes à haute tension.

I.6.4. Les réseaux biologiques

On peut considérer comme un réseau biologique toutes les interactions entre des éléments biologiques vivants tel que les associations entre tout type d'entité biologique comme les protéines, les gènes, petites molécules, métabolites, ligands, maladies, médicaments etc. Prenons quelques exemples de réseaux métaboliques :

Dans un réseau métabolique, les nœuds sont des molécules simples comme l'eau, et les liens sont les réactions biochimiques qui ont lieu entre ces molécules,

Dans un réseau d'interaction protéine-protéine (Mashaghi et al., 2004), les protéines n'agissent jamais seules mais interagissent les unes avec les autres pour assurer leurs fonctions. Les protéines peuvent être considérées comme des nœuds d'un réseau complexe dans lequel deux protéines sont connectées si elles peuvent interagir physiquement. Ce type de réseaux contient des informations sur la façon dont les différentes protéines fonctionnent les unes avec les autres pour permettre un processus biologique au sein d'une cellule.

Enfin, Les neurones du cerveau sont profondément connectés les uns aux autres, ce qui entraîne la présence de réseaux complexes dans les aspects structurels et fonctionnels du cerveau, et les interactions dans le cerveau sont également complexes (Bullmore and Sporns, 2009).

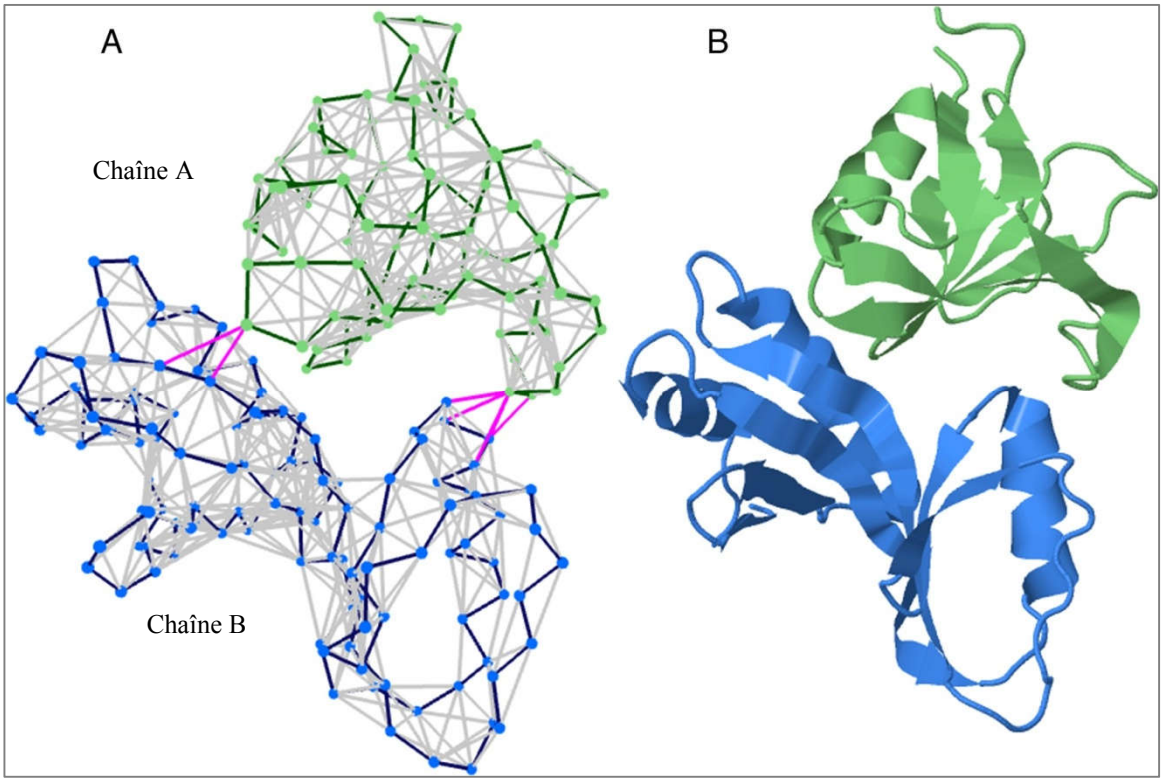


Figure I.9 : réseau d'interaction protéine-protéine (chaîne A et B). (A) Vue du réseau montrant la représentation du réseau avec les chaînes A et B colorées respectivement en bleu et vert. Les arêtes joignant les nœuds de la chaîne A et B sont surlignées en couleur magenta. (B) Structure protéique 3D (Chakrabarty and Parekh, 2016)

I.7. Les modèles des réseaux complexes

Les modèles de réseaux complexes sont utilisés pour expliquer le comportement des systèmes complexes du monde réel, et permettent aussi de générer des graphes aléatoires avec des propriétés similaires à celles des réseaux réels. Différents types de réseaux sont utilisés dans la théorie des réseaux complexes pour décrire la structure et le comportement des réseaux complexes. Il existe plusieurs types de réseaux complexes, qui se distinguent par leur structure et leurs propriétés émergentes. Voici quelques exemples :

I.7.1. Modèle de réseaux aléatoires (Random Graphs)

Les modèles de graphes aléatoires ont pour but de générer des structures ayant des propriétés similaires à celles observées dans les réseaux réels.

Un graphe aléatoire est un graphe qui est généré par un processus aléatoire. Un graphe aléatoire de taille N est un graphe à N sommets dont on a choisi aléatoirement les arrêtes. Le modèle (Erdos and Rényi, 1960) suppose que l'existence de chaque arête est

indépendante de celle des autres et que chaque arête a une probabilité p d'exister. La distribution des degrés des sommets du graphe suit une loi de Poisson.

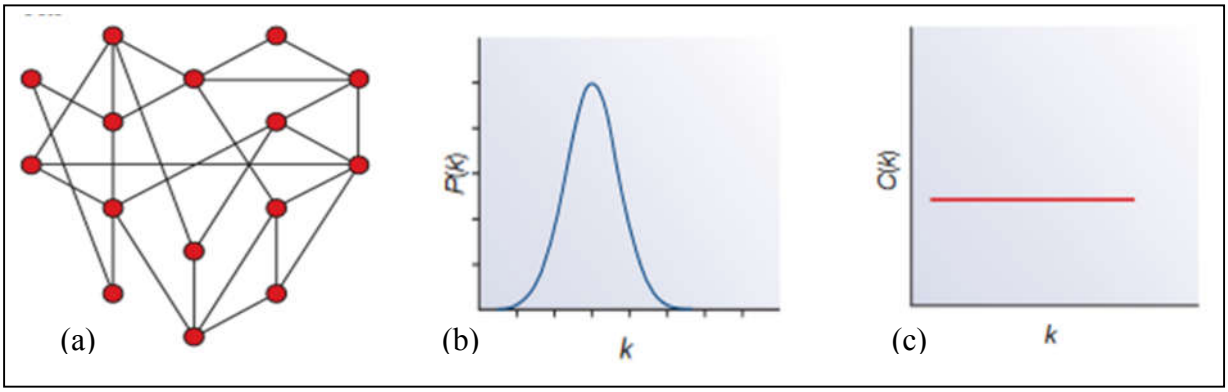


Figure I.10 : (a) la représentation d'un réseau aléatoire, (b) la distribution en loi de Poisson des degrés k, (c) le coefficient de regroupement (Barabási and Oltvai, 2004)

I.7.2. Modèle de réseaux "petit-monde"(Small World)

Un réseau est dit petit-monde (*small-world network*) quand il présente deux caractéristiques :

- La distance moyenne entre toute paire de sommets est faible.
- Le niveau de regroupement local est élevé, c'est-à-dire que les sommets sont généralement très connectés à leurs voisins immédiats. Par définition, il est nécessaire que le graphe de départ soit connexe : un chemin doit exister entre toutes les paires de sommets. La méthode de construction choisie par les mathématiciens (Watts and Strogatz, 1998) consiste à partir d'un graphe k-régulier (tous les sommets ont un même degré k) et à modifier de façon aléatoire un lien. La distance moyenne chute rapidement tandis que le niveau de regroupement reste longtemps élevé.

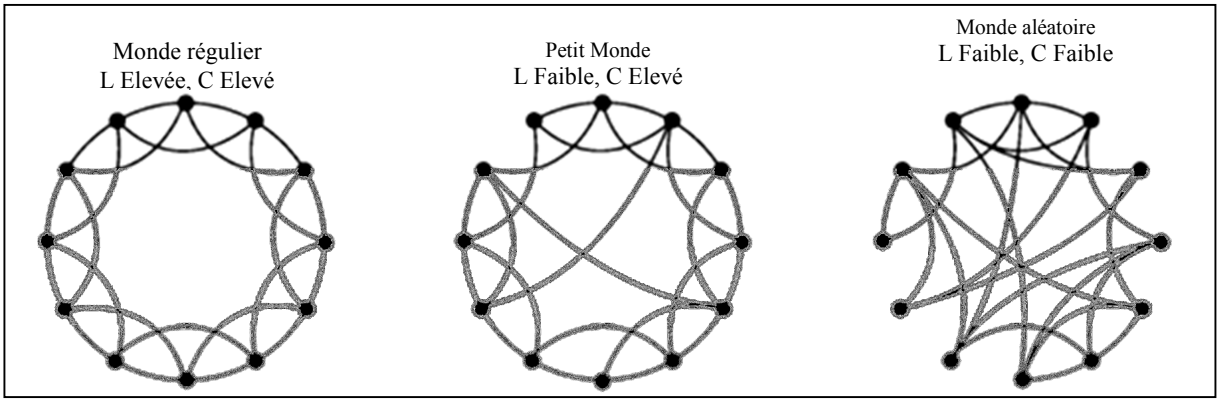


Figure I.11 : Le modèle du petit-monde où C est le coefficient de regroupement et L la longueur moyenne des plus courts chemins du graphe

I.7.3. Modèle de réseaux sans-échelles (Scale Free)

Ces derniers suivent une loi de Puissance (Barabási and Albert, 1999). Quelques nœuds centralisent la majorité des liens et constituent un hub. Il n'y a donc pas de nœud consensus qui permettrait de caractériser tous les nœuds du réseau. Quelque soit le règne ou

l'organisme, les réseaux cellulaires sont de ***type scale*** free dirigés, où les nœuds sont des métabolites et les liens représentent les réactions biochimiques.

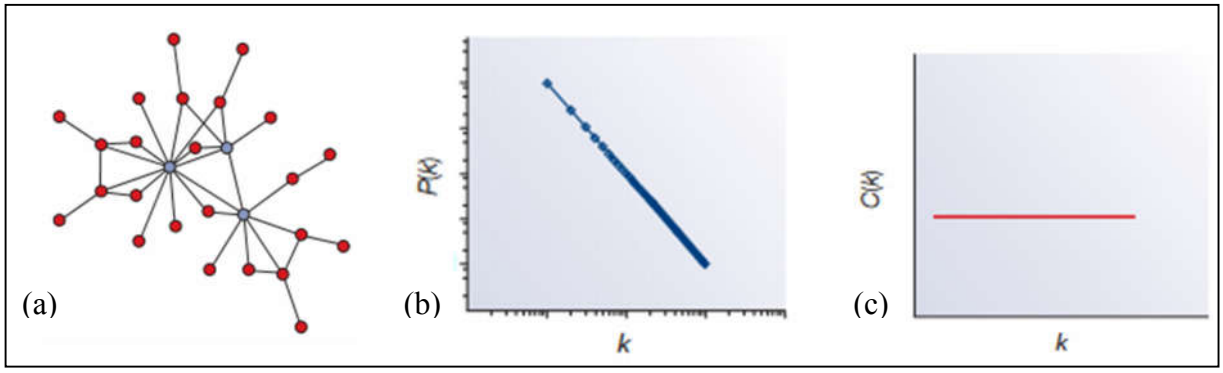


Figure I.12 : (a) un réseau sans-échelles, (b) la distribution en loi de puissance des degrés k, (c) le coefficient de regroupement (Barabási and Oltvai, 2004)

I.7.4. Modèle de Réseaux hiérarchiques

Les modèles de réseaux complexes hiérarchiques sont des modèles qui permettent de représenter la structure hiérarchique des réseaux. Ils sont largement utilisés dans les domaines de la biologie, de la physique, de la sociologie et de l'informatique pour modéliser les réseaux complexes et pour comprendre leur structure et leur fonctionnement. Ils permettent de prédire les liens manquants dans les réseaux et de comprendre la dynamique des réseaux.

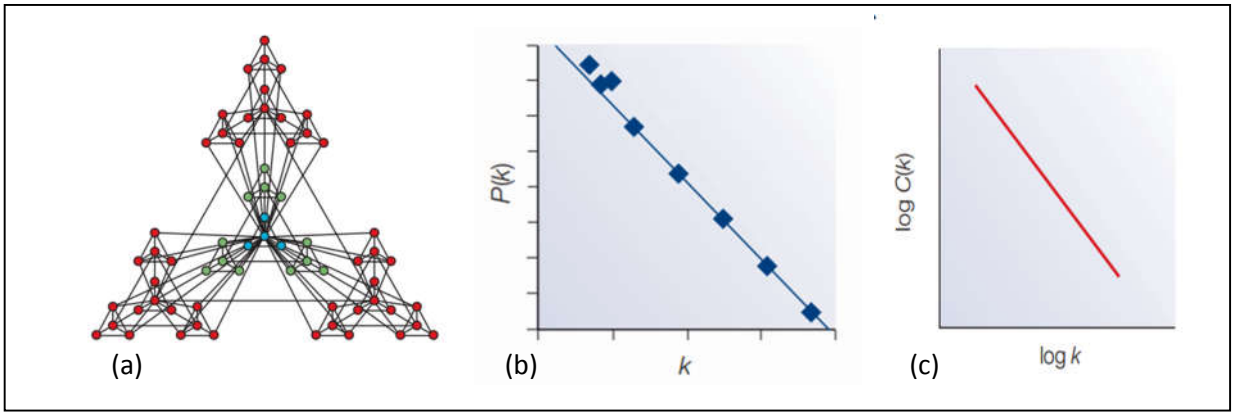


Figure I.13 : (a) un réseau hiérarchique, (b) la distribution des degrés *k*, (c) le coefficient de regroupement (Barabási and Oltvai, 2004)

I.8. Fouille d'interaction :

La fouille d'interaction dans les réseaux complexes consiste à extraire des informations sur les interactions entre les nœuds d'un réseau complexe, telles que les relations, les interactions et les flux d'information. Cette technique peut aider à comprendre la structure et les propriétés émergentes du réseau, ainsi qu'à identifier les communautés, les nœuds centraux, les modèles de diffusion et les dynamiques de l'évolution du réseau.

La fouille d'interaction dans les réseaux complexes peut être réalisée en utilisant des algorithmes d'analyse de réseaux. Ces algorithmes peuvent aider à identifier les nœuds

centraux, les groupes de nœuds hautement interconnectés, les chemins critiques et les modèles de diffusion. La fouille d'interactions peut être utilisée pour des applications telles que la recommandation de produits, la prédiction de l'évolution du réseau, la détection de communautés et l'identification des nœuds centraux importants, la diffusion de l'information, etc.

L'état de l'art de la fouille d'interaction dans les réseaux complexes se divise en plusieurs axes principaux. On peut citer notamment :

I.8.1. Tâches reliées aux nœuds

Dans les réseaux complexes, les nœuds sont des entités qui représentent des objets, des personnes, des groupes ou des concepts, qui sont reliés entre eux par des liens. Les tâches liées aux nœuds consistent à analyser et à comprendre les propriétés et les comportements des nœuds, ainsi qu'à extraire des informations sur leur importance, leur rôle et leur impact sur le réseau.

Voici quelques tâches courantes liées aux nœuds dans les réseaux complexes :

1. ***Analyse de centralité.*** Mesure de l'importance des nœuds dans le réseau en termes de leur position centrale. Les mesures de centralité les plus courantes incluent la centralité de degré, la centralité d'intermédiarité et la centralité de proximité.
2. ***Détection de communauté.*** Identification des groupes de nœuds hautement interconnectés et cohérents, qui peuvent révéler des sous-groupes ou des clusters dans le réseau.
3. ***Classification de nœuds***. Catégorisation des nœuds en fonction de leurs attributs ou de leurs comportements, par exemple pour prédire le type de nœud ou pour détecter les anomalies.
4. ***Analyse de la similarité.*** Mesure de la ressemblance ou de la similarité entre les nœuds en fonction de leurs attributs ou de leurs comportements, par exemple pour recommander des produits similaires ou pour détecter des groupes de nœuds similaires.
5. ***Analyse de la diffusion***. Etude de la propagation des informations, des idées ou des comportements dans le réseau en mesurant la diffusion des nœuds ou des liens.
6. ***Analyse de la robustesse.*** Evaluation de la résilience ou de la vulnérabilité du réseau en fonction de la suppression ou de la perturbation de certains nœuds.

I.8.2. Tâches reliées aux liens

Dans les réseaux complexes, les liens sont les relations qui relient les nœuds entre eux. Les tâches liées aux liens consistent à analyser et à comprendre les propriétés et les comportements des liens, ainsi qu'à extraire des informations sur leur importance, leur rôle et leur impact sur le réseau.

Voici quelques tâches courantes liées aux liens dans les réseaux complexes :

1. ***Prédiction de liens.*** Prédiction des liens manquants ou futurs dans le réseau en utilisant des modèles d'apprentissage automatique ou des techniques de prédiction de série temporelle.
2. ***Persistance des liens***. Elle fait référence à la stabilité ou à la durabilité des relations entre les nœuds dans un réseau complexe. Elle mesure la probabilité qu'un lien qui existe à un moment donné dans le réseau, continue d'exister à l'avenir, ou qu'un lien qui n'existe pas encore dans le réseau n'apparaisse ps à l'avenir.

I.8.3. Tâches reliées aux graphes

Dans les réseaux complexes, les graphes sont des structures mathématiques qui représentent les relations entre les nœuds et les liens. Les tâches liées aux graphes consistent à analyser et à comprendre les propriétés structurelles et topologiques du réseau en utilisant des mesures de graphes et des techniques d'analyse de graphes.

Voici quelques tâches courantes liées aux graphes dans les réseaux complexes :

1. ***Segmentation du graphe***. Division du graphe en sous-graphes ou en communautés distinctes, en fonction de certaines propriétés ou caractéristiques des nœuds et des liens. La segmentation du graphe peut aider à identifier des groupes de nœuds ayant des propriétés ou des comportements similaires.
2. ***Détection de communautés.*** Identification des groupes de nœuds dans le réseau qui ont des connexions plus denses entre eux que les nœuds en dehors du groupe.
3. ***Recherche de motifs de sous-graphes.*** Identification des motifs de sous-graphes récurrents dans le réseau, en utilisant des techniques de recherche de motifs de sous-graphes telles que l'algorithme de recherche de motif.
4. ***Analyse de sous-graphes denses.*** Extraction de sous-graphes denses ou de cliques dans le réseau, en utilisant des techniques de détection de cliques ou des algorithmes de densité maximale.
5. ***Analyse de sous-graphes bipartis*** Extraction de sous-graphes bipartis dans le réseau, qui sont des sous-graphes dans lesquels les nœuds peuvent être divisés en deux ensembles distincts, tels que les utilisateurs et les produits dans un réseau de commerce électronique.

I.9. Prédiction de liens

La prédiction de liens est un problème paradigmatique de la science des réseaux qui tente de découvrir les liens manquants ou de prédire les liens futurs (Zhou, 2021)

I.9. 1. Définition du problème

La prédiction de liens consiste à deviner les liens qui apparaitront dans le réseau en se basant sur son état actuel. Donc, l'objectif est de trouver des liens manquants ou cachés (dans les graphes statiques) qui ne figurent pas dans le graphe actuel, ou de prédire la probabilité

d'apparition de liens futurs (dans les graphes dynamiques) (Liben-Nowell and Kleinberg, 2007a).

D'autre part, la tâche de prédiction de liens concerne aussi le problème de classement de nouveaux liens, qui apparaissent dans le réseau en deux classes : *{normal, anormal}*, en fonction de l'évolution du réseau (Rattigan and Jensen, 2005).

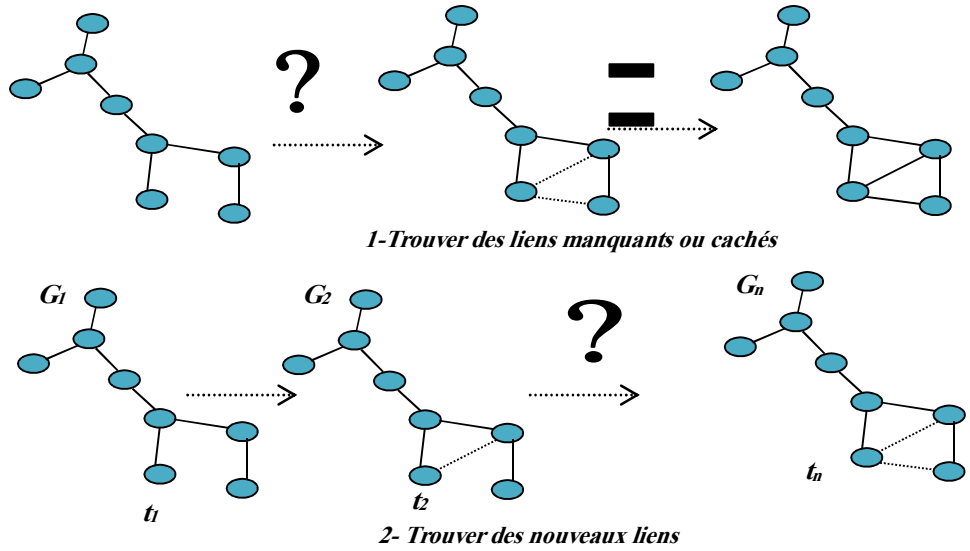


Figure I.14 : Les types de prédiction de liens

I.9.2. Description de problème

Un réseau complexe est défini par un graphe non orienté $G(V, E)$ où V est un ensemble de nœuds et $E \subseteq V \times V$ est un ensemble de liens entre paires de nœuds.

On pose $n = |V|$ (nombre de nœuds) et $e = |E|$ (nombre de liens) et on note E' l'ensemble des liens absents dans le graphe : $E' = \{ (x, y) \in V^2 \mid (x,y) \notin E\}$.

Notez que si G est un graphe complet (où il y a un lien entre chaque paire de nœuds), alors le nombre de liens est $n.(n-1)/2$. En général, la taille de E' qui représente l'ensemble des liens non existants dans le réseau est $n.(n-1)/2 - e$.

Supposons qu'il y ait des liens manquants (ou liens futurs) dans l'ensemble E', la tâche de la prédiction de lien consiste à découvrir ces liens. Pour résoudre le problème, on attribue à chaque élément $e(x,y)$ tel que $x, y \in V$ et $e(x,y) \in E'$, un score $S(x,y)$ qui représente la mesure de la probabilité d'existence de liens entre x et y.

Une liste ordonnée décroissante de tous les liens non observés est fournie en fonction de leurs scores et les liens situés en haut de la liste sont les plus possibles d'exister.

I.9.3. Domaine d'applications de prédiction de liens

La prédiction de liens peut être utilisés dans divers disciplines où les systèmes sont modélisés par des réseaux complexes :

- Elle a été utilisée en bio-informatique pour prédire les interactions protéines-protéines (PPI), la prédiction des interactions protéine-protéine (IPP) aide à saisir les racines

moléculaires de la maladie. Cependant, les expériences en laboratoire pour prédire les IPP sont limitées et coûteuses. L'utilisation basée sur l'apprentissage automatique peut non seulement identifier automatiquement les IPP, mais également fournir de nouvelles idées pour la recherche et le développement de médicaments à partir d'une alternative prometteuse (Yu et al., 2021)

- Elle est également utilisée dans les applications de sécurité pour identifier les groupes cachés de terroristes et criminels (Hasan and Zaki, 2011).
- Elle a été appliquée dans les soins de santé pour prédire la propagation des maladies épidémiques (Kamath et al., 2001), et dans le développement de stratégies pour immuniser les personnes potentiellement affectées afin de limiter la propagation de l'épidémie.
- La prédiction de lien a été utilisée dans le développement de réseaux routiers pour améliorer les itinéraires (Lü and Zhou, 2011).
- Elle a été utilisée dans les réseaux communautaires connectés en ligne où des associations futures peuvent être suggérées comme des amitiés probables. Ceci peut aider le système à recommander de nouveaux amis et ainsi augmenter leur fiabilité vis-à-vis du service (Dorogovtsev and Mendes, 2002).
- Un autre exemple de l'utilisation pratique de la prédiction de lien est sur Amazon et Produits Alibaba, où les films recommandés sur Netflix et les publicités sont affichés aux utilisateurs sur Google AdWords et Facebook (Ibrahim and Chen, 2015) .
- Une autre application est de proposer des collaborations entre chercheurs basées sur la co-écriture (Liben-Nowell and Kleinberg, 2007a).

I.9.4. Processus de prédiction de liens

Le processus de prédiction de liens, en général, peut se composer des quatre grandes étapes suivantes (Guns, 2014) :

1. L’étape de la collection de données (Data gathering).
2. L’étape du prétraitement (Preprocessing).
3. L’étape de la prédiction de liens (Prediction).
4. L’étape de l’évaluation (Evalution).

La figure1.15 ci-dessous représente les quatre phases du système dans leur ordre d’exécution. L’entrée initiale du système possède un ensemble de données bruts appelé aussi jeux de données (Data Source).

1. ***Collection de données***. Tout d’abord on divise la base de données (Graphe) en deux sous-graphes. Le premier sous-graphe est utilisé pour l’entraînement (Training) et le deuxième pour le teste (Test). Le processus de division prend en considération les dates des interactions (liens) en divisant les données selon deux périodes : une période d'entraînement et une période de test.

2. ***Prétraitement.*** Il s'applique à la fois aux réseaux d'entraînement et de test. En règle générale, certaines opérations de prétraitement de base doivent être effectuées avant la prédiction pour générer un graphe d'entrainement et de test nettoyées.
    - On ne peut pas supposer que le réseau d'entraînement et le réseau de test contiennent tous deux exactement le même ensemble de nœuds. Pour permettre une comparaison équitable, nous devons limiter notre analyse aux nœuds qui sont communs aux réseaux d'apprentissage et de test. Le processus de prétraitement consiste ici à éliminer les nœuds isolés et les nœuds in-communs des deux sous-graphes Train et Test.
    - Par ailleurs, il est difficile de prédire quoi que ce soit sur les nœuds isolés (nœuds sans voisins), car on n'en dispose d'aucune information (basée sur le réseau). Par conséquent, les nœuds isolés doivent être éliminés.
    - De même, les nœuds de bas degré sont parfois écartés car on a trop peu d'informations à leur sujet (Liben-Nowell and Kleinberg, 2007a)

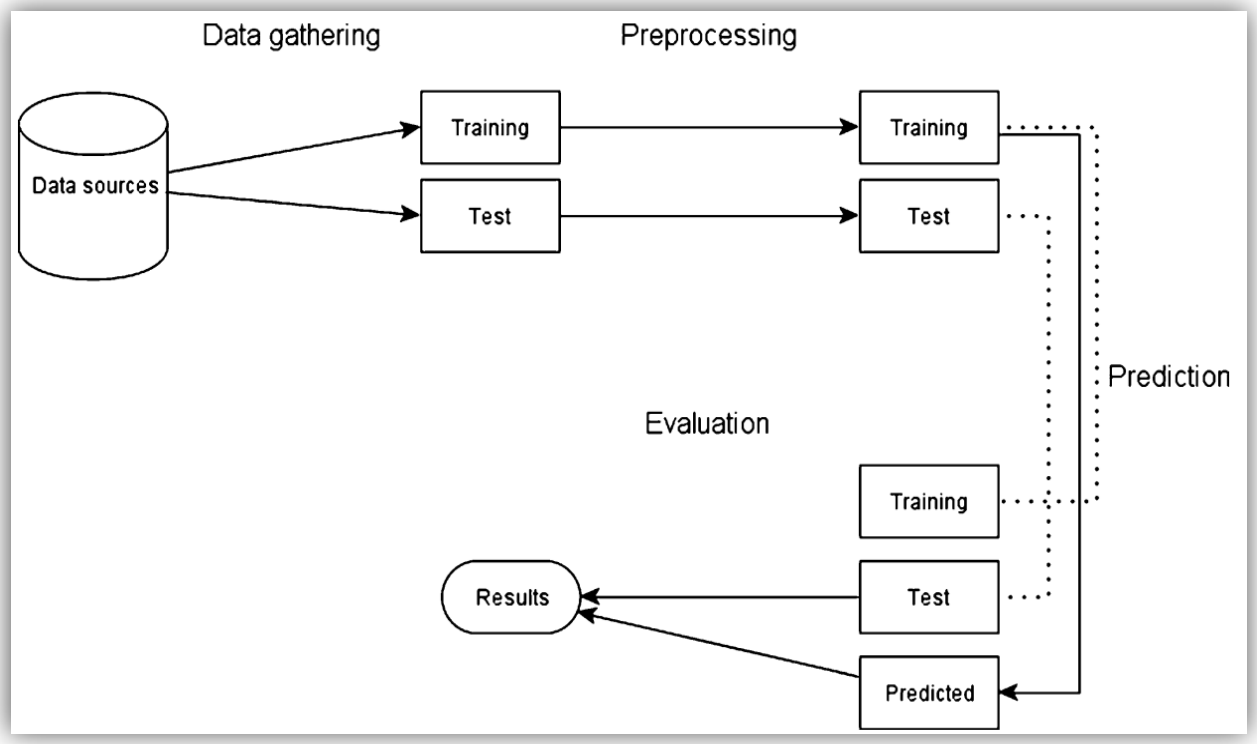


Figure I.15- Présentation du processus de prédiction de liens (Guns, 2014).

3. ***Prédiction***. Elle consiste à prendre le graphe d'entrainement nettoyé comme une source d'entrée, et choisir un prédicteur, d'une fonction ou d'un algorithme qui calcule un score de vraisemblance pour chaque paire de nœuds. L'application d'un prédicteur au graphe d'entrainement prétraité donne un certain nombre de prédictions. En pratique, l'étape de prédiction aboutit à une liste de liens potentiels avec une vraisemblance associée à un score. En classant les liens potentiels par ordre décroissant de score et en choisissant un seuil, on peut obtenir à la fin un nouveau graphe prédit.

4. ***Evaluation des performances.*** Evaluer les algorithmes utilisés en comparant le graphe de test et le graphe prédit et en calculant le nombre de liens existant en plus ou en moins.

I.9.5. Classification des approches de prédiction de liens

La prédiction de lien est une tâche d'analyse de réseau qui consiste à prédire la probabilité d'une connexion ou d'un lien entre deux nœuds d'un réseau. Il existe différentes approches pour la prédiction de liens qui peuvent être classées selon différents critères classifications.

Nous adoptons dans notre travail la classification proposée dans (Wang and Le, 2020) et représentée ci-dessous dans la Figure I.8 où les auteurs ont classé les approches de prédiction de liens dans deux grandes classes : les approches basées sur l'apprentissage et les approches basées sur les caractéristiques.

Dans chacune de ces grandes classes apparaît un ensemble de familles de méthodes explorées afin de réaliser la tâche de prédiction de liens.

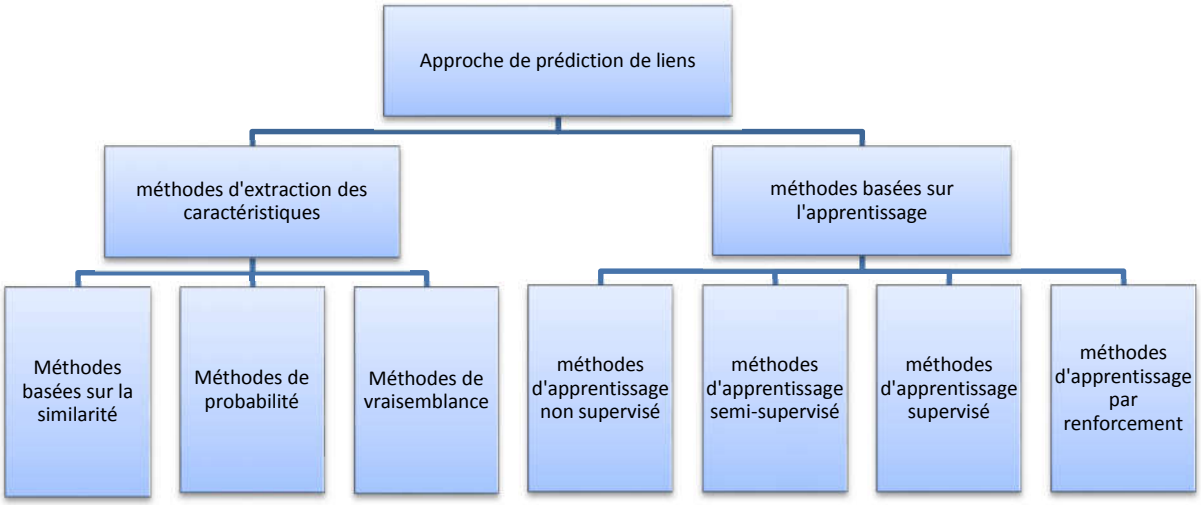


Figure I.16 - classification des approches de prédiction de liens (Wang and Le, 2020).

I.9.5.1. Les méthodes basées sur l'extraction des caractéristiques

Les méthodes d'extraction des caractéristiques peuvent être utilisées pour extraire des caractéristiques à partir de données de réseau. Ces caractéristiques peuvent ensuite être utilisées pour entraîner des modèles de prédiction de lien, tels que des méthodes basées sur la similarité, probabilistes et de vraisemblance.

I.9.5.1.1 Méthodes basées sur la similarité

Les méthodes basées sur la similarité sont largement utilisées pour la prédiction de liens dans les réseaux complexes grâce à la facilité de leur mise en œuvre. Ces méthodes consistent à mesurer la similarité entre les nœuds du réseau pour prédire les liens manquants et ce en utilisant des caractéristiques concernant les voisinages et les chemins entre les nœuds

ainsi que les caractéristiques concernant les nœuds. Dans le chapitre qui suit, nous examinerons ces méthodes de façon détaillée.

I.9.5.1.2 Méthodes basées sur la vraisemblance

Les méthodes de vraisemblance se divisent en deux grands types d'algorithmes : les algorithmes des modèles de structure hiérarchiques et les algorithmes des modèles de blocs stochastiques.

Les algorithmes des modèles de structure hiérarchiques sont utilisés pour prédire les liens manquants dans les réseaux à une certaine structure hiérarchique afin de révéler des relations structurelles cachées. Cette méthode s'applique aux réseaux avec une hiérarchie évidente, tels que les réseaux d'attaques terroristes et les réseaux de la chaîne alimentaire (Clauset et al., 2008) ,(Girvan and Newman, 2002).

Les algorithmes des modèles de blocs stochastiques sont utilisés pour certains réseaux qui ne rentrent pas dans un schéma hiérarchique (Marsden, 2006). Ils sont basés sur l'estimation de la vraisemblance. Leur idée de base est de regrouper les nœuds du réseau selon les caractéristiques de la modularité du réseau. Savoir si deux nœuds sont reliés par des liens est déterminé par leurs groupes. Les modèles de blocs stochastiques peuvent prédire les liens manquants et juger quels liens sont considérés comme mauvais (comme la mauvaise compréhension de l'interaction des protéines) en fonction de la crédibilité. Même si les algorithmes basés sur ces modèles sont en moyenne plus performants que ceux basés sur les modèles hiérarchiques, ils partagent tout de même avec eux le même problème de complexité de calcul temporelle élevé.

I.9.5.1.3 Méthodes basées sur la probabilité

Les méthodes basées sur la probabilité sont une autre approche courante pour la prédiction de liens dans les réseaux. Ces méthodes utilisent des modèles probabilistes pour estimer la probabilité qu'un lien existe entre deux nœuds dans le réseau. Voici quelques exemples de méthodes basées sur la probabilité : les méthodes relationnelles probabilistes (Taskar et al., 2012), Les méthodes relationnelles d'entités (Heckerman et al., 2004) et les méthodes relationnelles stochastiques (Yu et al., 2006).

L'avantage des méthodes probabilistes réside dans leur grande précision de prédiction puisqu'elles utilisent les informations de la structure du réseau et impliquent les informations des attributs des nœuds. Cependant, leur complexité de calcul et leurs paramètres non universels limitent son champ d'application.

I.9.5.2. Les méthodes basées sur l'apprentissage

Ces méthodes traitent la prédiction de liens comme un problème à deux classes : Chaque paire de nœuds non connectés correspond à une instance et s'il existe un lien entre ces nœuds, la paire est marquée comme positive, sinon, la paire est marquée comme négative.

Ces méthodes utilisent diverses caractéristiques des nœuds et des liens, telles que la similarité entre les nœuds, le degré des nœuds, la centralité des nœuds et d'autres caractéristiques encore (Kumar et al., 2020).

Les méthodes actuelles de prédiction de liens basées sur l'apprentissage peuvent être divisées en quatre catégories selon le type d'apprentissage utilisé, à savoir : l'apprentissage supervisé, l'apprentissage semi-supervisé, l'apprentissage non supervisé et l'apprentissage par renforcement.

I.9.5.2.1. Méthodes d'apprentissage non supervisée

Comme leur nom l'indique ces méthodes utilisent des techniques et des algorithmes d'apprentissage non supervisé pour apprendre à prédire les liens manquants dans les réseaux. De ce fait, ces méthodes utilisent des données non étiquetées (non classées), comme l'information sur les nœuds, la topologie et la théorie social pour calculer la similarité entre les paires de nœuds non connectés.

L'approche non supervisée a l'avantage d'avoir généralement une faible complexité et donc un calcul simple et rapide. De nos jours, les méthodes de prédiction de liens basées sur l'apprentissage non supervisées les plus connues sont : DeepWalk (Perozzi et al., 2014), LINE (Large-scale Information Network Embedding) (Tang et al., 2015), GraRep (Graph Representations) (Cao et al., 2015), DNGR (Deep Neural Networks for Graph Representation) (Cao et al., 2016), SDNE (Structural Deep Network Embedding) (Wang, 2016), Node2Vec (Scalable Feature Learning for Networks) (Grover and Leskovec, 2016), HOPE (High-Order Proximity preserved Embedding)(Ou et al., 2016), GraphGAN (Graph Representation Learning with Generative Adversarial Nets) (Wang et al., 2018) et Struct2Vec (Learning Node Representations from Structural Identity) (Ribeiro et al., 2017). Notons que ces méthodes sont toutes basées sur la représentation matricielle du graphe.

I.9.5.2.2. Méthodes d'apprentissage semi-supervisé

Les méthodes d'apprentissage semi-supervisé combinent des données étiquetées et non étiquetées pour l'apprentissage de modèles servant à prédire les liens manquants dans les réseaux. Elles utilisent diverses caractéristiques des nœuds et des liens, telles que la similarité entre les nœuds, le degré des nœuds, et la centralité des nœuds.

Actuellement, les méthodes de prédiction de liens basées sur l'apprentissage semi-supervisé les plus couramment utilisées sont : GCN (Graph Convolutional Networks) (Kipf and Welling, 2017), GNN (Graph Neural Network) (Scarselli et al., 2009), GAN (Generative Antagonism Network) (Lei et al., 2019), LSTM (Long-Short-Term Memory) (Chen et al., 2021), GAE (Graph Auto-Encoder) (Kipf and Welling, 2016) et GAT (Graph Attention Network)(Gu et al., 2019).

I.9.5.2.3. Méthodes d'apprentissage supervisé

L'apprentissage supervisé consiste à entrainer un modèle à partir de données préalablement étiquetées ou annotées (classées). Il est utilisé aussi bien pour ll'analyse prédictive que pour la prédiction de liens dans les réseaux.

Les méthodes d'apprentissage supervisé les plus utilisées dans la prédiction de liens sont : SVM (Support Vector Machine) (Laishram, 2015), KNN (K-Nearest Neighbor) (Aouay et al., 2014), LR (Logistic Regression) (Zhou et al., 2017), EL (Ensemble learning) (Pachaury et al., 2018), RF (Random Forrest) (Guns and Rousseau, 2014), MP (Multilayer Perceptron) (Kastrin et al., 2016), NB (Naïve Bayes) (Valverde-Rebaza et al., 2015) et MF (Matrix Factorization) (Menon and Elkan, 2011).

I.9.5.2.4. Méthodes d'apprentissage par renforcement

L'apprentissage par renforcement peut être vue comme intermédiaire entre l'apprentissage supervisé et non supervisé. Il enregistre les résultats de différentes itérations, il met à jour son modèle et utilise le modèle actuel pour guider la prochaine itération. Une fois que l'itération suivante a obtenu des résultats, elle met à jour le modèle et itère ou répète jusqu'à ce que le modèle converge.

Parmi les méthodes d'apprentissage par renforcement pour la prédiction de liens, on peut citer : GCPN (Graph Convolutional Policy Network) (You et al., 2019) et GTPN(Graph Transformation Policy Network) (Do et al., 2018).

I.10.Conclusion

Le but de ce chapitre était de mettre en évidence l'importance des réseaux complexes comme technologie permettant de modéliser plusieurs systèmes naturels ou artificiel, qui touchent à plusieurs secteurs de notre vie, d'où une grande variété des méthode et d'algorithmes qui sont appliqués pour l'analyse de tels réseaux.

Les réseaux de transports, sociaux, technologies, d'information, et même de biologies sont considérés comme des réseaux complexes. Elles ont généralement une topologie extrêmement compliquée, dynamique, et auto-organisée.

La représentation des réseaux complexes sous forme de graphe a significativement facilité l'analyse de ces réseaux. Cette représentation a permis de calculer et d'extraire plusieurs informations et propriétés caractérisant les nœuds, les liens et le réseau complet. Dans ce chapitre nous avons détaillé les concepts et les caractéristiques les plus importants dans ce domaine. Ensuite nous avons exposé les nombreux domaine d'application de l'analyse des réseaux complexe en mettant l'accent sur la prédiction de liens qui constitue notre cas d'étude dans cette thèse.

CHAPITRE –II-

Prédiction de liens basée sur la similarité : Etat de l'art

Chapitre –II-

Prédiction de liens basée sur la similarité : Etat de l'art

II.1. Introduction

Les réseaux complexes sont fortement dynamiques : de nouveaux nœuds et liens sont ajoutés aux graphes d'un instant à un autre. Comprendre cette évolution est un problème très complexe dû à l'existence d'un nombre important de paramètres.

Comprendre, maîtriser et contrôler cette évolution constitue un domaine de recherche important et varié. Dans les réseaux complexes, nous nous intéressons souvent aux liens entre les nœuds, et la prédiction de liens est l'un des problèmes les plus fondamentaux et les plus essentiels dans les réseaux complexes.

Dans ce chapitre, nous détaillons les approches de prédiction de liens basées sur la similarité. Nous passons en revue les mesures locales, globales et quasi-locales les plus reconnues, qui ont tenté de trouver une solution optimale à ce problème.

.

II.2. Méthodes de prédiction de liens basées sur la similarité

En raison de sa large portée (biologie, physique, informatique et bien d'autres domaines), de nombreuses études se sont orientées vers la prédiction de liens dans les réseaux complexes. Plusieurs méthodes ont été conçues et appliquées pour prédire des liens futurs et/ou découvrir des liens manquants dans différents types de réseaux.

La plupart des algorithmes ou modèles de prédiction de liens qui ont été proposés s'appuient sur la notion de similarité et visent à attribuer généralement un score à chaque paire de sommets afin de quantifier sa probabilité d'existence. Ces méthodes ont connu un succès remarquable et sont largement utilisés en raison de leur précision de la prédiction et leur efficacité de calcul (Wang and Le, 2020).

Les algorithmes basés sur la similarité peuvent être classés en trois principales catégories (Lü and Zhou, 2011) (Kumar et al., 2020) : les algorithmes basés sur des informations locales des nœuds qui exploitent les propriétés de voisinage d'une paire de nœuds donnée, les algorithmes basés sur des informations globales qui s'appuient sur les chemins du graphe et enfin les algorithmes Quasi-Locaux qui exploitent aussi bien les propriétés de voisinage que les chemins du graphe).

La complexité algorithmique est calculée pour ces méthodes, en se basant sur trois variables qui sont considérés pour mesurer la taille du problème : le nombre *n* de nœuds, le nombre *e* de liens et le degré maximum d'un nœud noté *k*. Cette complexité dépend de la façon dont les ensembles de voisins communs de deux nœuds sont stockés. Le cas le plus simple est celui où tous les voisins sont dans une liste.

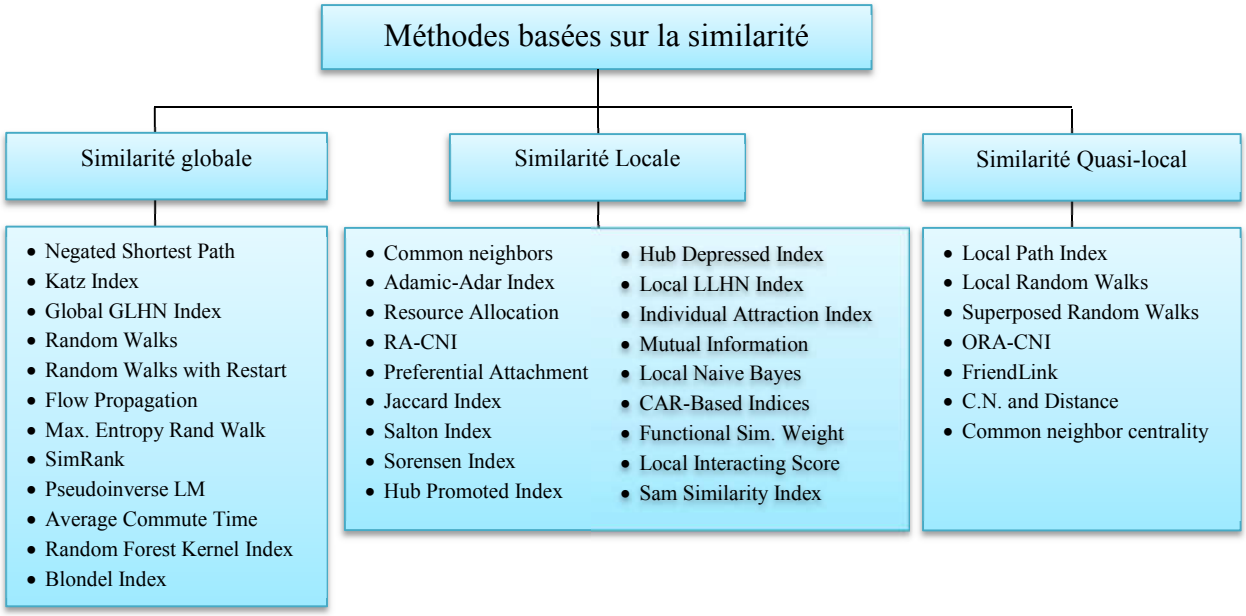


Figure -II-1 : Classification des méthodes de similarité

II.3. Méthodes de similarité locale

Les méthodes basées sur la similarité locale reposent sur des informations structurelles comme le voisinage. Ces méthodes sont rapides, efficaces et hautement parallélisables et donc plus nombreuses que les autres types de méthodes. En revanche, dans ces méthodes les attributs des nœuds sont cachés ou ne sont généralement pas disponibles (Lü and Zhou, 2011).

Parmi les mesures que l'on retrouve couramment dans la littérature, on peut citer :

II.3.1. Common Neighbors (Liben-Nowell and Kleinberg, 2007a).

Cette méthode est la plus citée et utilisée. La méthode des "Common Neighbors" (ou voisins communs) est une méthode de prédiction de liens dans un réseau qui repose sur le nombre de voisins communs entre deux nœuds. Si deux nœuds ont de nombreux voisins communs, cela suggère qu'il est plus probable qu'ils soient liés par un lien dans le futur. Cette mesure simple et intuitive est définie comme suit :

$$S_{xy}^{CN} = |\Gamma(x) \cap \Gamma(y)| \qquad \text{(II-1)}$$

Les valeurs des scores obtenues par cette méthode ne sont pas normalisés. Elles varient entre 0 et le max degré maximal des nœuds.

L'utilisation de cette mesure pour calculer la similarité pour tous les liens possibles aboutit à une technique de prédiction de liens dont la complexité temporelle est $O(nk^3)$

II.3.2. Adamic Adar index (Adamic and Adar, 2003).

Cette mesure permet d'ajouter la somme des poids des nœuds qui sont connectés aux deux nœuds x et y. Ce poids est dépendant du degré des nœuds. Elle est basée sur l'hypothèse que les voisins en commun sont plus importants lorsque ceux-ci sont moins courants, c'est-à-dire qu'ils ont moins de voisins en commun. Cette mesure donne donc une valeur plus grande aux voisins en commun qui sont rares dans le réseau. Elle est définie comme suit :

$$S_{xy}^{AA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log k(z)} \qquad \text{(II-2)}$$

- Cette mesure attribue à chaque lien une valeur non normalisée. Il est préférable de normaliser les valeurs avant d'appliquer le seuil de jugement.

- La complexité temporelle de l'ensemble du calcul de l'indice Adamic-Adar pour tous les paires de nœuds dans un graphe de v nœuds est : $O(nk^3)$

II.3.3. Resource allocation index (Zhou et al., 2009) .

Dans cette mesure, on considère une paire de nœuds, x et y, qui ne sont pas connectés directement, et on suppose que le nœud x doit donner des ressources à y, avec leurs voisins communs comme émetteurs. Cette mesure de similarité est basée sur l'hypothèse que deux nœuds qui partagent des voisins rares ont une probabilité plus élevée de former un lien que deux nœuds qui partagent des voisins très courants. Elle est définie comme suit :

$$S_{xy}^{RA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k(z)} \qquad \text{(II-3)}$$

- Il est préférable de normaliser les valeurs avant d'appliquer le seuil, car sinon, les résultats obtenus ne sont pas normalisés, c'est-à-dire ne variant pas entre 0 et 1.

- Cette méthode a aussi la même complexité temporelle $O(nk^3)$ que la méthode précédente.

II.3.4. Resource Allocation Based on Common Neighbor Interactions (Zhang et al., 2014)**.**

Cette mesure est motivée par le processus d'allocation des ressources où chaque nœud envoie une unité de ressource à ses voisins. Cependant, cette méthode prend également en considération le retour des ressources dans la direction opposée : elle combine les principes de Resource Allocation Index (RA) et Common Neighbors (CN). Elle calcule la similarité entre deux nœuds en utilisant la somme des ressources allouées à leurs voisins en commun, pondérée par le degré de chaque voisin. En d'autres termes, elle mesure à quel point deux nœuds partagent leurs voisins en commun et à quel point ces voisins sont importants en termes de ressources. Elle est définie comme suit :

$$S_{xy}^{RA-CNI} = \sum_{Z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{k(z)} + \sum_{e_{i,j}\in E,\Gamma(i)<\Gamma(j),i\in\Gamma(x),j\in\Gamma(y)} \left(\frac{1}{k(i)} - \frac{1}{k(j)}\right) \quad \text{(II-4)}$$

- Pour cette mesure, les scores donnés pour les nouveaux liens ne sont pas normalisés. Donc, pour les utiliser dans une comparaison, il est préférable de les normaliser.

- La complexité de calcul de cette méthode est estimé à ***O(nk⁴).***

II.3.5. Preferential Attachment Index (Barabási and Albert, 1999; Mitzenmacher, 2004)

Cette mesure suppose que la probabilité de création d'un lien entre deux nœuds dépend du nombre de liens que chacun des nœuds possède déjà. Plus précisément, la probabilité qu'un nouveau lien soit créé entre un nœud *x* et un nœud *y* est proportionnelle au produit du degré des deux nœuds *x* et *y*. Cette mesure repose sur l'hypothèse que les nœuds qui ont déjà un grand nombre de liens sont plus susceptibles de se lier à de nouveaux nœuds que ceux qui ont moins de liens. Ainsi, les nœuds qui ont un degré élevé ont une plus grande probabilité d'acquérir de nouveaux liens que les nœuds ayant un degré faible. Le score associé à la possibilité d'existence d'un lien entre *x* et *y* est le suivant :

$$S_{xy}^{PA} = k(x) * k(y) \quad \text{(II-5)}$$

- Les valeurs des scores obtenues par cette méthode ne sont pas normalisées. Elles varient entre 0 et le carré du dgré maximal.

- Dans sa forme la plus simple, la complexité de cette méthode est : ***O(nk²)***

II.3.6. Jaccard index (Jaccard, 1901)

Pour chaque paire de nœuds *(x, y)* qui ne sont pas connectés dans un graphe, l'indice de Jaccard calcule la similarité entre les ensembles de voisins communs de ces deux nœuds. L'indice de Jaccard est défini comme le nombre de voisins communs des deux nœuds divisé par le nombre total de voisins des deux nœuds.

L'indice de Jaccard normalise le score obtenu par la méthode des voisins communs, cette mesure est définie comme suit :

$$S_{xy}^{JI} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \quad \text{(II-6)}$$

- Cet indice varie entre 0 et 1, où 0 signifie qu'il n'y a pas d'éléments communs entre les deux ensembles des voisins et 1 signifie que les deux ensembles sont identiques.

- La complexité de l'indice de Jaccard dépend de la façon dont les ensembles de voisins communs des deux nœuds sont calculés. Elle est égale à ***O(nk³).***

II.3.7. Salton index (Salton and McGill, 1983).

L'indice de Salton, également appelé indice cosinus de Salton, est utilisé pour trouver l'indice de similarité basé sur l'angle cosinus entre les lignes de la matrice de contiguïté. Le score de prédiction de lien est défini comme suit:

$$S_{xy}^{SI} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{\Gamma(x) * \Gamma(y)}} \quad \text{(II-7)}$$

- La valeur de S_{xy}^{SI} est comprise entre 0 et 1, où une valeur de 1 indique que les nœuds *x* et *y* sont très similaires tandis qu'une valeur de 0 indique qu'ils sont très différents.

- La complexité de cette méthode est aussi $O(nk^3)$.

II-3.8. Sorensen index (McCune et al., 2002; Sorensen, 1948) .

Cet indice a été développé par Thorvald Sørensen en 1948 pour comparer la similarité entre différentes communautés écologiques. Malgré sa similitude avec l'indice Jaccard, puisqu'il mesure le score relative d'une intersection d'ensembles de voisins, l'indice de Sørensen peut être utilisé pour mesurer la similitude entre deux nœuds en termes de leurs voisins communs. Il est défini comme suit :

$$S_{xy}^{Sorensen} = \frac{2|\Gamma(x) \cap \Gamma(y)|}{k(x) + k(y)} \quad \text{(II-8)}$$

- Cet indice varie de 0 à 1, où une valeur de 1 indique que les deux nœuds ont les mêmes voisins, tandis qu'une valeur de 0 indique que les nœuds n'ont pas de voisins communs. Plus la valeur de l'indice est élevée, plus la similarité entre les deux nœuds est importante.

- La complexité temporelle de cette méthode pour tous les liens possibles est : $O(vk^3)$.

II.3.9. Hub promoted index (Ravasz et al., 2002).

C'est une méthode de prédiction de liens qui se base sur le fait que les nœuds ayant une connectivité élevée (les hubs) ont tendance à favoriser la création de nouveaux liens. C'est une mesure définie comme le rapport des voisins communs des nœuds *x* et *y* au minimum des degrés des nœuds *x* et *y*. Il est calculé comme suit :

$$S_{xy}^{HPI} = \frac{|\Gamma(x) \cap \Gamma(y)|}{min\{k(x), k(y)\}} \quad \text{(II-9)}$$

- Le score S_{xy}^{HPI} est normalisé : ces valeurs possibles varient entre 0 et 1.
- La complexité du calcul de cette mesure est : $O(nk^3)$.

II.3.10. Hub depressed index (Ravasz et al., 2002).

C'est une mesure de prédiction de liens qui se base sur les nœuds ayant une connectivité élevée (les hubs), comme la méthode précédente. Cette mesure est définie comme le rapport des voisins communs des nœuds *x* et *y* au maximum des degrés des nœuds *x* et *y*. Elle est calculée comme suit :

$$S_{xy}^{HDI} = \frac{|\Gamma(x) \cap \Gamma(y)|}{max\{k(x), k(y)\}} \quad \text{(II-10)}$$

- Le score S_{xy}^{HDI} est normalisé : ces valeurs varient entre 0 et 1.
- La complexité du calcul de cette mesure est : $O(nk^3)$ qui est la même que celle de la mesure S_{xy}^{HPI} .

II.3.11. Leicht-Holme-Newman index (Leicht et al., 2006)

C'est une variante de la méthode des voisins communs, similaire à l'index de Salton. Elle mesure la similarité de voisinage entre les nœuds *x* et *y* en tenant compte de leurs degrés respectifs. C'est le rapport des voisins communs des nœuds *x* et *y* au produit des degrés des deux nœuds. L'indice est calculé comme suit :

$$S_{xy}^{LHNI} = \frac{|\Gamma(x) \cap \Gamma(y)|}{k(x) * k(y)} \quad \text{(II-11)}$$

- Les valeurs possibles de l'indice LHN varient entre 0 et 1. Les paires de nœuds ayant une valeur plus élevée sont considérées comme plus similaires.

- Cette méthode a une complexité de $O(nk^3)$, comme les autres méthodes basées sur le voisinage.

II.3.12. The Individual Attraction Index (Dong et al., 2011)

Cette mesure est basée sur l'idée que les nœuds avec des degrés élevés sont plus attractifs pour former de nouvelles connexions que les nœuds avec des degrés faibles. On suppose que deux nœuds avec le même nombre de voisins partagés sont plus possible d'être connectés si leurs voisinages sont également fortement connectés parmi eux. Ces liens sont appelés liens internes. Cette mesure est calculée comme suit :

$$S_{xy}^{IA1} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{e_z}{k(z)} \quad \text{(II-12)}$$

Ooù e_z est le nombre de liens entre le nœud z avec le nœud *x, y* et leurs voisins communs.

- Les valeurs d'*IA1* ne sont pas normalisées : elles peuvent varier de 0 à l'infini en fonction de la structure du graphe. Il est donc conseillé de les normaliser pour pouvoir appliquer un seuil d'arbitrage.

- La complexité de calcul de cette version-là d'***IA1*** est : ***O(nk⁴).***

Une autre version de cette méthode qui est moins coûteuse en termes de calculs est appelée l'indice d'attraction individuelle simple (Dong et al., 2011). Dans cette version, pour chaque voisin commun, nous calculons seulement l'inverse du degré de nœud commun, et les liens internes dans son ensemble. L'indice d'attraction individuel simple est défini par :

$$S_{xy}^{IA2} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k(z)} * \frac{e+2}{|\Gamma(x) \cap \Gamma(y)|} \qquad \text{(II-13)}$$

Ooù *e* est le nombre de liens dans le réseau qui sont composées de tous les voisins communs du nœud *x, y.*

- Les valeurs d'***IA2*** ne sont pas normalisées non plus : elles peuvent varier de 0 à l'infini en fonction de la structure du graphe. Donc, il vaut mieux les normaliser avant d'utilisée un seuil de jugement.

- La complexité de calcul de cette version moins coûteuse ***est*** : ***O(nk³).***

II.3.13. Mutual Information (Tan et al., 2014).

Cette méthode calcule l'auto-information conditionnelle de l'existence d'un lien en se basant sur l'ensemble des voisins communs. Elle est basée sur la mesure de l'information mutuelle entre les nœuds dans un graphe. Elle s'appuie sur l'hypothèse que deux nœuds qui partagent des voisins similaires ont plus de chance de former un lien dans le futur. Le score entre deux nœuds est calculé comme suit :

$$s_{xy}^{MI} = -I\left(e_{x,y} \middle| \Gamma(x) \cap \Gamma(y)\right) = -I(e_{x,y}) + \sum_{z \in \Gamma(x) \cap \Gamma(y)} I(e_{x,y}; z) \qquad \text{(II-14)}$$

Ooù $e_{x,y}$ est l'auto-information de *x* et *y* étant connectés, qui est calculée comme suit :

$$I(e_{x,y}) = -log_2\left(1 - \prod_{i=1}^{\Gamma(y)} \frac{|E| - |\Gamma(x)| - i + 1}{|E| - i + 1}\right)$$

et $I(e_{x,y}; z)$ est l'information mutuelle de l'existence d'un lien entre *x* et *y* et l'ensemble des voisins partagés de ces nœuds, qui est calculé comme :

$$I(e_{x,y}; z) = \frac{1}{|\Gamma(z)|(|\Gamma(z)| - 1)} \sum_{u,v \in |\Gamma(z)|; u \neq v} (I(e_{u,v}) - I(e_{u,v}|z))$$

Enfin, l'auto-information conditionnelle de x et *y* étant connectés est calculée comme suit :

$$I(e_{x,y}|z) = -log_2 \frac{\left|\{e_{x,y}: x, y \in \Gamma(z), e_{x,y} \in E\}\right|}{\frac{1}{2}|\Gamma(z)|(|\Gamma(z)| - 1)}$$

- Les valeurs s_{xy}^{MI} peuvent être normalisées pour varier dans l'intervalle [0, 1] pour rendre la comparaison équitable avec les autres méthodes de prédiction de liens.

- La complexité de cette méthode est plus élevée que les autres méthodes. Elle est de ***O(nk⁶)*** et elle est donc coûteuse en termes de temps de calcul.

II.3.14. Local Naïve Bayes (Liu et al., 2011)**.**

Cette mesure utilise une approche de classification bayésienne naïve pour estimer la probabilité de l'existence d'un lien entre deux nœuds dans un réseau en se basant sur les informations locales des nœuds. Pour chaque paire de nœuds déconnectés *(x, y)*, l'idée est de calculer leur probabilité de connexion sur la base de la condition : *x* et *y* peuvent avoir un groupe de voisins communs, dont chacun a un couple de probabilités conditionnelles correspondant à encourageant et entravant les connexions de ses deux voisins Cette méthode estime la similarité de deux nœuds comme suit :

$$S_{xy}^{LNB} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} f(z) \log(oR_z) \qquad \text{(II-15)}$$

Où ***o*** est une constante pour le réseau.

$$o = \frac{\frac{1}{2}|V|(|V| - 1)}{|E|} - 1$$

Rz est le rôle ou l'influence du nœud, qui est calculé comme suit :

$$Rz = \frac{2\left|e_{x,y}: x, y \in \Gamma(z), e_{x,y} \in E\right| + 1}{2\left|e_{x,y}: x, y \in \Gamma(z), e_{x,y} \notin E\right| + 1}$$

et ***f(z)*** est une fonction qui mesure l'influence du nœud :

- ***f(z) = 1*** pour la méthode des voisins communs ***(LNB-CN).***
- ***f(z) = 1/log |Γ(z)|*** pour la méthode d'index d'Adamic-Adar ***(LNB-AA).***
- ***f(z) = 1/ |Γ(z)|*** pour la méthode d'allocation des ressources ***(LNB-RA).***

- La mesure LNB ne produit pas de scores normalisés compris entre 0 et 1. Il est donc recommandé de les normaliser si nécessaire pour les comparer avec d'autres méthodes de prédiction de liens, en utilisant le même seuil de comparaison.

- La complexité de calcul de cette méthode est : ***O(nk³).***

II.3.15. CAR-Based Indices (Cannistraci et al., 2013).

Cette mesure suppose qu'il est plus probable que deux nœuds soient liés si leurs premiers voisins communs sont membres d'un ensemble de voisins communs fortement connexe. Cette hypothèse permet de donner plus d'importance aux nœuds interconnectés avec autres voisins.

- Une version de CAR est basée sur des voisins communs est définie comme suit *:*

$$S_{xy}^{CAR-CN} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} 1 + \frac{|\Gamma(x) \cap \Gamma(y) \cap \Gamma(z)|}{2} \quad \text{(II-16)}$$

- Une variante de CAR est basée sur l'allocation des ressources.

$$S_{xy}^{CAR-RA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{|\Gamma(x) \cap \Gamma(y) \cap \Gamma(z)|}{|\Gamma(z)|} \quad \text{(II-17)}$$

- Une autre variante de CAR est basée sur l'indice d'Adamic Adar.

$$S_{xy}^{CAR-AA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{|\Gamma(x) \cap \Gamma(y) \cap \Gamma(z)|}{log|\Gamma(z)|} \quad \text{(II-18)}$$

- Les valeurs S_{xy}^{CAR-CN} , S_{xy}^{CAR-RA} et S_{xy}^{CAR-AA} ne sont pas normalisées. Il est toujours conseillé de les normaliser pour avoir une vue claire par rapport d'autre méthode de prédiction de liens.

- La complexité pour les trois méthodes est la même, elle est : ***O(nk⁴).***

II.3.16. Functional Similarity Weight (Chua et al., 2006)

Cette méthode est dérivée de l'indice de Sørensen en considérant que la probabilité qu'un nœud *x* interagisse avec un nœud *y* est indépendante de la probabilité que le nœud *y* interagisse avec le nœud *x* dans les réseaux orientés. Cependant, ce score peut également être appliqué aux réseaux non orientés comme suit :

$$S_{xy}^{FSW} = \left(\frac{2|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) - \Gamma(y)| + 2|\Gamma(x) \cap \Gamma(y)| + \lambda}\right)^2 \quad \text{(II-19)}$$

où λ est calculé comme suit :

$$\lambda = \max(0, \langle k \rangle - (|\Gamma(x) - \Gamma(y)| + |\Gamma(x) \cap \Gamma(y)|)$$

- Les valeurs obtenues par cette méthode ne sont pas normalisées. Il est donc préférable de les normaliser avant de les utiliser dans une comparaison entre les méthodes de prédiction de liens.

- La complexité de calcul de cette méthode est ***O(nk³).***

II.3.17. Local InTeracting Score (Liu et al., 2008)

Le score de cette méthode est défini sur la base de l'observation que si deux protéines ont de nombreux voisins communs, alors ces deux protéines peuvent interagir l'un avec l'autre. Elle est définie comme suit :

$$S_{xy}^{LIT} = \frac{\sum_{u\in\Gamma(x)\cap\Gamma(y)} S_{z,x}^{LIT}(t-1) + \sum_{v\in\Gamma(x)\cap\Gamma(y)} S_{z,y}^{LIT}(t-1)}{\sum_{u\in\Gamma(x)} S_{z,x}^{LIT}(t-1) + \sum_{v\in\Gamma(y)} S_{z,y}^{LIT}(t-1) + \lambda(x) + \lambda(y)} \qquad \text{(II-20)}$$

où $\lambda(x)$ est calculé comme suit :

$$\lambda(x) = \max(0, \sum_{u\in V}\sum_{v\in\Gamma(u)} S_{u,x}^{LIT}(t)/|V| - \sum_{z\in\Gamma(x)\cap\Gamma(y)} S_{z,x}^{LIT}(t-1))$$

- Initialement, les poids ont comme attribut $S_{x,y}(0) = 1$ pour les paires de nœuds connectés et $S_{x,y}(0) = 0$ pour le reste des paires.

- Quant à la normalisation des valeurs, cela dépend du contexte et de l'objectif de l'application de la méthode. Dans certains cas, il peut être approprié de normaliser les scores pour qu'ils soient compris entre 0 et 1.

- Cette mesure est une fonction récursive. La complexité de calcul de chaque itération est ***O(vk³).*** Si le nombre d'itérations est ***l***, la complexité de calcul finale est : ***O(lnk³).***

II.3.18. Sam Similarity (Samad et al., 2019)

Cette méthode est basée sur la similitude de *x* à *y* ainsi que la similitude de *y* à *x*. Il divise le calcul de la similarité en deux parties. Tout d'abord, on détermine à quel point *x* est similaire à *y*, ensuite, on calcule à quel point *y* est similaire à *x*. et enfin, on prend la moyenne des deux résultats. Cette mesure est donnée par :

$$S_{xy}^{Sam} = \frac{\frac{|\Gamma(x)\cap\Gamma(y)|}{\Gamma(x)} + \frac{|\Gamma(x)\cap\Gamma(y)|}{\Gamma(y)}}{2} \qquad \text{(II-21)}$$

- Les valeurs obtenues par cette méthode sont la moyenne de deux parties normalisées, Elles sont donc normalisées.

- En termes de complexité, la mesure de similarité ***Sam*** peut être considérée comme relativement simple et efficace, car elle combine des mesures de similarité relativement simples. La complexité est : ***O(nk³).***

II.3.19. Clustering Coefficient for Link Prediction (Wu et al., 2016b) .

Cette mesure emploie plus d'informations sur la structure locale des liens tel que les triangles. Elle partage la même idée que l'indice CAR, mais coûte moins de temps de calcul. L'idée clé de la méthode est d'exploiter la valeur des liens entre autres voisins de voisins communs. Elle est basée sur l'intuition que les nœuds qui ont un coefficient de clustering élevé sont plus susceptibles de former de nouvelles connexions à l'avenir. Elle est définie comme suit :

$$S_{xy}^{CCLP} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{t_z}{k(z)(k(z)-1)/2} \qquad \text{(II-22)}$$

tel que tz est le nombre de triangles passant par le nœud *z*.

- Les valeurs obtenues par cette méthode ne sont pas normalisées. Il est fréquent que les mesures de similarité soient normalisées afin d'assurer une comparaison équitable entre différents ensembles de données et méthodes.

- La complexité moyenne de l'attribution des scores finaux de cette similarité est $O(n^2k^3)$, puisqu'elle nécessite de calculer le coefficient de regroupement pour chaque nœud.

II.3.20. Triangle Structure Index (Bai et al., 2018)

Cette méthode nous propose un nouvel indice de similarité, en combinant la structure en triangle susmentionnée et l'idée de l'indice ***RA***. Elle est définie comme suit :

$$S_{xy}^{TRA} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1 + \Delta(x,y;z)/2}{k(z)} \qquad \text{(II-23)}$$

où $\Delta(x,y;z)$ est le nombre de triangles TRA formés par *x, y* et *z*, lequel est :

$$\Delta(x,y;z) = \Delta(x,z) + \Delta(y,z)$$

- Les résultats finaux sont des valeurs entre 0 et 1, c'est-à-dire les valeurs sont normalisées.

- La complexité de cette méthode est $O(n^2k^3)$, puisqu'elle nécessite de calculer le les triangles pour chaque nœud.

II.3.21. Common neighbor plus preferential attachment index (Zeng, 2016)

Cette mesure vise estimer la probabilité de l'existence d'un lien entre deux nœuds en basant sur la combinaison des deux méthodes Common Neighbors et preferential attachment. Elle est définie comme suit :

$$S_{xy}^{CN+PA} = |\Gamma(x) \cap \Gamma(y)| + \alpha \frac{|\Gamma(x)| * |\Gamma(y)|}{\frac{\sum_{z \in V} |\Gamma(z)|}{|V|}} \qquad \text{(II-24)}$$

où $\boldsymbol{\alpha}$ est un paramètre, (eg. $\boldsymbol{\alpha = 0.01}$), $\frac{\sum_{z \in V} |\Gamma(z)|}{|V|}$ est égal au degré moyen ⟨k⟩ du réseau

- Les scores $\boldsymbol{S_{xy}^{CN+PA}}$ ne sont pas des valeurs normalisées.
- La complexité de cette méthode est ***O(nk³).***

II.3.22. Common Neighbor to Degree (Mumin et al., 2022)

Considérons deux nœuds non connectés *x* et *y*. En supposant qu'ils ont des voisins communs entre eux, ils sont capables de présenter et de propager certaines informations à travers ces voisins dans leur interaction. Les totaux des ressources allouées à la paire de nœuds cibles sont estimés à l'aide de l'équation suivante :

$$\boldsymbol{S_{xy}^{CN2D}} = |\boldsymbol{\Gamma(x) \cap \Gamma(y)}| + \beta \left(\frac{\mathbf{1}}{\boldsymbol{max(k_x, k_y)}} \sum_{\boldsymbol{z \in \Gamma(x) \cap \Gamma(y)}} |\boldsymbol{\Gamma(z)}| \right) \quad \text{(II-25)}$$

avec un paramètre β compris entre 0 et 1, (eg. β=0.01)

- les valeurs obtenues par cette méthode ne sont pas normalisées.
- la complexité de cette méthode est : ***O(nk³).***

II.3.23. Adaptive degree penalization for link prediction (Martínez et al., 2016a)

Cette mesure s'appuie sur la constatation que le degré de pénalisation qui obtient mieux les résultats de la prédiction de lien, peut être estimé en considérant le coefficient de regroupement du réseau. La technique de prédiction s'adapte automatiquement au réseau. Elle est définie comme suit :

$$\boldsymbol{S_{xy}^{ADP}} = \sum_{\boldsymbol{z \in \Gamma(x) \cap \Gamma(y)}} |\boldsymbol{\Gamma(z)}|^{-c\beta} \quad \text{(II-26)}$$

où $\boldsymbol{C}$ est le coefficient de regroupement moyen du réseau et $\boldsymbol{\beta}$ est une constante (eg. 2.5).

- Les scores obtenus sont normalisées.
- la complexité de cette méthode est ***O(nk³).***

II.3.24. Node and Link Clustering coefficient (Wu et al., 2016a)

Cette méthode de similarité est basée sur la caractéristique topologique de base d'un réseau appelé ***"Clustering Coefficient***". Les coefficients de regroupement des nœuds et des liens sont incorporés pour calculer le score de similarité.

$$\boldsymbol{S_{xy}^{NLC}} = \sum_{\boldsymbol{z \in \Gamma(x) \cap \Gamma(y)}} \frac{\boldsymbol{S_{xz}^{CN}}}{\boldsymbol{k_z - 1}} * \boldsymbol{C(z)} + \frac{\boldsymbol{S_{yz}^{CN}}}{\boldsymbol{k_z - 1}} * \boldsymbol{C(z)} \quad \text{(II-27)}$$

- Les scores obtenus ne sont pas normalisées.
- la complexité de cette méthode est ***O(nk⁴).***

II.3.25. Local Affinity Structure Index : (Sun et al., 2017)

Cette méthode montre la relation d'affinité entre une paire de nœuds et leurs voisins communs. L'hypothèse utilisée ici est que plus l'affinité entre deux nœuds et le nombre de leurs voisins communs sont élevés, plus la probabilité de créer un lien entre eux est élevée. Elle est définie comme suit :

$$S_{xy}^{\mathbf{LAS}} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)|} + \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(y)|} \qquad \text{(II-28)}$$

- Le score S_{xy}^{LAS} est normalisé, les valeurs sont comprises entre 0 et 1.
- La complexité de cette méthode est : ***O(nk³).***

II.3.26. Local Neighbors Link Index :(Yang et al., 2015)

Cette mesure est motivée par la cohésion entre les voisins communs et les nœuds prédits. Les caractéristiques attributaires et topologiques sont examinées dans la proposition de cette méthode. Elle est définie par :

$$S_{xy}^{\mathrm{LNL}} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{\sum_{u \in \Gamma(x) \cup x} \delta(z,u) + \sum_{v \in \Gamma(y) \cup y} \delta(z,v)}{|\Gamma(z)|} \qquad \text{(II-29)}$$

tel que $\delta(u,v)$ représente s'il existe un lien entre u et v et il est défini comme suit :

$$\delta(u,v) = \begin{cases} 0 & e_{u,v} \notin E \\ 1 & e_{u,v} \in E \end{cases}$$

- les résultats obtenus ne sont pas normalisés.
- la complexité de cette méthode est : ***O(nk⁴).***

II.4. Méthodes de similarité globale

Les chemins entre deux nœuds peuvent également être utilisés pour calculer les similarités de paires de nœuds. Par exemple, la distance du chemin le plus court est basée sur le fait que plus la distance entre deux nœuds est courte, plus la probabilité que ces derniers soient connectes est élevée.

Les approches basées sur les chemins utilisent l'information topologique globale du réseau. Elles donnent des résultats meilleurs que ceux donnés par les méthodes basées sur les nœuds voisins. Par contre le calcul d'un indice global est couteux en termes de complexité, plus particulièrement pour les réseaux de grandes échelles où l'information topologique globale n'est pas toujours accessible (Lü and Zhou, 2011). Voici quelques méthodes parmi les plus populaires de cette classe :

II.4.1. Negated Shortest Path (Liben-Nowell, 2005)

Le plus court chemin est la base de la mesure de cette méthode. Celle-ci est basée sur la plus courte distance ou chemin négative entre deux nœuds. Les chemins les plus courts peuvent être efficacement calculés avec l'algorithme de Dijkstra. Cette mesure est définie comme suit :

$$S_{xy}^{NSP} = = -|shortest\ path_{x,y}|. \quad \text{(II-30)}$$

- La normalisation est nécessaire pour les valeurs obtenues par cette méthode. La normalisation peut être appliquée pour rendre les valeurs entre 0 et 1.

- La complexité de cette méthode est ***O(e.n.log(n))*** tel que *e* est le nombre de liens dans le graphe.

II.4.2. The Katz Index (Katz, 1953)

Cette mesure compte tous les chemins de différentes longueurs qui relient les pairs de nœuds, en utilisant une pondération en fonction de la longueur d'un chemin, de sorte que les chemins courts ont un poids élevé par rapport aux chemins longs auxquels on affecte des poids faibles.

$$S_{xy}^{KI} = \sum_{l=1}^{\infty} \beta^{l} \cdot |paths_{x,y}^{l}| \quad \text{(II-31)}$$

où $paths_{x,y}^{l}$ est l'ensemble de tous les chemins de longueur ***l*** reliant **x** et **y**, et **β** est un paramètre libre contrôlant les poids des chemins **(0 < β < 1).**

- Les valeurs de l'indice de Katz ne sont pas normalisées, mais peuvent être transformées en valeurs dans la plage [0,1] en utilisant une formule de normalisation.
- Dans cette méthode, la complexité de calcul est de ***O(n³),***

II.4.3. The Global Leicht-Holme-Newman Index (Leicht et al., 2006)

C'est une variante de Katz Index, basée sur l'idée que deux nœuds sont similaires si leurs voisins sont similaires. On compte tous les chemins entre deux nœuds, mais on les pondère par le nombre attendu de ces chemins dans un graphe aléatoire avec la même distribution de degré. Elle est définie comme suit :

$$S_{xy}^{GLHNI} = 2m\lambda_1 D^{-1}(I - \frac{\varphi A}{\lambda_1})^{-1} D^{-1} \quad \text{(II-32)}$$

où *m* est le nombre de liens dans le réseau, λ_1 est la plus grande valeur propre de matrice de contiguïté ***A***, ***D*** est une matrice diagonale de degré de la forme $diag\{k_1, \ldots, k_n\}$, φ est un paramètre libre $(0 < \varphi < 1)$. Le choix de φ dépend du réseau étudié : un φ plus petit attribue plus de poids sur des chemins plus courts.

- les valeurs obtenues par cette méthode ne sont pas normalisées.
- la complexité de cette méthode est $O(cn^2k)$, tel que *c* est le nombre d'itérations nécessaires.

II.4.4. Random Walks (Liu and Lü, 2010),

La marche aléatoire est une chaîne de Markov qui décrit une séquence de nœuds visités par un marcheur aléatoire. Ce processus peut être décrit par la probabilité de transition qu'un marcheur aléatoire à partir du nœud *x* se situe au nœud *y* après *t* étapes. La probabilité d'atteindre chaque nœud peut être approximée de manière itérative par :

$$S_{x,y}^{RW}(t) = P^T S_{x,y}^{RW}(t-1) \quad \text{(II-33)}$$

Initialement, P est une matrice des probabilités de transition, $P_{xy} = {a_{xy}}/{k_x}$ $a_{xy} = 1$ pour les paires de nœuds connectés et $a_{xy} = 0$ pour le reste des paires, et où $\boldsymbol{S_x(0)}$ est un vecteur $N \times 1$ avec le x[ième] élément égal à 1 et les autres à 0.

- Les valeurs de cette mesure sont normalisées, puisque les résultats sont des valeurs de probabilité.
- La complexité de cette méthode est $\boldsymbol{O(cn^2k)}$ où c est le nombre d'itérations jusqu'à la convergence.

II.4.5. Random Walks with Restart (Tong et al., 2006).

Cette méthode estime itérativement la proximité entre deux nœuds. L'idée de base des marches aléatoires est qu'un nœud est important s'il est connecté à d'autres nœuds importants. A partir d'un nœud, le marcheur est confronté à deux choix à chaque étape : soit se déplacer à un voisin choisi au hasard avec une probabilité $\boldsymbol{\alpha}$, ou revenir au nœud de départ avec la probabilité $\boldsymbol{(1-\alpha)}$ elle est définie comme suit :

$$\boldsymbol{S_{xy}^{RWR} = \vec{P}_y^x + \vec{P}_x^y} \quad \text{(II-34)}$$

Tel que :

$$\boldsymbol{\vec{P}^x\,(t) = \alpha M^T \vec{P}^x\,(t-1) + (1-\alpha)\vec{s^x}}$$

où M est la matrice de probabilité de transition, $M_{xy} = {1}/{k_x}$ si $A_{xy} = 1$ et 0 pour les autres cas , $\vec{s^x}$ est un vecteur de taille |V| , $\vec{s^x[\imath]} = 1\ si\ i = x, et\ 0\ si\ i \neq x$.

- Les valeurs obtenues par cette méthode ne sont pas normalisées.
- La complexité de cette méthode est : $\boldsymbol{O(nv^2k)}$.

II.4.6. Flow Propagation (Vanunu and Sharan, 2008)

C'est une méthode itérative basé sur la propagation. Elle fonctionne plus rapidement et est garantie de converger vers la solution du système. Cette méthode propose d'appliquer le RWR en remplacement de la matrice d'adjacence avec la matrice Laplacienne normalisée, qui peut être calculée comme suit :

$$\boldsymbol{S_{xy}^{FP} = \vec{P}_y^x + \vec{P}_x^y} \quad \text{(II-35)}$$

tel que :

$$\boldsymbol{\vec{P}^x\,(t) = \alpha W' \vec{P}^x\,(t-1) + (1-\alpha)\vec{s^x}}$$

où $\boldsymbol{D}$ une matrice diagonale telle que $\boldsymbol{D\ (i,\ i)}$ est la somme de la ligne i de la matrice d'adjacence $\boldsymbol{W}$, et $\boldsymbol{W'_{i,j} = W_{i,j}/\sqrt{D(i,i)D(j,j)}}$.

- Les valeurs obtenues par cette méthode ne sont pas normalisées.
- La complexité de cette méthode est : $\boldsymbol{O(cn^2k)}$.

II.4.7. Maximal Entropy Random Walk (Burda et al., 2009).

Les nœuds ont tendance à être liés aux nœuds centraux en réseaux structurés. La méthode de marche aléatoire à entropie maximale intègre la centralité des nœuds afin de

modéliser ce comportement. Cette méthode vise à maximiser le taux d'entropie. Elle est définie comme suit :

$$S_{xy}^{MERW} = \lim_{t\to\infty} \frac{S_{xy}^{t}}{t} \qquad \text{(II-36)}$$

Où S_t est l'entropie de Shannon de l'ensemble des trajectoires de longueur t commençant au nœud i :

$$S_{xy}^{t} = -\sum_{path_{x,y}^{l} \in paths^{l}} p(path_{x,y}^{l}) ln\, p(path_{x,y}^{l})$$

où $p(path_{x,y}^{l}) = M_{x,h}M_{h,i} \dots M_{i,j}$. Afin de maximiser l'entropie, chaque élément de la matrice de transition est calculé par :

$$M_{i,j} = \frac{A_{i,j}}{\lambda} \frac{\psi_i}{\psi_j}$$

où λ est la plus grande valeur propre de la matrice d'adjacence et ψ est la valeur normalisée du vecteur propre par rapport à λ vérifiant $\sum_{x\in V} \psi_x^2 = 1$.

- Les valeurs obtenues par cette méthode sont normalisées.
- La complexité de cette méthode est $O(cn^2k)$.

II.4.8. SimRank (G.Jeh and Widom, 2002) .

C'est une mesure de similarité qui suppose que deux nœuds sont considérés comme similaires s'ils sont référencés par des nœuds similaires. Elle est calculée récursivement comme suit :

$$S_{xy}^{SR} = \begin{cases} 1 & if \quad x = y \\ \gamma . \frac{\sum_{a\in\Gamma(x)} \sum_{b\in\Gamma(y)} S_{ab}^{SR}}{|\Gamma(x)|.|\Gamma(y)|} & sinon \end{cases} \qquad \text{(II-37)}$$

γ est un paramètre qui varie entre [0,1], c'est le facteur de décroissance.

- En ce qui concerne la normalisation, les valeurs de similarité SimRank sont normalisées.

- La complexité de cette méthode dépend du nombre d'itérations. Elle est égale à $O(n^2k^{2l+2})$ tel que l est le nombre d'itérations.

II.4.9. Pseudoinverse of the Laplacian Matrix (Spielman, 2012).

Cette méthode est basée sur La matrice Laplacienne du graphe qui fournit une représentation alternative de celui-ci **et** qui peut être considérée comme une matrice de similarité. La similarité peut être calculée à l'aide d'une mesure basée sur le produit interne telle que le cosinus comme suit :

$$S_{xy}^{PLM} = \frac{L_{x,y}^{+}}{\sqrt{L_{x,x}^{+} L_{y,y}^{+}}} \quad \text{(II-38)}$$

où $L = D - A$, D est une matrice diagonale de taille $|V|$ avec chaque élément $D_{i,i} = |\Gamma_i|$ et A est la matrice d'adjacence du graphe.

$$L^{+} = \left(L - \frac{ee^{T}}{n}\right)^{-1} + \frac{ee^{T}}{n}$$

où e est un vecteur composé de *1*, n est le nombre des nœuds.

- La matrice Laplacienne est construite à partir des degrés des nœuds du graphe, ce qui peut conduire à des valeurs non normalisées.
- La complexité de cette méthode globale est $O(n^3)$.

II.4.10. Average Commute Time (Liu and Lü, 2010)

Cette méthode suppose que deux nœuds sont considérés comme similaires s'ils ont un petit temps de déplacement moyen, Le poids des chemins est généralement déterminé par la similarité structurale entre les nœuds. Ainsi, la similarité entre les nœuds *x* et *y* peut être défini comme suit :

$$S_{xy}^{ACT} = \frac{1}{L_{x,x}^{+} + L_{y,y}^{+} - 2L_{x,y}^{+}} \quad \text{(II-39)}$$

Où $L = D - A$, D est une matrice diagonale de taille $|V|$ avec chaque élément $D_{i,i} = |\Gamma_i|$ et A est la matrice d'adjacence du graphe.

$$L^{+} = \left(L - \frac{ee^{T}}{n}\right)^{-1} + \frac{ee^{T}}{n}$$

- Les valeurs de cette méthode sont normalisées.
- La complexité de cette méthode est la même que nous avons obtenue pour la méthode de Laplacienne, elle est de $O(n^3)$.

II.4.11. Random Forest Kernel Index (Chebotarev and Shamis, 2006).

Cette méthode utilise l'algorithme Random Forest, qui est un ensemble d'arbres de décision, pour extraire les caractéristiques discriminantes des paires de nœuds. Elle est définie comme suit :

$$S_{xy}^{\mathrm{RFK}} = (I + L)^{-1} \quad \text{(II-40)}$$

Où $L = D - A$, D est une matrice diagonale de taille $|V|$ avec chaque élément $D_{i,i} = |\Gamma_i|$ et A est la matrice d'adjacence du graphe et *I* est une matrice d'identité

- Les valeurs de cette méthode sont normalisées.
- La complexité de cette méthode est $O(n^3)$.

II.4.12. Blondel Index (Blondel et al., 2004; Martínez et al., 2016b)

Cette méthode est initialement proposée dans le domaine de la détection de communautés dans les réseaux pour mesurer la similarité entre groupes de nœuds dans le graphe. Cependant, elle peut être adaptée pour fonctionner pour deux nœuds. Cette mesure est calculée itérativement comme suit :

$$S_{xy}^{\mathrm{BI}}(t) = \frac{AS(t-1)A^T + A^T S(t-1)A}{\|AS(t-1)A^T + A^T S(t-1)A\|_F} \quad \text{(II-41)}$$

où $S(0) = I$ et $\|M\|_F$ est la matrice norme de Frobenius. Elle est calculée comme suit :

$$\|M_{mxn}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(M_{i,j})^2}$$

- Il est important de noter que les valeurs obtenues par cette méthode ne sont pas normalisées.
- La complexité de cette méthode est : $O(cn^2k)$.

II.4.13. L$^+$ directly Index . (Fouss et al., 2007)

Cette méthode fournit une mesure directe de la similarité, car ses éléments sont les produits internes de vecteurs d'un espace euclidien, qui préserve le temps de trajet moyen entre les nœuds. Elle est définie comme suit :

$$S_{xy}^{\mathrm{LDI}} = L^+ \quad \text{(II-42)}$$

où $L = D - A$, D est une matrice diagonale de taille $|V|$ avec chaque élément $D_{i,i} = |\Gamma_i|$ et A est la matrice d'adjacence du graphe.

$$L^+ = \left(L - \frac{ee^T}{n}\right)^{-1} + \frac{ee^T}{n}$$

- Les valeurs de cette méthode sont normalisées.
- La complexité de cette méthode est : $O(n^3)$.

II.4.14. Newton's Gravitational Law index (Ashraf et al., 2018)

Cette approche est inspirée de la loi de la gravitation universelle de Newton, qui stipule que la force exercée entre deux masses est proportionnelle au produit de ces masses, et inversement proportionnel au carré de la distance entre leurs centres. Elle est définie comme suit :

$$S_{xy}^{NGLI} = \frac{C_D(x) * C_D(y)}{SP(x,y)} \quad \text{(II-43)}$$

où C_D désigne le degré de centralité, SP est le plus court chemin.

- Les valeurs obtenues par cette méthode ne sont pas normalisées. Il est conseillé de normaliser les valeurs obtenues pour les comparer ou les interpréter correctement.
- La complexité de cette méthode est d'ordre de $O(n^2)$.

II.4.15. Rooted PageRank.(Liben-Nowell and Kleinberg, 2007a)

C'est une méthode utilisée pour la prédiction de liens dans les réseaux. Elle est basée sur l'algorithme de PageRank, qui mesure l'importance des nœuds dans un réseau en utilisant la structure du graphe. Cette méthode attribue des scores aux nœuds du réseau en fonction de leur probabilité de se connecter à un nœud cible spécifique. Elle est définie comme suit :

$$S_{xy}^{RPR} = (1-\gamma)(I-\gamma P_{x,y})^{-1} \quad \text{(II-44)}$$

où $P_{x,y} = D^{-1} A$, D est une matrice diagonale de taille $|V|$ avec chaque élément $D_{i,j} = 0$, $D_{i,i} = |\Gamma_i|$ et A est la matrice d'adjacence du graphe.

De plus, il existe un facteur γ qui spécifie à quel point l'algorithme est susceptible de visiter les voisins du nœud plutôt que de recommencer à partir de zéro

- Les valeurs obtenues par cette méthode ne sont pas normalisées.
- Ce qui concerne la complexité est d'ordre de $O(n^2k)$.

II.5. Les méthodes de similarité Quasi-Locales

Les approches quasi-locales utilisent des informations topologiques supplémentaires. Ils calculent le score en fonction des nœuds avec une distance de chemin ne dépassant pas une longueur de deux, ce qui est similaire à l'indice des approches locales. Parmi les indices quasi-locaux, qui incluent l'indice de chemin local, les plus connues actuelles pour la prédiction de liens dans les réseaux complexes :

II.5.1. Local Path Index. (Lu et al., 2009) .

Cette méthode prend en compte les chemins locaux plus courts. Elle considère des chemins de longueur trois et est définie comme suit :

$$S_{xy}^{LPI} = (A)^2 + \beta(A)^3 \quad \text{(II-45)}$$

Où A est la matrice d'adjacence du graphe, $\beta < 1$ est une petite valeur afin que les chemins plus courts obtiennent plus de poids $(exe : \beta=0.01)$.

- Les valeurs obtenues par cette méthode ne sont pas normalisées.
- La complexité est de l'ordre de $O(n^2k)$.

II.5.2. Local Random Walks (Liu and Lü, 2010) .

Les marches aléatoires locales limitent le nombre d'itérations à un petit nombre fixe a priori t. La mesure basée sur les marches aléatoires locales est définie comme suit :

$$S_{xy}^{\mathrm{LRW}}(t) = \frac{k_x}{2|E|}.\pi_{xy}(t) + \frac{k_y}{2|E|}.\pi_{yx}(t) \quad \text{(II-46)}$$

où $\boldsymbol{\pi_{xy}(t)}$ est le vecteur de probabilité obtenu par la marche aléatoire à l'itération t.

$$\boldsymbol{\pi_{xy}(t) = P^T \pi_{xy}(t-1)}$$

et k_x est le dégrée de nœud x, $\boldsymbol{P}$ est la matrice de probabilité de la marche aléatoire.

- Les valeurs de cette méthode ne sont pas normalisées.
- La complexité dépend du nombre d'itérations, c'est-à-dire la longueur des marches aléatoires $\boldsymbol{l}$. Elle est de l'ordre de $\boldsymbol{O(ln^2k).}$

II.5.3. Superposed Random Walks (Liu and Lü, 2010).

Les méthodes basées sur la marche aléatoire sont trop sensibles à la topologie du réseau dans les zones éloignées. La méthode de la marche aléatoire superposée, qui est basée sur la méthode de la marche aléatoire locale, a été proposée pour résoudre ce problème en lançant continuellement le marcheur au nœud de départ. Ce comportement peut être obtenu en superposant la contribution de chaque marcheur :

$$S_{xy}^{\mathrm{SRW}}(t) = \sum_{l=1}^{t}\left(\frac{k_x}{2|E|}.\pi_{xy}(l) + \frac{k_y}{2|E|}.\pi_{yx}(l)\right) \quad \text{(II-47)}$$

-Les valeurs obtenues par cette méthode ne sont pas normalisées.
-La complexité est de l'ordre de $\boldsymbol{O(ln^2k).}$

II.5.4. Third-Order Resource Allocation Based on Common Neighbor (Zhang et al., 2014).

Cette mesure est une extension de l'allocation des ressources basée sur les interactions communes entre voisins qui prend également en considération les chemins à une distance de longueur trois. Ceci redéfinit l'allocation des ressources pour les nœuds à distance trois. Elle est calculée comme suit :

$$S_{xy}^{\mathrm{ORA-CNI}} = \sum_{Z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{k(z)} + \sum_{e_{i,j}\in E,\Gamma(i)<\Gamma(j),i\in\Gamma(x),j\in\Gamma(y)} \left(\frac{1}{k(i)} - \frac{1}{k(j)}\right) + \beta \sum_{[x,p,q,y]\in path_{x,y}^{3}} \frac{1}{k(p)k(q)} \quad \text{(II-48)}$$

où β est un facteur d'amortissement pour ajuster l'influence du terme d'allocation de ressources à trois sauts.

- les valeurs obtenues par cette méthode ne sont pas normalisées.
- La complexité pour cette méthode est de l'ordre de $\boldsymbol{O(ln^2k).}$

II.5.5. FriendLink (Papadimitriou et al., 2012).

Cette méthode est basée sur le nombre de chemins entre les nœuds x et y, comme l'indice de chemin local. Le score entre deux nœuds x et y est défini comme suit :

$$S_{xy}^{FL} = \sum_{i=1}^{l} \frac{1}{i-1} \cdot \frac{|paths_{x,y}^{i}|}{\prod_{j=2}^{i}(n-j)} \quad \text{(II-49)}$$

où n est le nombre de nœuds du réseau, l est la longueur maximale d'un chemin pris en compte entre x et y, $paths_{x,y}^{i}$ est l'ensemble de tous les chemins de longueur i de x à y

-Les valeurs des scores obtenues ne sont pas normalisées.

-La complexité de cette méthode est de l'ordre de $O(ln^2k)$.

II.5.6. Common Neighbor and Distance index (Yang and Zhang, 2016).

C'est une extension des voisins communs. Elle est basée sur deux propriétés d'un réseau complexe : le voisin commun et la distance entre deux nœuds *x* et *y*. Elle est définie comme suit :

$$S_{xy}^{CND} = \begin{cases} \frac{CN_{xy}+1}{2} & \Gamma(x) \cap \Gamma(y) \neq \emptyset \\ \frac{1}{d_{xy}} & sinon \end{cases} \quad \text{(II-50)}$$

CN_{xy} est le nombre de nœuds communs entre le nœud x et y, d_{xy} est la distance entre et x et y.

- Dans cette méthode, nous avons deux types de valeurs. Les valeurs CN_{xy} ne sont pas normalisées tandis que les valeurs obtenues pour 1/ d_{xy} sont normalisées. Donc en générale les valeurs ne sont pas normalisées.
- La complexité est ; $O(nk^3)$.

II.5.7. Common neighbor centrality index (Ahmad et al., 2020)

Cette méthode est basée sur deux propriétés vitales des nœuds, à savoir le nombre de voisins communs et leur centralité. Le voisin commun fait référence aux nœuds communs entre deux nœuds. La centralité fait référence au prestige qu'un nœud bénéficie dans un réseau. La mesure est définie comme suit :

$$S_{xy}^{CNC} = \alpha(|\Gamma(x) \cap \Gamma(y)|) + (1-\alpha).\frac{n}{d_{xy}} \quad \text{(II-51)}$$

α est un paramètre qui varie entre 0 et1, d_{xy} désigne la distance la plus courte entre *x* et *y*.

-Les valeurs sont normalisées,

-la complexité est de l'ordre de $O(nk^3)$.

II.5.8. Relative-path-based algorithm for link prediction (Li et al., 2020)

Cette méthode suppose que les chemins entre les nœuds et les voisins fournissent des caractéristiques de similarité de base. Cette méthode utilise des informations factorielles sur

les chemins entre les nœuds et voisins, en plus des chemins entre les paires de nœuds, dans le calcul de similarité pour la prédiction de lien. Elle est définie comme suit :

$$S_{xy}^{\mathrm{RP}} = \alpha \frac{S_{x,y}^{DP_2}}{\frac{1}{2}\left(S_{x,N_x^2}^{DP_2} + S_{y,N_y^2}^{DP_2}\right) + 1} + \beta \frac{S_{x,y}^{DP_3}}{\frac{1}{2}\left(S_{x,N_x^3}^{DP_3} + S_{y,N_y^3}^{DP_3}\right) + 1} \qquad \text{(II-52)}$$

où N_x^2 et N_x^3 sont des ensembles de nœuds tel que chaque nœud a au moins un chemin vers le nœud *x* et la longueur du chemin sont 2 et 3 respectivement et :

- $S_{x,N_x^2}^{DP_2} = \frac{1}{|N_x^2|} \sum_{i \in N_x^2} \sum_{z \in P_2(x,i)} \frac{1}{k_z}$
- $S_{x,N_x^3}^{DP_2} = \frac{1}{|N_x^3|} \sum_{i \in N_x^3} \sum_{z_1,z_2 \in P_2(x,i)} \frac{1}{k_{z_1}.k_{z_2}}$
- $S_{x,y}^{DP_2} = \sum_{z \in P_2(x,y)} \frac{1}{k_z}$
- $S_{x,y}^{DP_3} = \sum_{z_1,z_2 \in P_3(x,y)} \frac{1}{k_{z_1}.k_{z_2}}$.

De plus, $P_2(x,y)$ *et* $P_3(x,y)$ représentent tous les nœuds intermédiaires sur les chemins de longueur 2, et 3 respectivement entre les nœuds *x* et *y*.

- les résultats obtenus sont normalisés.
- la complexité de cette méthode est : $O(nk^4)$.

II.5.9. local major path degree (Yang et al., 2018)

Le chemin local entre deux nœuds est égal à la somme des degrés des nœuds intermédiaires (à l'exclusion des deux nœuds de départ). Le chemin local désigne les chemins en deux et trois étapes entre deux nœuds. Cette mesure est définie comme suit :

$$S_{xy}^{\mathrm{LMPD}} = \sum_{\omega=1}^{L_2} \frac{1}{(LPD_\omega)^\alpha} + \sum_{m=1}^{L_3} \frac{1}{(LPD_m)^\alpha} \qquad \text{(II-53)}$$

où $LPD_i = \sum_{j=1}^{n_i} k_j$.

- Les valeurs obtenues sont normalisées.
- La complexité de cette méthode est : $O(nk^4)$.

II.6. Comparaison entre les méthodes de similarités

Ce paragraphe classe et résume les méthodes de prédiction de liens basées sur la similarité, soit suivant leurs types (locale, globale et quasi-locale), soit suivant d'autre critères comme la complexité ou la normalisation des valeurs. La comparaison est basée sur un extrait des informations à partir de l'état de l'art sur les travaux de recherches publiés. Cette comparaison est présentée dans le tableau II.1. Suivant :

Tableau II.1: étude comparative entre les différentes les méthodes de prédiction de liens basent sur la similarité locale, globale et Quasi-locale

	Désignation	Référence	Type	Complexité	Nor
JI	Jaccard index	(Jaccard, 1901)	locale	$O(nk^3)$	Oui
SI	Salton index	(Salton and McGill, 1983).	locale	$O(nk^3)$	Oui
SO	Sorensen index	(McCune et al., 2002; Sorensen, 1948)	locale	$O(nk^3)$	Oui
HPI	Hub promoted index	(Ravasz et al., 2002)	locale	$O(nk^3)$	Oui
HDI	Hub depressed index	(Ravasz et al., 2002)	locale	$O(nk^3)$	Oui
AA	Adamic Adar index	(Adamic and Adar, 2003)	locale	$O(nk^3)$	Oui
PA	Preferential Attachment Index	(Mitzenmacher, 2004;Barabási, 1999)	locale	$O(nk^2)$	Non
LHNI	Leicht-Holme-Newman index	(Leicht et al., 2006)	locale	$O(nk^3)$	Oui
FSW	Functional Similarity Weight	(Chua et al., 2006)	locale	$O(nk^3)$	Oui
CN	Common Neighbors	(Liben-Nowell and Kleinberg, 2007a)	locale	$O(nk^3)$	Oui
LIT	Local InTeracting Score	(Liu et al., 2008)	locale	$O(lnk^3)$.	Non
RA	Resource allocation index	(Zhou et al., 2009)	locale	$O(nk^3)$	Oui
IA1	The Individual Attraction	(Dong et al., 2011)	locale	$O(nk^4)$	Non
IA2			locale	$O(nk^3)$	Non
LNB-CN	Local Naïve Bayes	(Liu et al., 2011)	locale	$O(nk^4)$	Non
LNB-AA			locale	$O(nk^4)$	Non
LNB-RA			locale	$O(nk^4)$	Non
CAR-CN	CAR-Based Indices	(Cannistraci et al., 2013)	locale	$O(nk^4)$	Non
CAR-RA			locale	$O(nk^4)$	Non
CAR-AA			locale	$O(nk^4)$	Non
MI	Mutual Information	(Tan et al., 2014)	locale	$O(nk^6)$	Non
RA-CNI	Resource Allocation Based on Common Neighbor Interactions	(Zhang et al., 2014)	locale	$O(nk^4)$	Non
CCLP	Clustering Coefficient for Link Prediction	(Wu et al., 2016b)	locale	$O(n^2k^3)$	Non
CN+PA	CN plus preferential attachment	(Zeng, 2016)	locale	$O(nk^3)$	Non
ADP	Adaptive degree penalization	(Martínez et al., 2016a)	locale	$O(nk^3)$	Oui
NLC	Node and Link Clustering	(Wu et al., 2016a)	locale	$O(nk^4)$	Non
LMPD	local major path degree	(Yang et al., 2018)	Quasi-L	$O(nk^4)$	Oui
TRA	Triangle Structure Index	(Bai et al., 2018)	locale	$O(n^2k^3)$	Non
Sam	Sam Similarity	(Samad et al., 2019)	locale	$O(nk^3)$	Oui
RP	Relative-path	(Li et al., 2020)	Quasi-L	$O(nk^4)$	Oui
CN2D	Common Neighbor to Degree	(Mumin et al., 2022)	locale	$O(nk^3)$	Non
KI	The Katz Index	(Katz, 1953)	globale	$O(n^3)$	Non
NSP	Negated Shortest Path	(Liben-Nowell, 2005)	globale	O(e.n.log(n))	Non
GLHN	The Global LHN Index	(Leicht et al., 2006)	globale	$O(cn^2k)$	Non
RW	Random Walks	(Liu and Lü, 2010)	globale	$O(cn^2k)$	Oui
RWR	Random Walks with Restart	(Tong et al., 2006)	globale	$O(cn^2k)$	Non
FP	Flow Propagation	(Vanunu and Sharan, 2008)	globale	$O(cn^2k)$	Non
MERW	Maximal Entropy Random Walk	(Burda et al., 2009)	globale	$O(cn^2k)$	Oui
SR	SimRank	(G.Jeh and Widom, 2002)	globale	$O(n^2k^{2l+2})$	Oui
PLM	Pseudoinverse of the Laplacian Matrix	(Spielman, 2012)	globale	$O(n^3)$	Oui

ACT	Average Commute Time	(Liu and Lü, 2010)	globale	$O(n^3)$	Oui
RFK	Random Forest Kernel Index	(Chebotarev and Shamis, 2006)	globale	$O(n^3)$	Oui
BI	The Blondel Index	(Blondel et al.,2004;Martínez et al.,2016b)	globale	$O(cn^2k)$	Non
LDI	L^+ directly Index	(Fouss et al., 2007)	globale	$O(n^3)$	Oui
NGL	Newton's Gravitational Law	(Ashraf et al., 2018)	globale	$O(n^2)$	Non
RP	Rooted PageRank	(Liben-Nowell and Kleinberg 2007a)	globale	$O(n^2k)$	Non
LPI	The Local Path Index.	(Lu et al., 2009)	Quasi-L	$O(n^2k)$	Non
LRW	Local Random Walks	(Liu and Lü, 2010)	Quasi-L	$O(ln^2k)$	Non
SRW	Superposed Random Walks	(Liu and Lü, 2010)	Quasi-L	$O(ln^2k)$	Non
ORA-CN	Third-Order Resource Allocation Based on CN	(Zhang et al., 2014).	Quasi-L	$O(ln^2k)$	Non
FL	FriendLink	(Papadimitriou et al., 2012)	Quasi-L	$O(ln^2k)$	Non
CND	CN and Distance index	(Yang and Zhang, 2016)	Quasi-L	$O(nk^3)$	Non
CNC	CN centrality index	(Ahmad et al., 2020)	Quasi-L	$O(nk^3)$	Oui

II.7. Conclusion

Dans ce chapitre nous avons présenté un état de l'art concernant toutes les méthodes de similarité locales, globales et Qasi-locales basées respectivement sur le voisinage des nœuds, les chemins entre deux nœuds, et enfin, un mélange entre le chemins et voisinage.

Les méthodes de prédiction de liens basées sur la similarité locale, globale et quasi-locale offrent différentes approches pour aborder le problème de la prédiction de liens dans un réseau. Chaque approche a ses avantages et ses limites, et sa performance peut varier en fonction des caractéristiques spécifiques du réseau.

Les méthodes de prédiction de liens locales se concentrent sur les caractéristiques locales des nœuds pour identifier les liens futurs, en se basant sur l'idée que la structure du réseau influence la construction des liens. Elles sont souvent simples et rapides à calculer.

Les méthodes de prédiction de liens globales prennent en compte l'ensemble du réseau pour identifier des motifs globaux et des structures émergentes. Elles peuvent fournir une vision plus complète du réseau et être plus résistantes aux perturbations locales, mais elles peuvent être plus complexes à mettre en œuvre et nécessitent des ressources de calcul plus importantes.

Les méthodes de prédiction de liens quasi-locales combinent des éléments des deux approches précédentes, en prenant en compte à la fois les caractéristiques locales des nœuds et les motifs globaux du réseau. Elles cherchent à tirer parti des avantages des deux approches pour obtenir des prédictions de liens plus précises. Cependant, elles peuvent être plus complexes à mettre en œuvre et nécessitent des compromis dans les performances en termes de rapidité et de ressources.

Il est important de noter que le choix de la méthode de prédiction de liens dépendra des caractéristiques spécifiques du réseau, de la disponibilité des données et des objectifs de l'analyse. Pour obtenir les meilleures performances de prédiction de liens dans un contexte donné, il peut être bénéfique d'explorer les différentes approches et techniques

La fin de ce chapitre résumé tous les méthodes proposées pour résoudre le problème de prédiction de liens dans un tableau synthétique. Le résumé est basé sur les caractéristiques de chaque méthode.

CHAPITRE –III-

Une nouvelle approche rapide pour la prédiction de liens

Chapitre –III-

Une nouvelle approche rapide pour la prédiction de liens

III.1. Introduction

En raison de la complexité des calculs élevée des algorithmes et de la limitation de la mémoire, les algorithmes traditionnels de prédiction de lien ne peuvent pas gérer efficacement les réseaux complexes à grande échelle. Il est nécessaire d'utiliser une autre alternative basée sur la décomposition du problème en sous-problèmes moins couteux en termes de complexité, de mémoire de stockage et de temps de calcule en évitant les calculs qui ne sont pas nécessaires.

Dans ce chapitre, nous présentons une nouvelle approche pour la prédiction de liens en utilisant la décomposition des réseaux en composantes connexes. Cette approche nous permet de diviser le nombre de liens en des liens à calculer effectivement et d'autres liens à déduire qui sont les liens entre les composantes. Ainsi, on se limite uniquement aux calculs des liens internes aux composantes, ce qui nous permet de gagner beaucoup de temps, et d'espace de mémoire. Ce gain important sera exposé et prouvé dans la suite du présent chapitre.

III.2. Description du problème et solution proposée

Le but de cette section est de définir formellement le problème de la prédiction de liens et d'étudier l'impact d'utiliser comme entrée les composantes connexes d'un réseau au lieu de l'ensemble des nœuds du réseau, sur la quantité de calcul nécessaire et par conséquent sur le temps d'exécution.

III.2.1. Définition 1 : (Réseau complexe)

Considérons un réseau complexe non orienté défini par : $G(V, E)$, où

- V est l'ensemble des nœuds
- E est l'ensemble des liens entre paires de nœuds de la forme (x, y) où $x, y \in V$.
- n est le nombre de nœuds : $n = |V|$.
- e est le nombre de liens : $e = |E|$.

Le graphe complet correspondant à G contient l'ensemble de tous les liens possibles que G peut avoir. Le nombre de tous ces liens possibles est : $\frac{n\times(n-1)}{2}$. Soit E' représentant l'ensemble des liens inexistants dans le réseau $G : E' = \{(x, y): x, y \in V, (x, y) \notin E\}$. Le nombre de liens appartenant à E' est donc : $\frac{n\times(n-1)}{2} - e$.

Les algorithmes de prédiction de liens existants, basés sur la mesure locale et les chemins, calculent les scores pour tous les liens non existants appartenant à E' car ils considèrent que tout lien non existant actuellement peut devenir un lien réel à l'avenir.

Ainsi, pour les bases de données volumineuses, le nombre de liens potentiels à évaluer peut-être très important. Par exemple, dans la base de données Twitter (Reza and Huan, 2009), nous avons 11 316 811 nœuds et 85 331 846 liens. Par conséquent, le nombre de scores à calculer pour les liens non existants est supérieur à 64 000 milliards !

Pour pallier ce problème, la solution que nous proposons est basée sur la décomposition de notre graphe en ses composantes connexes. En effet, comme il sera montré plus loin, pour toutes les approches basées sur des mesures locales ou des chemins, nous n'avons besoin de calculer que des scores entre les nœuds qui appartiennent à une même composante. Cela permet de gagner du temps d'exécution en réduisant le nombre d'opérations de calcul nécessaires.

De plus, le fait que les composantes connexes puissent être traitées indépendamment, permet d'effectuer des calculs parallèles en affectant à chaque processeur disponible un sous-ensemble de composantes.

III.2.2. Définition 2 (composantes connexes)

Considérons un réseau complexe non orienté défini par : $G(V, E)$, et un ensemble $C = \{G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_c = (V_c, E_c)\}$ tel que chaque $G_i = (V_i, E_i)$ soit un sous-graphe de G . C définit une décomposition de G en ses composantes connexes si :

- $\cup_{i=1}^{c} V_i = V \; et \cup_{i=1}^{c} E_i = E$
- $\forall i \in \{1, \ldots, c\}, \forall x, y \in V_i,$ *il existe un chemin reliant* x *et* y
- $\forall i, j \in \{1, \ldots c\},$ *tel que* $i \neq j, V_i \cap V_j = \emptyset \; et \;\; E_i \cap E_j = \emptyset$

Les notations suivantes seront utilisées :

- $\Gamma(x)$ désigne l'ensemble des voisins du nœud x, c'est-à-dire les nœuds ayant des liens directs avec x.
- $paths_{x,y}$ désigne l'ensemble de tous les chemins reliant x et y.
- $d_{x,y}$ désigne la distance entre les nœuds x et y, c'est-à-dire la longueur du chemin le plus court entre x et y.

La proposition suivante stipule que les nœuds appartenant à des composantes distinctes ne partagent aucun voisin et ne sont liés par aucun chemin.

III.2.3. Proposition 1.

Soit $C = \{G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_c = (V_c, E_c)\}$ l'ensemble des composantes connexes d'un réseau complexe $G = (V, E)$. Nous avons :

$$\forall \mathbf{i},\mathbf{j} \in \{\mathbf{1},\ldots,\mathbf{c}\} | \mathbf{i} \neq \mathbf{j}, \forall \mathbf{x} \in \boldsymbol{V_i}, \forall \mathbf{y} \in \boldsymbol{V_j} \begin{cases} \boldsymbol{\Gamma}(\mathbf{x}) \cap \boldsymbol{\Gamma}(\mathbf{y}) = \emptyset \ (|\boldsymbol{\Gamma}(\mathbf{x}) \cap \boldsymbol{\Gamma}(\mathbf{y})| = \mathbf{0}) \\ \mathbf{et} \\ \mathbf{paths_{x,y}} = \emptyset \ (\mathbf{d_{xy}} = \infty) \end{cases} \quad \text{(III-1)}$$

Preuve.

Pour tout ***x, y*** tel que ***x*** $\in V_i$, ***y*** $\in V_j$ et $i \neq j$ on a : $\Gamma(x) \subseteq V_i$ et $\Gamma(y) \subseteq V_j$ (tous les voisins de ***x*** (resp. ***y***) sont dans V_i (resp .V_j)). Alors, puisque $V_i \cap V_j = \emptyset$, il s'ensuit que $\Gamma(x) \cap \Gamma(y) = \emptyset$. Ceci conduit immédiatement au fait que $|\Gamma(x) \cap \Gamma(y)| = 0$.

Maintenant, si on suppose qu'il existe un chemin reliant ***x*** et ***y*** alors il doit y avoir un chemin entre certains x' et y' tel que $x' \in V_i$ et $y' \in V_j$ ce qui est impossible puisque $V_i \cap V_j = \emptyset$ et $E_i \cap E_j = \emptyset$. On obtient donc $paths_{x,y} = \emptyset$. Puisqu'il n'y a pas de chemin entre ***x*** et ***y***, il s'ensuit immédiatement que $d_{xy} = \infty$.

Corollaire 1.

Étant donné une mesure de similarité locale ou basée sur le chemin ***f***, nous avons :

$$\forall \boldsymbol{i},\boldsymbol{j} \in \{\mathbf{1},\ldots,\boldsymbol{c}\} \mid \boldsymbol{i} \neq \boldsymbol{j}, \forall \boldsymbol{x} \in \boldsymbol{V_i}, \forall \boldsymbol{y} \in \boldsymbol{V_j} : \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) = \mathbf{0} \quad \text{(III-2)}$$

Preuve. Découle immédiatement de la proposition 1 et des définitions des mesures de similarité locales et basées sur le chemin. Ce corollaire est important puisqu'il stipule qu'il suffit de ne calculer que les mesures de similarité entre couples de nœuds appartenant à une même composante connexe.

Notons à présent par ***Gain(G),*** le gain obtenu en utilisant la décomposition de ***G*** en ses composantes connexes. ***Gain(G)*** est le nombre de couples de nœuds ***(x,y)*** où ***x*** et ***y*** appartiennent à deux composantes différentes.

III.2.4. Proposition 2.

Soit $C = \{ G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, G_c = (V_c, E_c)\}$ l'ensemble des composantes connexes d'un réseau complexe $G = (V, E)$. On pose $|V| = n$, $|V_i| = n_i$ pour $i \in \{1,\ldots,c\}$ et $|E| = e$. Nous avons :

$$\boldsymbol{Gain(G)} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{c-1}} \sum_{\boldsymbol{j=i+1}}^{\boldsymbol{c}} (\boldsymbol{n_i} * \boldsymbol{n_j}) \quad \text{(III-3)}$$

Preuve.

Soit ***NEval(G)*** (resp. ***NEval'(G)***) le nombre de liens potentiels évalués en utilisant la décomposition (resp. sans utiliser la décomposition) alors ;

Gain(G) = NEval'(G) −NEval(G).

$$\boldsymbol{NEval(G)} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{c}} \left(\frac{\boldsymbol{n_i(n_i-1)}}{\mathbf{2}} - \boldsymbol{e_i} \right)$$

$$
\begin{aligned}
&= \frac{1}{2}\sum_{i=1}^{c} n_i(n_i - 1) - \sum_{i=1}^{c} e_i \\
&= \frac{1}{2}\sum_{i=1}^{c} (n_i^2 - n_i) - e \\
NEval'(G) \quad &= \frac{n(n-1)}{2} - e \\
&= \frac{1}{2}(n^2 - n) - e \\
&= \frac{1}{2}((n_1 + \cdots + n_c)^2 - (n_1 + \cdots + n_c)) - e \\
&= \frac{1}{2}\left(\left(n_1^2 + n_2^2 \ldots + n_c^2\right) + 2(n_1 n_2 + \cdots n_1 n_c + n_2 n_3 + \cdots + n_{c-1} n_c\right) \\
&\qquad\qquad - (n_1 + \cdots + n_c)) - e \\
&= \frac{1}{2}\left(\sum_{i=1}^{c} n_i^2 + 2\sum_{i=1}^{c-1}\sum_{j=i+1}^{c} n_i\, n_j - \sum_{i=1}^{c} n_i\right) - e \\
&= \frac{1}{2}\sum_{i=1}^{c} (n_i^2 - n_i) - e + \sum_{i=1}^{c-1}\sum_{j=i+1}^{c} n_i\, n_j \\
&= NEval(G) + \sum_{i=1}^{c-1}\sum_{j=i+1}^{c} n_i\, n_j
\end{aligned}
$$

On déduit que:

$$Gain(G) = \sum_{i=1}^{c-1}\sum_{j=i+1}^{c} (n_i * n_j)$$

Sans utiliser la décomposition, tous les indices de similarité basés sur des informations locales et des méthodes de chemin calculent une liste de scores (notés ***scores'(G)***) de tous les liens inexistants dans le graphe :

$$scores'(G) = \left\{\left(x, y, s_{xy}^{method}\right) / x, y \in Vet\ (x, y) \notin E\right\} \qquad \text{(III-4)}$$

Le nombre de liens évalués est :

$$\left|E'\right| = Gain(G) + \sum_{i=1}^{c}\left|E_i'\right| = \sum_{i=1}^{c-1}\sum_{j=i+1}^{c}\left(n_i * n_j\right) + \sum_{i=1}^{c}\left|E_i'\right| \qquad \text{(III-5)}$$

Après décomposition, cette liste de scores (notés ***scores (G)***) devient :

$$scores(G) = \bigcup_{i=1}^{c}\left\{\left(x, y, s_{xy}^{method}\right) / x, y \in V_i\ et\ (x, y) \notin E_i\right\} \qquad \text{(III-6)}$$

Le nombre de liens évalués est :

$$|E'| - Gain(G) = \sum_{i=1}^{c} |E'_i| \quad \text{(III-7)}$$

Pour tous les autres liens inexistants, la mesure de similarité est égale à 0 :

$$\forall i, j / \; i \neq j, \forall x \in V_i, \forall y \in V_j, s_{xy}^{method} = 0 \quad \text{(III-8)}$$

Considérons quelques cas particuliers :

- ***Cas d'une seule composante connexe*** *(**c = 1**)*. Dans ce cas, la décomposition n'a aucun effet et ***Gain(G) = 0***.
- ***Plusieurs composantes connexes (c >1).*** S'il y a plus d'une composante, le gain est strictement positif. On peut distinguer les deux cas particuliers suivants :
  - Cas où $n_1 \approx n_2 \approx \cdots \approx n_c \approx n_c$. Dans ce cas le gain est maximal et il est donné par :

$$\begin{aligned} Gain^{max}(G) &= \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} \left(\frac{n}{c}\right)^2 \\ &= \frac{c(c-1)}{2}\left(\frac{n}{c}\right)^2 \\ &= \frac{n^2(c-1)}{2c} \end{aligned}$$

  - Cas où $n_1 = (n - c + 1)$, $n_2 = n_3 = \ldots n_c = 1$. Ce cas correspond à la situation où $c-1$ des c composantes contient chacune un seul nœud, et une composante contient tous les n – c + 1 nœuds restants. Dans ce cas, le gain est minimal et est donné par :

$$\begin{aligned} Gain^{min}(G) &= \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} (n_i n_j) \\ &= \sum_{j=2}^{c} n_1 * 1 + \sum_{i=2}^{c-1} \sum_{j=i+1}^{c} (1 * 1) \\ &= \sum_{j=2}^{c} (n - c + 1) + \sum_{i=2}^{c-1} \sum_{j=i+1}^{c} (1) \\ &= (c-1)(n-c+1) + \frac{(c-1)(c-2)}{2} \\ &= \frac{(c-1)(2n-c)}{2} \end{aligned}$$

Exemple.

Considérons le graphe ***G (V, E)*** représenté sur la Figure 3.1-(a) avec $|V| = n = 9$ nœuds et $|E| = e = 8$ liens. ***G*** contient ***c = 3*** composantes connexes. Sans utiliser la décomposition, pour

prédire les liens futurs, nous devons calculer le score de tous les liens inexistants (28 liens) comme le montre la figure III.1-(b).

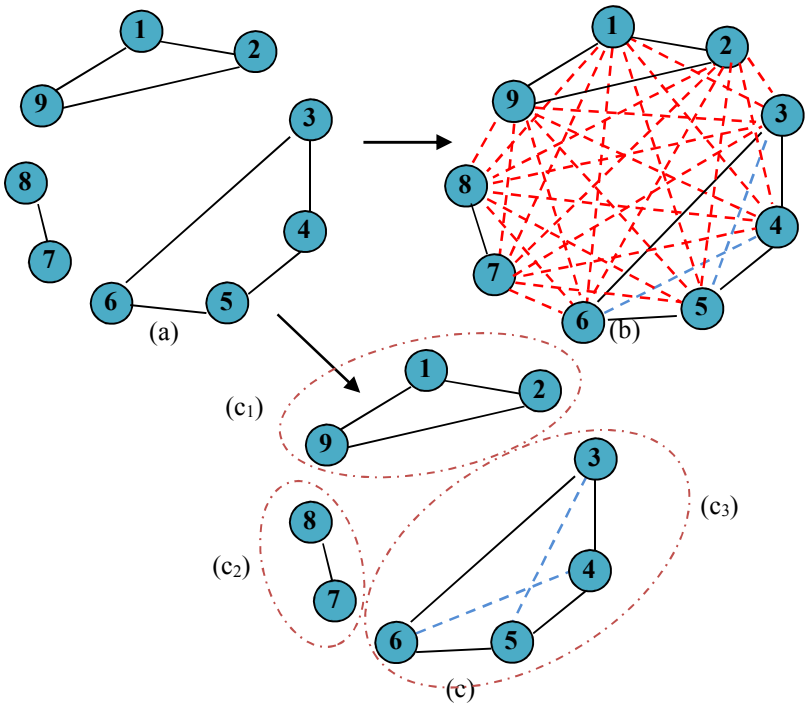


Figure III.1 : Le gain en nombre de liens calculés en utilisant la décomposition illustrée sur un réseau non orienté simple

En utilisant la décomposition de ***G*** en ses composantes connexes, au lieu de calculer les scores des liens qui n'existent pas dans le graphe complet, on ne calcule que les scores des liens qui figurent à l'intérieur des trois composantes (***c1***),(***c2***) et (***c3***) (voir Figure III.1-(c)). Nous avons 0 liens dans (***c1***), 0 liens dans (***c2***) et 2 liens dans (***c3***). Le gain obtenu est donc de 28 – 2 = 26.

III.3. Expression du gain par l'indice GINI et la variance

On suppose que ***G (V, E)*** est un graphe avec $|V| = n = 9$ nœuds. Le nombre de composantes connexes est $c = 3$.

Tableau III.1 les Différentes répartitions possibles des nœuds sur les composantes et leurs résultats de calcul de Gain, Gini et Variance

n°	n_1	n_2	n_3	Var	Var^{Norm}	Gini	$Gini^{Norm}$	Gain	$Gain^{Norm}$
1	7	1	1	8.00	1.00	0.37	0.00	15	0.00
2	6	2	1	4.67	0.58	0.49	0.42	20	0.42
3	5	1	3	2.67	0.33	0.57	0.67	23	0.67
4	5	2	2	2.00	0.25	0.59	0.75	24	0.75
5	4	4	1	2.00	0.25	0.59	0.75	24	0.75
6	4	3	2	0.67	0.08	0.64	0.92	26	0.92
7	3	3	3	0.00	0.00	0.67	1.00	27	1.00

Le tableau III.1 illustre les différents cas possibles de répartition des nœuds sur les composantes et le gain associé à chacune de ces répartitions.

On peut remarquer que le gain dépend clairement de la répartition des nœuds dans les composantes connexes : plus la répartition est homogène, c'est-à-dire où les nœuds sont plus uniformément répartis sur les composantes, plus le gain est important. En revanche, des distributions hétérogènes conduisent à des gains moins importants.

Maintenant, nous montrons que cette idée peut être capturée par deux mesures supplémentaires :

- **Le coefficient de Gini** est une mesure qui capte l'inégalité ou l'hétérogénéité d'une distribution. Il est utilisé dans différents domaines, notamment en apprentissage supervisé pour la construction d'arbres de décision (Breiman et al., 1984); (Daniya et al., 2020). Pour une décomposition $C = \{ G_1 = (V_1, E_1), G_2 = (V_2, E_2), ..., G_c = (V_c, E_c) \}$ d'un réseau $G = (V ,E)$, le coefficient de Gini est défini par la formule suivante :

$$Gini(G) = 1 - \sum_{i=1}^{c} p^2 (n_i) \qquad \text{(III-9)}$$

où $P(n_i)$ est la proportion de nœuds présents dans la composante $G_i(V_i,E_i)$; et il est déterminé comme suit :

$$p(n_i) = \frac{|V_i|}{|V|} = \frac{n_i}{n} \qquad \text{(III-10)}$$

- **La variance** est une mesure statistique de la dispersion des valeurs d'une distribution. Il est calculé comme suit :

$$Var(G) = \frac{\sum_{i=1}^{c} |n_i - m|^2}{c} \qquad \text{(III-11)}$$

où m est la valeur moyenne de la distribution, c'est-à-dire $m = {}^{n}/_{C}$.

Les trois mesures (***Gain***, ***Gini*** et ***Variance***) peuvent être normalisées pour prendre leurs valeurs en [0,1]. Pour cela, nous utilisons la formule suivante :

$$X^{norm} = \frac{X - X^{min}}{X^{max} - X^{min}} \in [0, 1] \qquad \text{(III-12)}$$

où $X \in \{ Gain, Gini, Var\}$, X_{norm}(resp. X_{max}, X_{min}) est la valeur normalisée (resp. maximale, minimale) de X. On peut vérifier que :

$$Gini^{max}(G) = 1 - \frac{1}{c} \quad \text{et} \quad Gini^{min}(G) = \frac{(c-1)(2n-c)}{n^2}$$

$$Var^{max}(G) = \frac{(c-1)(n-c)^2}{c^2} \quad \text{et} \quad Var^{min}(G) = 0$$

$$Gain^{max}(G) = \frac{n^2(c-1)}{2c} \quad \text{et} \quad Gain^{min}(G) = \frac{(c-1)(2n-c)}{2}$$

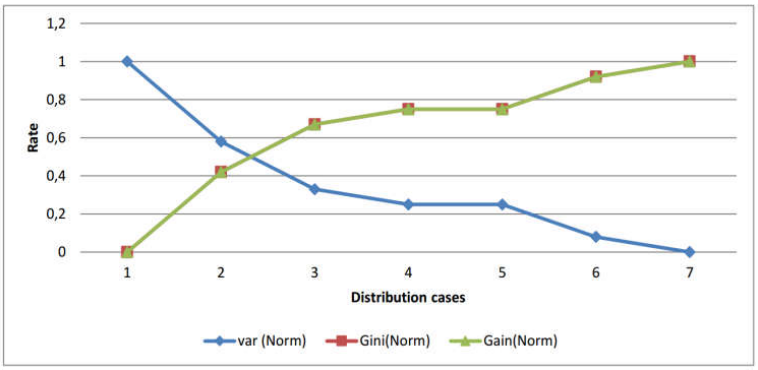


Figure III.2 : Taux de gain, de Gini et de variance en fonction de la répartition des nœuds sur les composants.

Le tableau III.1 donne les valeurs dirigées et normalisées des trois mesures (***Gain***, ***Gini*** et ***Variance***) pour les différentes distributions possibles dans notre exemple. La figure III.2 représente graphiquement les résultats obtenus.

De manière assez intéressante, nous pouvons remarquer que le gain normalisé et le coefficient de Gini normalisé coïncident ($Gain_{norm}(G) = Gini_{norm}(G)$). En revanche, la variance normalisée est inversement proportionnelle aux Gini et Gain normalisés, et plus précisément : $Var_{norm}(G) = 1 - Gain_{norm}(G)$.

III.4. L'architecture proposée

Dans cette section, nous proposons une architecture pour le problème de prédiction de liens qui est basée sur six étapes principales au lieu de quatre étapes (Ying et al., 2014), à savoir : la collecte de données, le prétraitement, la décomposition, la parallélisation, la prédiction et l'évaluation. Notre architecture est illustrée dans la Figure III.3.

- **Préparation :** Cette étape consiste à préparer les bases de données correspondant à un réseau complexe, en deux sous-ensembles : un d'apprentissage et un de test. La base de données complète est découpée selon l'un des deux cas suivants :
    - Le premier principe repose sur le temps d'évolution de la base de données (Liben-Nowell and Kleinberg, 2007b). Puisque chaque interaction a lieu à un instant particulier, on définit un intervalle d'apprentissage $[t_0, t'_0]$ qui à son tour définit le sous-graphe $G^{Tr} = (V^{Tr}, E^{Tr})$ contenant toutes les interactions réalisées entre les instants t_0, et t'_0. De même, un autre intervalle de test $[t_1, t'_1]$ définit un sous-graphe de test $G^{Ts} = (V^{Ts}, E^{Ts})$ contenant toutes les interactions effectuées entre les instants t_1, et t'_1. Ce principe est utilisé pour prédire la probabilité de liens futurs (dans les réseaux dynamiques (Kumar et al., 2020)).
    - Suivant le deuxième principe, nous divisons aléatoirement l'ensemble des liens de la base de données en deux sous-ensembles. Par exemple, 90% (Lü and Zhou, 2011); (Xiao and Zhang, 2015); (Wang et al., 2017); (Daminelli et al., 2015), 80%

(Ahmad et al., 2020); (Wu, 2018) ou 70% (Sui et al., 2013) de liens dans un sous-graphe d'entraînement $G^{Tr}= (V^{Tr}, E^{Tr})$ et 10% (20%, 30%) des liens dans un sous-ensemble de graphe de test $G^{Ts} = (V^{Ts}, E^{Ts})$ où $E = E^{Tr} \cup E^{Ts}$ et $E^{Tr} \cap E^{Ts} = \emptyset$. Ce principe est utilisé pour trouver les chaînons manquants (dans les réseaux statiques (Kumar et al., 2020)).

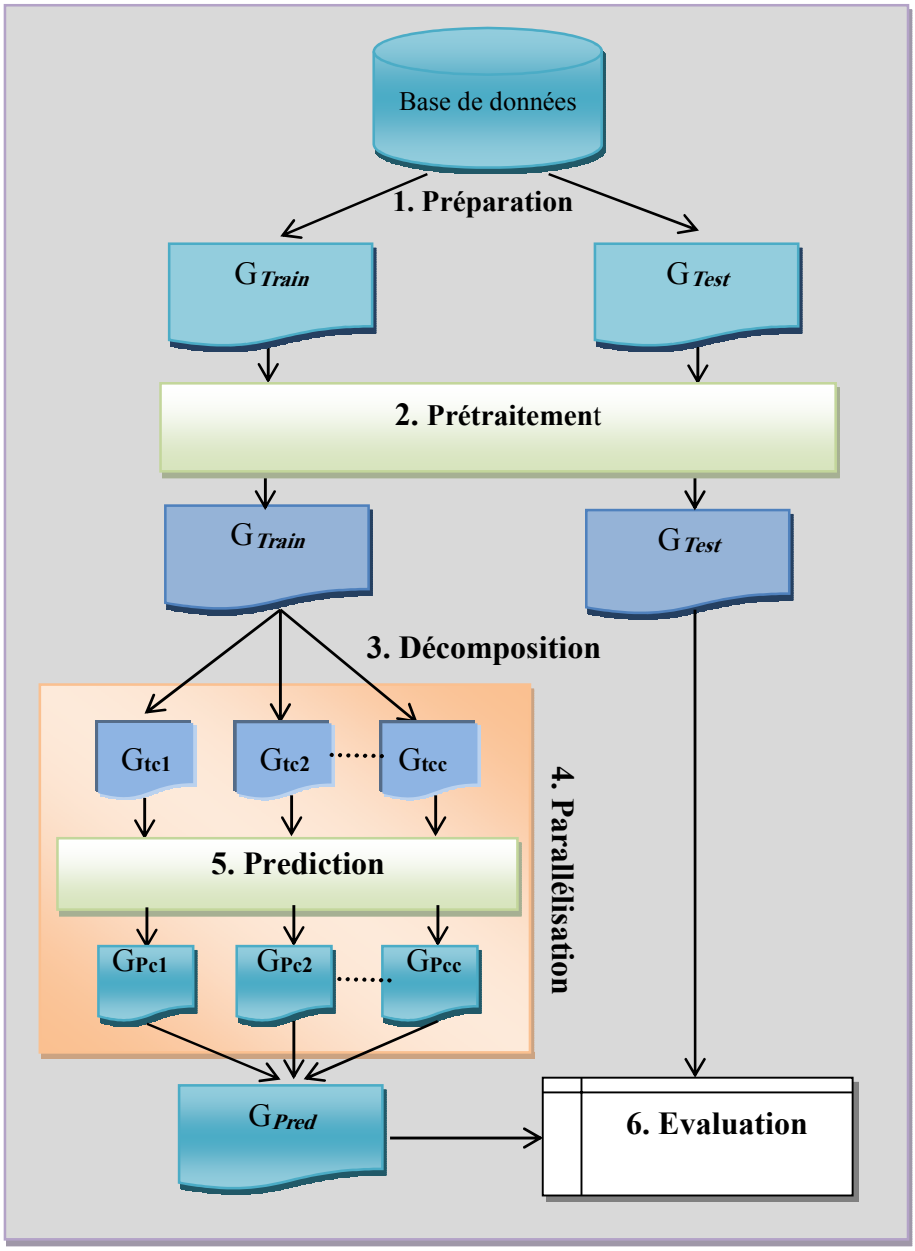


Figure III.3 : Architecture parallèle pour la prédiction de liens basée sur la décomposition en composantes connexes.

- **Prétraitement :** Cette étape consiste à filtrer certains nœuds de notre réseau par les actions suivantes :
    - Éliminer les nœuds isolés (Lu et al., 2009) puisqu'il est difficile de prédire quoi que ce soit sur ces nœuds, à partir du moment où ils ne disposent d'aucune information basée sur le réseau.

- Écarter parfois les nœuds ayant un degré très bas (Liben-Nowell and Kleinberg, 2007b); (Lu et al., 2009); (Scripps et al., 2009) , parce que nous n'avons pas suffisamment d'informations sur ces nœuds.
- Convertir un réseau orienté en un réseau non orienté avant d'appliquer les algorithmes de prédiction des liens (Shibata et al., 2012).
- Limiter l'analyse aux nœuds communs aux sous-réseaux d'apprentissage et de ***test*** (Lu et al., 2009) et éliminer les autres nœuds.

- **Décomposition**. Dans cette étape, nous proposons de décomposer le graphe d'apprentissage G = ***(V, E)*** en l'ensemble de ses composantes connexes :

$$C = \{ G_1 = (V_1,E_1), G_2 = (V_2, E_2), ..., G_c = (V_c, E_c)\}$$

- **Parallélisation.** Dans cette étape, nous proposons de paralléliser les calculs, puisque chaque composante est indépendante des autres composantes. Les composantes connexes sont distribuées sur plusieurs processeurs. Chaque processeur reçoit un ou plusieurs composantes. Nous utilisons le terme nœuds locaux pour désigner les nœuds que chaque processeur reçoit.

- **Prédiction.** Chaque composante est évaluée localement avec ses nœuds locaux, et une liste de scores partiels est établie. À ce stade, une seule mesure de similarité est utilisée. Dans la plupart des cas, nous voulons prédire seulement les nouveaux liens, c'est-à-dire les liens qui ne sont pas présents dans la composante d'apprentissage. A la fin de cette opération, tous les scores de listes locales sont fusionnés dans un score de liste globale.

- **Évaluation**. Enfin, il devient possible d'évaluer la performance des résultats des prévisions en les comparants à ceux du sous-ensemble de test. Nous utilisons différents seuils, paramètres et mesures de performance.

III.5. Les mesures de performances

Dans la prédiction de liens, nous parlons d'une **classification binaire**, où il s'agit de deviner si un lien apparaîtra dans le futur ou pas.

De nombreux modèles de classification retournent des **valeurs réelles**, qui peuvent souvent être interprétées comme la probabilité que le lien appartient à la classe positive ou à la classe négative suivant un **seuil** fixé pour séparer les négatifs des positifs.

Une matrice de confusion (Kohavi and Provost, 1998) contient des informations sur les classifications réelles et prévues effectuées par un système de classification. Les performances de ces systèmes sont généralement évaluées à l'aide des données de la matrice. Le tableau suivant montre la matrice de confusion pour un classificateur à deux classes.

Tableau III-2: Une matrice de confusion

		La Prédiction	
		Positive (lien)	Négative (pas de lien)
La réalité	Positive (lien)	True Positive(**TP**) Vrais positifs	False Negative(**FN**) Faux négatifs
	Négative (pas de lien)	False Positive(**FP**) Faux positifs	True Négative(**TN**) Vrais négatifs

Soit $G^r(V^r, E^r)$ un graphe réel, et $G^p(V^p, E^p)$ est un graphe prédit alors :

TP : Les valeurs réelles et prédites sont identiques. La valeur prédite du modèle est positive (il existe un lien), ainsi qu'une valeur positive réelle.

Si le lien $e(x,y) \in E^p$ et $e(x,y) \in E^r$

FN : Les valeurs réelles et prédites ne sont pas les mêmes. La valeur prédite du modèle est négative (pas de lien) et faussement prédite. Cependant, la valeur réelle est positive (il existe un lien). On peut parler d'une erreur.

Si le lien $e(x,y) \notin E^p$ et $e(x,y) \in E^r$

FP : Les valeurs réelles et prédites ne sont pas les mêmes. La valeur prédite du modèle est positive (il existe un lien) et faussement prédite. Cependant, la valeur réelle est négative (pas de lien). On peut parler d'une erreur.

Si le lien $e(x,y) \in E^p$ et $e(x,y) \notin E^r$

TN : Les valeurs réelles et prédites sont identiques. La valeur prédite du modèle est négative (pas de lien), avec une valeur négative réelle.

Si le lien $e(x,y) \notin E^p$ et $e(x,y) \notin E^r$

Il existe plusieurs façons d'évaluer les modèles de prédiction de liens. On peut extraire de la matrice de confusion différentes mesures de performance, comme le **rappel**, et la **précision**, qui reflètent différents aspects du modèle.

Rappel (en Anglais Recall)

Le Rappel ou la sensibilité ou le taux de vrais positifs (TPR) est la mesure des résultats positifs corrects renvoyés par rapport au nombre de résultats positifs corrects qui auraient pu être produits. Une valeur plus élevée de Rappel signifie qu'il y a moins de faux négatifs

$$\text{Rappel} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{III-13}$$

Précision (en Anglais Precision)

La précision est la mesure des résultats positifs corrects parmi tous les résultats positifs prédits

$$\text{Précision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{III-14}$$

Accuracy

L'accuracy est le rapport entre le nombre prédictions correctes, sur le nombre total de prédictions. Elle est définie comme suit :

$$\text{Accuracy} = \frac{\mathbf{TP + TN}}{\mathbf{TP + TN + FP + FN}} \quad \text{(III-15)}$$

F-Mesure

C'est la moyenne harmonique de la précision et du rappel. Elle mesure la capacité du système à donner toutes les solutions pertinentes et à refuser les autres

$$\mathbf{F - Mesure} = 2\frac{\mathbf{Précision.Recall}}{\mathbf{Précision + Recall}} \quad \text{(III-16)}$$

Courbe ROC (Receiver Operating Characteristic):

C'est la variation du taux de vrais positifs (TPR : True Positives Rate) par rapport au taux de faux positifs (FPR : False Positive Rate) à divers réglages de seuil, respectivement au axes *x* et *y*.

- TPR : fraction de vrais positifs sur le total des positifs.

$$\mathbf{TPR} = 1 - \text{FNR} = \frac{\mathbf{TP}}{\mathbf{TP + FN}} \quad \text{(III-17)}$$

- FPR : fraction de faux positifs sur le total des négatifs.

$$\mathbf{FPR} = 1 - \text{TNR} = \frac{\mathbf{FP}}{\mathbf{FP + TN}} \quad \text{(III-18)}$$

AUROC (Area Under the ROC) : La courbe ROC permet de décrire la performance d'un modèle à travers deux indicateurs : la sensitivité et la spécificité. L'aire sous cette courbe, nommée AUC ROC, mesure de façon globale la performance d'un modèle de classification.

III.6. Algorithmes

Dans cette section nous présentons et comparons, en termes d'exécution, différents scénarios pouvant être obtenus en considérant différents critères de choix : exploiter ou pas la décomposition du graphe en composantes connexes ; intégration ou pas de l'étape de décomposition dans l'algorithme et exécution séquentielle ou parallèle. Cela conduit à cinq scénarios (Algo1-Algo5) expliqués ci-dessous :

III.6.1. Sans décomposition

- **Algo1 :** Cet algorithme correspond au scénario de référence (algorithme de base), qui correspond à la plupart des travaux existants. Dans ce scénario, l'entrée est le graphe tout entier (aucune décomposition n'est faite), et bien sûr l'exécution est séquentielle.

Algorithme 1 sans décomposition (algorithme de base)

Require: $G = (V, E)$: a network.
Ensure: *List_of_scores*(G).
1: compute the scores of all absent links in G:

$$List\_of\_Scores(G) = \left\{\left(x, y, s_{xy}^{method}\right) / x, y \in V and\ (x, y) \notin E\right\}$$

2: **return** *List_of_Scores*(G);

III.6.2. Avec décomposition (notre approche)

Dans cette partie nous proposons quatre nouvelles variantes d'algorithmes qui exploitent la décomposition (voir Tableau III-2) : ***Algo2*** est l'algorithme de base de notre approche qui sera comparé à l'algorithme de base (***Algo1***). Les autres algorithmes (***Algo3***-***Algo5***) sont des améliorations de notre algorithme de base (***Algo2***). Les améliorations proposées sont les suivantes :

- Au lieu de considérer l'étape de décomposition à l'intérieur de l'algorithme de prédiction de liens, il suffit de l'effectuer une fois et de démarrer l'algorithme à partir des composantes connexes. Cela évite l'exécution répétitive inutile d'une même étape de décomposition en plusieurs exécutions.
- Comme chaque composant peut être traité séparément des autres, on peut paralléliser l'algorithme au lieu d'utiliser un traitement séquentiel. ***Algo3*** (resp. ***Algo4***, ***Algo5***) résulte de ***Algo2*** en intégrant la première amélioration (resp. deuxième amélioration, les deux améliorations).
-

Tableau III-3: Les quatre variantes d'algorithmes issus de notre approche basée sur la décomposition

Type d'exécution	Étape avec la décomposition	
	Inclus	Non Inclus
Exécution séquentielle	Algo2	Algo3
Exécution parallèle	Algo4	Algo5

Les quatre scénarios de calcul de temps sont schématisés dans l'image suivante Figure III.4.

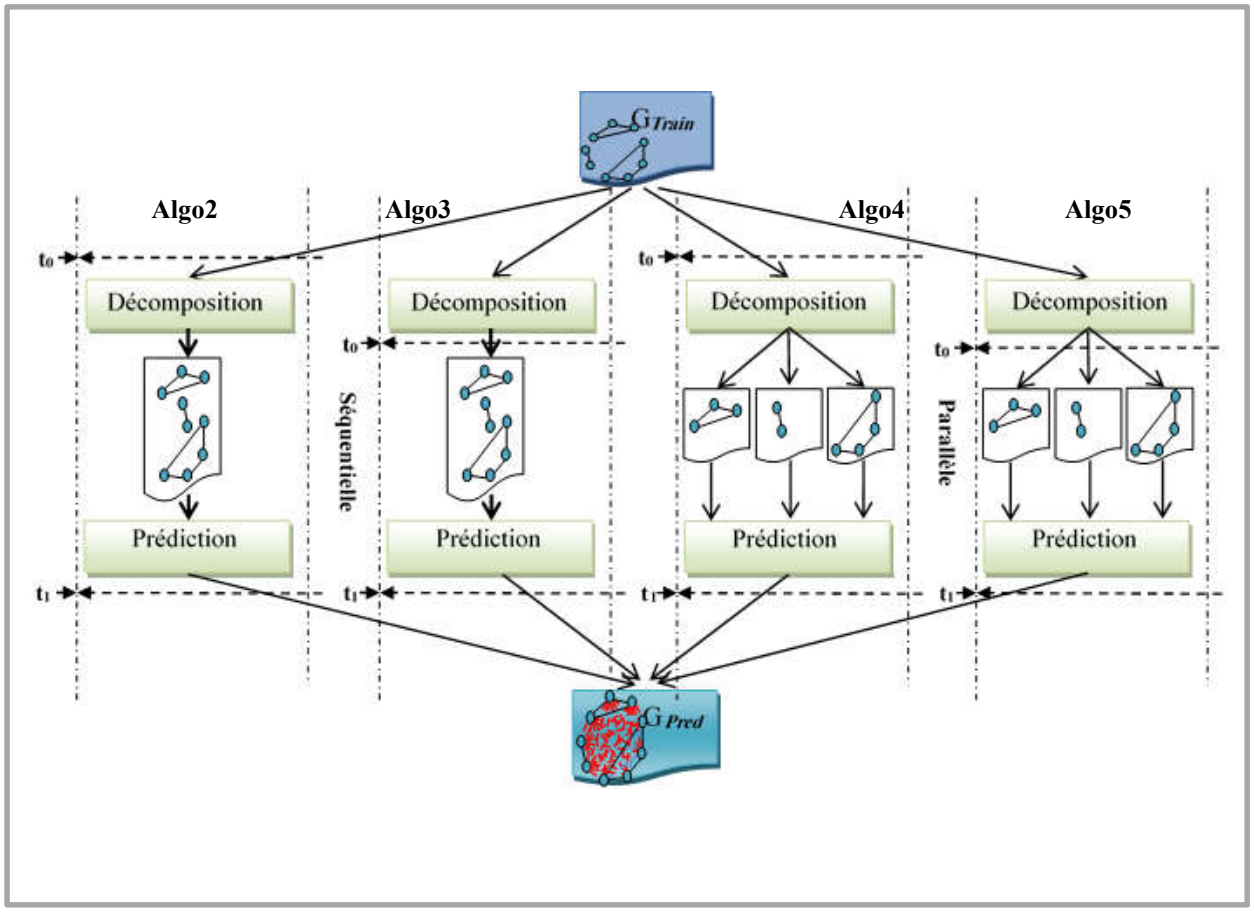


Figure III.4 : les quatre scénarios d'exécutions et de calcul de temps

- **Algo2 :** Cet algorithme inclut la décomposition du graphe d'entrée en composants connectés. Ensuite, la tâche de prédiction est exécutée séquentiellement sur l'ensemble des composants connectés.

Algorithme 2 Décomposition & Exécution séquentielle 1

Require: $G = (V,E)$: a network .

Ensure: *List_of_scores*(G).

1: Decompose the graph $G = (V,E)$ into c connected components :

$C \leftarrow \{G_1 = (V_1, E_1), G_2 = (V_2, E_2), \ldots, Gc = (V_c, E_c)\}$

2: For every component $G_i = (V_i,E_i)$, compute the scores of absent links:

$$Scores(G_i) = \left\{\left(x, y, s_{xy}^{method}\right)/x, y \in V_i and\ (x, y) \notin E_i\right\}$$

3: Merge the partial scores to form the global list of scores:

$$List\_of\_Scores(G) = \bigcup_{i=1}^{c} Scores(G_i) = \bigcup_{i=1}^{c} \left\{\left(x, y, s_{xy}^{method}\right)/x, y \in V_i and\ (x, y) \notin E_i\right\}$$

4: Return *List_of_Scores*(G);

- **Algo3 :** Cet algorithme est similaire à l'***Algo2***. mais il n'inclut pas l'étape de la décomposition, mais reçoit directement en entrée l'ensemble des composants connexes. La tâche de prédiction est ensuite exécutée séquentiellement sur l'ensemble des composants connexes. De ce fait, l'***Algo3*** est obtenu à partir d'Algo2., en modifiant l'entrée pour être l'ensemble des composants connexes et en éliminant la ligne 1.

Algorithme 3 Décomposition & Exécution séquentielle 2

Require: $C=\{G_1=(V_1, E_1), G_2=(V_2, E_2), ..., G_c=(V_c, E_c)\}$: a set of connected components of a network $G=(V, E)$.

Ensure: *List_of_scores*(G).

1: For every component $Gi=(Vi,Ei)$, compute the scores of absent links:

$$Scores(G_i) = \left\{(x, y, s_{xy}^{method})/x, y \in V_i and\ (x, y) \notin E\right\}$$

2: Merge the partial scores to form the global list of scores:

$$List\_of\_Scores(G) = \bigcup_{i=1}^{c} Scores(G_i) = \bigcup_{i=1}^{c} \left\{(x, y, s_{xy}^{method})/x, y \in V_i and\ (x, y) \notin E_i\right\}$$

3: Return *List_of_Scores*(G);

- **Algo4**. : Maintenant, cet algorithme inclut la décomposition du graphe en entrée en composantes connexes. Ensuite, la tâche de prédiction est effectuée par une exécution parallèle sur l'ensemble des composants connexes.

Algorithme 4 Décomposition & Exécution parallèle 1

Require: $G=(V, E)$: a network.

Ensure: *List_of_scores*(G).

1: Decompose the graph $G=(V, E)$ into c connected components :

$C \leftarrow \{ G_1=(V_1, E_1), G_2=(V_2, E_2), ..., G_c=(V_c, E_c)\}$

2: Find an equitable distribution of the set of components on the set of available m processors: $Proc_1,...,Proc_m$.

3.Parallel computation of the scores of absent links in each component:

$$Scores(G_i) = \left\{(x, y, s_{xy}^{method})/x, y \in V_i and\ (x, y) \notin E_i\right\}$$

$Scores(G_i)$ is computed in processor $proc_j$ to which the component G_i is affected.

4: Merge the partial scores to form the global list of scores:

$$List\_of\_Scores(G) = \bigcup_{i=1}^{c} Scores(G_i) = \bigcup_{i=1}^{c} \left\{(x, y, s_{xy}^{method})/x, y \in V_i and\ (x, y) \notin E_i\right\}$$

5: Return *List_of_Scores*(G);

- ***Algo5***. : Cet algorithme est similaire à *Algo4*, mais il n'inclut pas l'étape de décomposition mais reçoit directement en entrée l'ensemble des composants connexes. La tâche de prédiction est alors exécutée en parallèle sur les m processeurs. *Algo5* est donc obtenu à

partir *d'Algo4*, en modifiant l'entrée pour être l'ensemble des composants connexes et en éliminant la ligne 1.

Algorithme 5 Décomposition & Exécution parallèle 2

Require: $C=\{G_1 = (V_1, E_1), G_2 = (V_2, E_2), ..., G_c = (V_c, E_c)\}$: a set of connected components of a network $G = (V, E)$.
Ensure: *List_of_scores*(*G*).
1: Find an equitable distribution of the set of components on the set of available *m* processors : $Proc_1,...,Proc_m$.
2. Parallel computation of the scores of absent links in each component:

$$Scores(G_i) = \{(x, y, s_{xy}^{method})/x, y \in V_i and\ (x, y) \notin E_i\}$$

Scores (*G*i) is computed in processor $proc_j$ to which the component G_i is affected.
3: Merge the partial scores to form the global list of scores:

$$List\_of\_Scores(G) = \bigcup_{i=1}^{c} Scores(G_i) = \bigcup_{i=1}^{c} \{(x, y, s_{xy}^{method})/x, y \in V_i and\ (x, y) \notin E_i\}$$

4: Return *List_of_Scores*(*G*);

III-7-Conclusion

Dans ce chapitre, nous avons montré à ‘aide d’une étude théorique l’importance de la décomposition en composants connexes, pour réduire le nombre d’opérations de calcul dans l’ensemble des méthodes de prédiction de liens qui utilisent des mesures de similarité basées sur des informations sur le voisinage local, ou les chemins existant entre deux nœuds.

Le gain en termes de nombre d’opérations de calcul est exprimé formellement de façon précise. Nous avons démontré à travers des exemples que ce gain dépend fortement de la distribution des nœuds sur les composants connexes. Nous avons montré que la nature de cette distribution peut être capturé de façon quantitative à l’aide de deux concepts statistiques qui sont la variance et l’indice Gini.

Dans ce même chapitre, nous avons également proposé une nouvelle architecture globale parallèle pour la prédiction de lien. Par rapport à l’architecture traditionnelle, nous avons ajouté deux étapes à savoir : l’étape de décomposition en composants connexes et l’étape de parallélisation.

Pour mettre en évidence l’impact et l’importance de la décomposition en composantes connexes d’un côté et le parallélisme d’un autre côté, sur de temps d’exécution, nous avons proposé quatre variantes d’algorithmes exploitant la décomposition. Ces algorithmes feront l’objet d’une étude comparative avec l’algorithme de base traditionnel dans le prochain chapitre.

CHAPITRE –IV-

Résultats expérimentaux

Chapitre -IV-

Résultats expérimentaux

IV-1- Introduction

Dans ce chapitre, nous présentons une étude expérimentale détaillée pour valider l'intérêt de l'approche que nous avons proposée pour la prédiction de liens. Après une brève description des bases de données utilisées dans les expériences, nous présentons une série d'expériences permettant d'étudier l'impact de la décomposition sur le nombre de liens examinés et l'impact de la décomposition et de la parallélisation sur le temps d'exécution. L'étude expérimentale inclut également une comparaison entre les performances de nombreuses mesures de similarité.

IV-2- Les bases de données

Nous avons effectué nos expériences sur plusieurs ensembles de données (***Datasets***), représentant différents réseaux complexes du monde réel et provenant de différentes sources et domaines d'application. Ces ensembles de données sont couramment utilisé dans le domaine de la prédiction de liens.

Cette collection couvre un large éventail de propriétés, de tailles, de degrés moyens, de coefficients de regroupement et d'indices d'hétérogénéité (voir le tableau VI-1)[1].

- **ADV** (P.Massa et al., 2009) est un modèle de réseau qui prend en compte les comportements et les activités des individus pour prédire les liens sociaux qui se forment entre eux. Ceci en reconnaissant que les liens sociaux peuvent être influencés par des facteurs tels que les interactions en face à face, les activités communes et les centres d'intérêt partagés.
- **BUP** (Kunegis, 2013) est une base qui représente la blogosphère politique américaine et elle contient des informations sur les liens entre les blogs politiques, tels que les hyperliens entre les pages de blogs et les liens d'abonnement entre les blogs.
- **CDM** (Newman, 2001). contient un réseau de collaborations scientifiques de co-auteurs pour plusieurs domaines de recherche, notamment la physique, la biologie, la médecine, la psychologie et les sciences sociales.

[1] Cette collection est téléchargeable à l'adresse : http://noesis.ikor.org/datasets/linkprediction

- **CEG** (Watts and Strogatz, 1998) a été développé pour étudier les propriétés structurelles et dynamiques des réseaux de neurones biologiques. Le réseau CEG a été utilisé pour étudier divers aspects de la connectivité neuronale, notamment les propriétés topologiques, les dynamiques de réseau et les motifs de connexions.
- **CGS** (Batagelj and Mrvar, 2006) est une collection de réseaux de co-auteurs dans plusieurs domaines de la recherche. La base CGS contient des nœuds représentant les auteurs et des liens entre les auteurs qui sont co-auteurs d'un article scientifique. Les poids des liens représentent le nombre d'articles co-écrits par les auteurs respectifs, ainsi que le nombre de fois où chaque auteur s'est auto-cité dans ses propres articles.
- **EML** (Guimera et al., 2003) contient des informations sur les communications par courrier électronique entre les employés d'Enron. Elle était utilisée pour étudier la structure des réseaux sociaux au sein de l'entreprise et pour comprendre comment les informations et les idées se propagent dans un environnement organisationnel.
- **ERD** (Batagelj and Mrvar, 2006) contient des informations sur les co-auteurs et les collaborations entre chercheurs européens, basées sur l'analyse d'articles scientifiques publiés dans les revues entre 1991 et 2003. La base de données ERD permet d'analyser les réseaux de co-auteurs à différents niveaux d'agrégation, allant des collaborations entre individus aux collaborations entre institutions et pays.
- **FBK** (McAuley and Leskovec, 2012) est un ensemble de données de réseaux sociaux contenant des informations sur les interactions de "j'aime" entre les utilisateurs de Facebook. elle a été utilisée pour étudier divers aspects des interactions sociales, notamment les motifs de connexion, la diffusion de l'information et les tendances de popularité. Elle a été utilisée pour entraîner des algorithmes de recommandation.
- **GRQ** (Leskovec et al., 2007) a été proposé pour mesurer la réciprocité des liens dans les réseaux de co-auteurs dans quatre domaines de recherche différents : la biologie moléculaire, la sociologie, la psychologie et l'informatique. Un lien est considéré comme réciproque si deux auteurs se sont mutuellement cités dans leurs travaux.
- **HMT** (Kunegis, 2013) contient des informations sur les liens entre les acteurs, les films et les studios de production hollywoodiens. La base de données est construite en reliant les acteurs qui ont joué dans les mêmes films, les films qui partagent les mêmes acteurs, et les studios de production qui ont produit les mêmes films.
- **HPD** (S. Peri et al., 2003) est une base de données d'un réseaux biologiques, qui intègre une mine d'informations pertinentes sur la fonction des protéines humaines dans la santé et la maladie. Des données relatives à des milliers d'interactions protéine-protéine.
- **HTC** (Newman, 2001) est un réseau de co-auteurs en utilisant les données sur les publications dans les 20 principaux journaux scientifiques et de sciences sociales. Elle a identifié les auteurs les plus cités dans chaque domaine de recherche et a cartographié les liens de collaboration entre eux pour créer des réseaux de co-auteurs.

- **INF** (Isella et al., 2011) est une base de données qui a été collectée dans le cadre d'une étude visant à modéliser la propagation des maladies infectieuses au sein des réseaux de contacts humains. Elle contient des informations sur les interactions en face-à-face entre les individus dans différents contextes, tels que les lieux de travail, les écoles, les foyers, etc.
- **KNH** (Batagelj and Mrvar, 2006) est utilisé dans le contexte des réseaux de co-auteurs. C'est une base de données de discours officiels prononcés par les membres de l'assemblée nationale de Corée du Sud. Cette base de données a été utilisée pour créer des réseaux de co-auteurs en utilisant les informations sur les co-signataires de projets de loi ou de résolutions législatives.
- **LDG** (Batagelj and Mrvar, 2006) est basé sur des données de co-publication de chercheurs, où deux chercheurs sont reliés par une arête s'ils ont publié un article scientifique ensemble. La base de données LDG contiennent un réseau de co-publication pour plusieurs domaines de recherche différents, tels que la physique, l'informatique, la médecine, la biologie, etc.
- **NSC** (Newman, 2006) est un réseau de collaboration scientifique de co-auteurs pour comprendre les caractéristiques des collaborations entre scientifiques et les structures de collaboration dans différents domaines scientifiques.
- **PGP** (Boguña et al., 2004) est une base de données d'utilisation des logiciels de cryptage et de déchiffrement de données. Ils sont souvent utilisés pour protéger les communications électroniques, telles que les e-mails, en utilisant des algorithmes de cryptage à clé publique.
- **SMG** (Batagelj and Mrvar, 2006) est une base qui a a été développée pour étudier la structure de collaboration scientifique dans le domaine de la gestion, en utilisant les données de co-auteurs provenant de la base de données Social Science Citation Index. Le modèle SMG permet d'analyser les motifs de collaboration entre les auteurs, les clusters thématiques et les tendances temporelles de la collaboration dans les revues de gestion.
- **UAL** (P.Massa et al., 2009) contient la structure et les propriétés du réseau de transport aérien aux États-Unis. Elle représente les connexions aériennes entre les aéroports desservis par United Airlines. Elle a été utilisée pour étudier divers aspects du trafic aérien, tels que la distribution des flux de passagers et de marchandises, les itinéraires de vol les plus fréquents et les aéroports les plus connectés.
- **UPG** (Watts and Strogatz, 1998) est un réseau non orienté qui contient des informations sur le réseau électrique des États de l'Ouest des États-Unis d'Amérique. Une arête représente une ligne d'alimentation. Un nœud est soit un générateur, un transformateur ou une sous-station.
- **YST** (D. Bu, 2003) contient des informations sur l'expression des gènes dans la levure en réponse à différents types de stress, tels que la chaleur, le froid, la faim, les

rayonnements UV et les toxines environnementales. Elle peut être utilisée pour étudier les réseaux de régulation génique et les voies de signalisation impliquées dans les réponses au stress dans la levure.

- **ZWL** (Batagelj and Mrvar, 2006) est une base de données qui a été créé pour étudier la structure de collaboration entre les auteurs dans différentes disciplines. Les réseaux ZWL contiennent des nœuds représentant les auteurs et des liens entre les auteurs qui ont co-écrit un article scientifique. Les poids des liens représentent le nombre d'articles co-écrits par les auteurs respectifs.

Le tableau suivant illustre les propriétés des différentes bases de données

Tableau –IV-1 : Illustration des propriétés de 22 réseaux du monde réel. |V| désigne le nombre de nœuds, |E| est le nombre d'arêtes, ⟨*k*⟩ désigne le degré moyen, C représente le coefficient de regroupement (***clustering***), ASPL est la longueur moyenne du plus court chemin, D est la densité du graphe, H est l'indice d'hétérogénéité et R est le degré d'assortativité.

Name	|v|	|E|	⟨*k*⟩	C	ASPL	D	H	R
ADV	5155	39285	15.24	0.25	3.22	9	5.4060	-0.0951
BUP	105	441	8.40	0.49	3.08	7	1.4207	-0.1279
CDM	16264	47594	5.85	0.64	5.82	18	2.2087	0.1846
CEG	297	2148	14.46	0.29	2.46	5	1.8008	-0.1632
CGS	6158	11898	3.86	0.49	3.62	14	3.9467	0.2426
EML	1133	5451	9.62	0.22	3.61	8	1.9421	0.0782
ERD	6927	11850	3.42	0.12	3.78	4	12.6708	-0.1156
FBK	4024	87887	43.68	0.59	3.98	13	2.4320	0.0707
GRQ	5241	14484	5.53	0.53	5.05	17	3.0523	0.6593
HMT	2426	16630	13.71	0.54	3.15	10	3.1011	0.0474
HPD	8756	32331	7.38	0.11	4.19	14	4.5133	-0.0510
HTC	7610	15751	4.14	0.49	5.68	19	2.0986	0.02939
INF	410	2765	13.49	0.46	3.63	9	1.3876	0.2258
KHN	3772	12718	6.74	0.25	3.63	12	9.4220	-0.1205
LDG	8324	41532	9.98	0.31	4.37	16	6.1880	-0.0997
NSC	1461	2742	3.75	0.69	2.59	17	1.8486	0.4616
PGP	10680	24316	4.55	0.27	7.49	24	4.1465	0.2382
SMG	1024	4916	9.60	0.31	2.98	6	3.9475	-0.1925
UAL	332	2126	12.81	0.63	2.74	6	3.4639	-0.2079
UPG	4941	6564	2.67	0.08	18.99	46	1.4504	0.0035
YST	2284	6646	5.82	0.13	4.29	11	2.8479	-0.0991
ZWL	6651	54182	16.29	0.32	3.85	10	2.5851	0.0006

IV.3. Résultats et discussion

Pour démontrer l'efficacité de notre idée sur des bases de données du monde réel, nous avons divisé aléatoirement les liens existants de chaque réseau en deux ensembles : l'ensemble d'apprentissage et l'ensemble de test représentant respectivement 80 % et 20 % du total des liens.

Nous avons utilisé 22 différentes bases de données dans les expériences pour évaluer les performances des différentes méthodes proposées dans cette étude.

Tous les calculs ont été effectués sur une machine de type HP Z620Workstation avec processeur de modèle : Intel (R) Xeon (R) CPU E5-2620 v2 12 cœurs de fréquence 2,10 GHz et une taille de mémoire de 128 Go.

IV.3.1. L'impact de la décomposition sur le nombre de liens calculés

La première partie de nos expérimentations est consacrée à évaluer le gain obtenu pour les différentes bases de données utilisées. Les résultats obtenus sont résumés dans le ***Tableau IV.2***.

On précise que $\boldsymbol{Gain_1}$ dénote la différence entre le nombre total de liens calculés sans décomposition et ceux calculés en utilisant la décomposition :

$$\boldsymbol{Gain_1} = \left|\boldsymbol{E'_{Tr}}\right| - \sum_{\boldsymbol{i=1}}^{\boldsymbol{c}} \left|\boldsymbol{E'_{i,Tr}}\right| \qquad \text{(IV-1)}$$

Et

$$\boldsymbol{Taux_{Gain1}} = \frac{\left|\boldsymbol{E'_{Tr}}\right| - \sum_{\boldsymbol{i=1}}^{\boldsymbol{c}} \left|\boldsymbol{E'_{i,Tr}}\right|}{\left|\boldsymbol{E'_{Tr}}\right|} = \frac{\boldsymbol{Gain_1}}{\left|\boldsymbol{E'_{Tr}}\right|} \qquad \text{(IV-2)}$$

$\boldsymbol{Gain_2}$ dénote le gain normalisé compris dans l'intervalle ***[0,1]*** :

$$\boldsymbol{Gain_2} = \frac{\boldsymbol{Gain_1} - \boldsymbol{Gain_{min}}}{\boldsymbol{Gain_{max}} - \boldsymbol{Gain_{min}}} \qquad \text{(IV-3)}$$

Notons que ***Var*** et ***Gini*** dans le ***Tableau IV.2*** désignent les valeurs normalisées.

Tableau IV.2 Diverses distributions possibles des nœuds sur les composantes et leurs résultats de calcul de gain, Gini et variance

Base d'apprentissage			Sans Décomposition	Avec Décomposition		$Gain_1$		Taux (Norm)		
G	$\|V_{Tr}\|$	$\|E_{Tr}\|$	$\|E'_{Tr}\|$	$\|C\|$	$\sum\|E'_{c,Tr}\|$	Nb/liens	$Taux_{gain1}$ %	$Gain_2$	Var	Gini
BUP	105	352	5108	1	5108	**0**	-	0.00	0.00	0.00
CEG	297	1718	42238	1	42238	**0**	-	0.00	0.00	0.00
FBK	4024	70310	8023966	27	7867859	**156107**	**1.95**	0.01	0.99	0.01
INF	410	2212	81633	5	80003	**1630**	**2.00**	0.00	1.00	0.00
ZWL	6651	43346	22071229	114	21241223	**830006**	**3.76**	0.00	1.00	0.00
SMG	1024	3932	519844	35	483648	**36196**	**6.96**	0.00	1.00	0.00
EML	1133	4360	636918	48	583711	**53207**	**8.35**	0.00	1.00	0.00
LDG	8324	33225	34607101	337	31089311	**3517790**	**10.16**	0.02	0.98	0.02
UAL	332	1701	53245	14	47764	**5481**	**10.29**	0.03	0.97	0.03
KHN	3772	10175	7101931	218	6123668	**978263**	**13.77**	0.03	0.97	0.03
ADV	5155	31428	13253007	330	11333166	**1919841**	**14.49**	0.02	0.98	0.02
HPD	8756	25865	38303525	669	31436609	**6866916**	**17.93**	0.04	0.96	0.04
YST	2284	5316	2601870	202	2068432	**533438**	**20.50**	0.04	0.96	0.04
ERD	6927	9480	23978721	1045	17245416	**6733305**	**28.08**	0.00	1.00	0.00
CDM	16264	38075	132212641	1288	87661470	**44551171**	**33.70**	0.22	0.78	0.22
HMT	2426	13304	2928221	201	1907080	**1021141**	**34.87**	0.23	0.77	0.23
PGP	10680	19453	57006407	1230	35945368	**21061039**	**36.95**	0.19	0.81	0.19
UPG	4941	5275	12198995	458	7494909	**4704086**	**38.56**	0.25	0.75	0.25
GRQ	5241	11588	13719832	652	7387831	**6332001**	**46.15**	0.30	0.70	0.30
HTC	7610	12601	28939644	1022	14331576	**14608068**	**50.48**	0.34	0.66	0.34
CGS	6158	9519	18947884	1305	5437329	**13510555**	**71.30**	0.54	0.46	0.54
NSC	1461	2193	1064337	329	61890	**1002447**	**94.19**	0.90	0.10	0.90

A partir du ***Tableau IV.2***, nous pouvons extraire deux graphiques :

– Dans le premier graphe (voir ***Figure IV.1***), on peut observer que la valeur du *gain* est nulle pour les deux premières bases de données car ils ne contiennent qu'une seule composante connexe et donc le nombre de liens calculés avant et après la décomposition est le même. On a ensuite une progression de gain qui atteint **94,19%** du nombre de liens calculés sans décomposition mais ignorés en utilisant la décomposition.

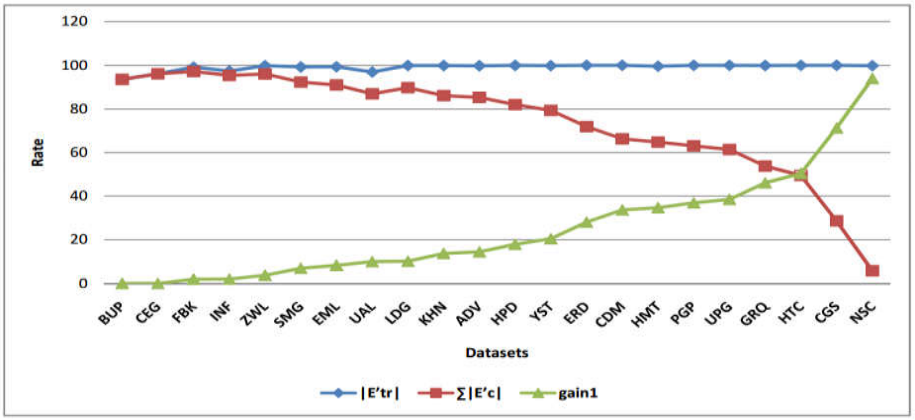


Figure IV.1 : Comparaison entre le nombre de liens calculé avec et sans décomposition.

– Le deuxième graphe (***Figure IV.2***) illustre l'influence du degré d'hétérogénéité de la distribution des nœuds à travers les composantes connexes sur le **gain** obtenu, capturé de manière équivalente par le ***Gini***, la ***Variance*** ou les mesures normalisées du ***gain*** direct. On peut remarquer que pour les cas où il n'y a qu'une seule composante, la distribution n'a pas de sens puisque toutes les mesures (***Gain***, ***Gini*** et ***Variance***) sont égales à zéro. Lorsque le nombre de composantes est supérieur à 1, la figure confirme que plus (resp. moins) la distribution est uniforme, plus (resp. moins) le gain est important, c'est-à-dire le nombre de liens ignorés.

Ce graphe confirme également, sur plusieurs bases de données réelles, les relations entre le gain normalisé, le Gini et les mesures de variance déjà évoquées dans le chapitre précédent par un petit exemple.

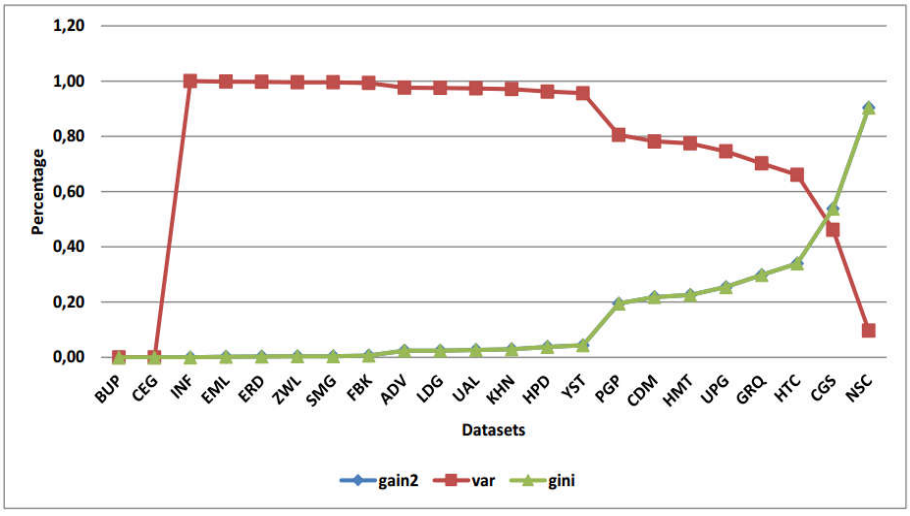


Figure IV.2 : Comparaison entre les valeurs Gain, Variance et Gini sur plusieurs bases de données

Pour terminer cette section, nous pouvons noter que la décomposition nous apporte certainement un gain lorsque le réseau complexe contient au moins deux éléments connectés.

Ce gain obtenu est plus important dans les situations où la distribution des nœuds à travers les composants est plus uniforme, ce qui nous permet de gagner considérablement en temps d'exécution, en particulier pour les grandes bases de données. La section suivante se concentre sur l'étude de l'impact de la décomposition sur le temps d'exécution

IV.3.2. Impact de la décomposition sur le temps d'exécution

Le but de cette section est de montrer comment la décomposition du graphe d'apprentissage permet de diminuer considérablement le temps d'exécution. La réduction du temps d'exécution est assurée grâce à la décomposition par deux moyens :

- D'abord comme discuté plus haut, la décomposition permet de réduire la quantité même de calculs nécessaires à effectuer (première expérience).
- De plus, la décomposition permet de gagner du temps d'exécution supplémentaire en parallélisant le processus de prédiction de lien sur plusieurs processeurs (deuxième expérience).

De plus, pour avoir une vue globale, nous donnons une comparaison globale de tous les algorithmes (l'algorithme de base et les différentes variantes de nos algorithmes) en utilisant quatre mesures de similarité et quatre bases de données (troisième expérience).

IV.3.2.1. Comparaison entre notre algorithme (Algo2) et l'algorithme de base(Algo1)

Dans cette première expérience, nous utilisons l'indice de ***Jaccard*** comme mesure de score. Nous exécutons tous les algorithmes 100 fois sur différentes bases de données synthétiques et nous prenons le temps d'exécution moyen obtenu pour chaque base de données. Nous partons de la base de données BUP (voir ***Tableau IV.1***) qui contient un composant et nous générons de nouveaux une base de données synthétique en dupliquant successivement la première pour obtenir des réseaux complexes à 2 composantes, 3 composantes et ainsi de suite jusqu'à 10 composantes. Notons que les bases de données obtenues sont parfaitement homogènes puisque toutes les composantes sont identiques et par conséquent, le gain obtenu est optimal théoriquement.

Nous comparons notre algorithme de base proposé qui utilise la décomposition et fonctionne de manière séquentielle (***Algo2***) avec l'algorithme de base sans décomposition qui correspond aux approches existantes (***Algo1***). Le but de cette comparaison est de montrer l'impact de l'exploitation de la décomposition en composants connexes sur la réduction du temps d'exécution dans la prédiction de lien. Le ***Tableau IV.3*** illustre les résultats obtenus à partir de la première expérience.

Tableau IV.3 Résultats obtenus par ***Algo1*** (algorithme de base) et notre algorithme ***Algo2*** (avec décomposition) sur différentes bases de données synthétiques avec l'index de ***Jaccard***

Base d'apprentissage			Sans Décomposition		Avec Décomposition		Gain			
							liens		temps	
|C|	$\|V_{Tr}\|$	$\|E_{Tr}\|$	$\|E'_{tr}\|$	Algo1 (sec)	$\sum\|E'_c\|$	Algo2 (sec)	Diff.	Taux %	Diff.	Taux %
1	105	353	5107	0.23	5107	0.23	0	0.00	0.00	0.00
2	210	706	21239	0.95	**10214**	**0.46**	11025	51.91	0.49	51.58
3	315	1059	48396	2.16	**15321**	**0.69**	33075	68.34	1.47	68.06
4	420	1412	86578	3.84	**20428**	**0.91**	66150	76.41	2.93	76.30
5	525	1765	135785	6.00	**25535**	**1.15**	110250	81.19	4.85	80.83
6	630	2118	196017	8.68	**30642**	**1.38**	165375	84.37	7.30	84.10
7	735	2471	267274	11.80	**35749**	**1.60**	231525	86.62	10.20	86.44
8	840	2824	349556	15.46	**40856**	**1.82**	308700	88.31	13.64	88.23
9	945	3177	442863	19.59	**45963**	**2.06**	396900	89.62	17.53	89.48
10	1050	3530	547195	24.15	**51070**	**2.28**	496125	90.67	21.87	90.56

À partir du ***Tableau IV.3***, nous pouvons extraire le graphe illustré dans la ***Figure IV.3***.

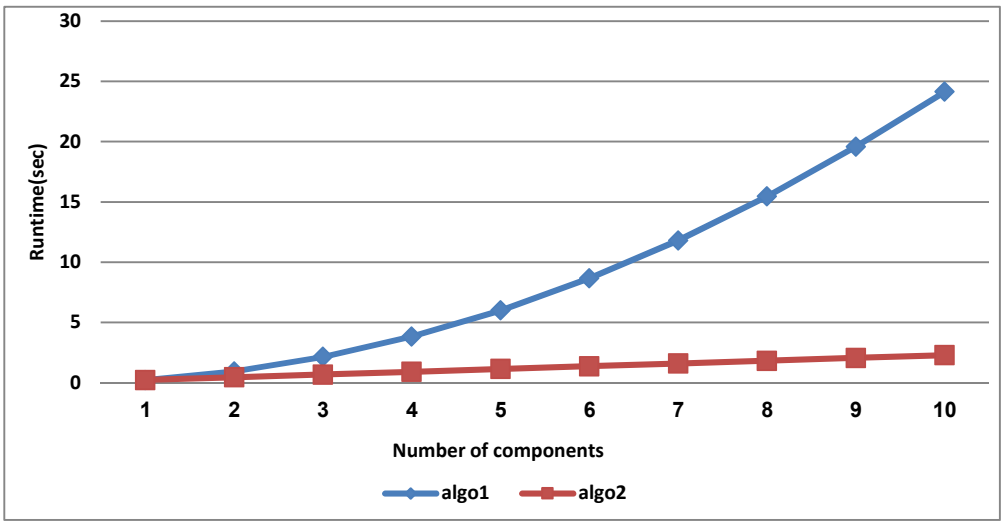


Figure IV.3 Comparaison entre l'algorithme de base (Algo1) et notre algorithme proposé (Algo2) en termes d'exécution sur des bases de données synthétiques

D'après le graphe représenté sur la ***Figure IV.3***, nous pouvons remarquer que notre algorithme utilisant la décomposition (***Algo2***) surpasse clairement l'algorithme de base qui n'utilise pas la décomposition (***Algo1***) avec un écart très important, en particulier dans le cas de grands graphes avec un nombre important de composants. Il est donc très bénéfique de décomposer le graphe avant les calculs.

IV.3.2.2. Comparaison entre les variantes de notre approche proposée

Nous avons démontré dans l'expérience 1 ci-dessus, l'intérêt de de la décomposition pour minimiser le nombre de liens et le temps d'exécution. Dans cette deuxième expérience, nous limitons la comparaison aux quatre variantes de notre approche proposée (***Algo2-Algo5***).

Rappelons que les quatre variantes sont basées sur la décomposition et qu'elles varient selon : (1) l'utilisation ou non du parallélisme et/ou (2) l'inclusion ou non de l'étape de décomposition dans le processus de calcul. Notez que dans cette expérience, nous utilisons les mêmes paramètres et suivons les mêmes étapes que dans l'expérience 1. Le ***Tableau IV.4*** illustre les résultats obtenus à partir de la deuxième expérience.

Tableau IV.4 les résultats obtenus par nos quatre algorithmes avec décomposition en composantes connexes sur différents jeux de données synthétiques avec index Jaccard

Base d'apprentissage			Avec Décomposition				
				Exécution séquentielle		Exécution parallèle	
|C|	$\|V_{Tr}\|$	$\|E_{Tr}\|$	$\sum\|E'_c\|$	Algo2(sec)	Algo3(sec)	Algo4(sec)	Algo5(sec)
1	105	353	5107	0.23	0.23	0.23	**0.23**
2	210	706	10214	0.46	0.45	0.27	**0.26**
3	315	1059	15321	0.69	0.67	0.28	**0.27**
4	420	1412	20428	0.91	0.89	0.31	**0.28**
5	525	1765	25535	1.15	1.12	0.33	**0.30**
6	630	2118	30642	1.38	1.34	0.36	**0.30**
7	735	2471	35749	1.60	1.57	0.38	**0.35**
8	840	2824	40856	1.82	1.79	0.43	**0.40**
9	945	3177	45963	2.06	2.01	0.45	**0.40**
10	1050	3530	51070	2.28	2.23	0.47	**0.42**

Du **Tableau IV.4**, nous pouvons extraire le graphe illustré à la Figure IV.4.

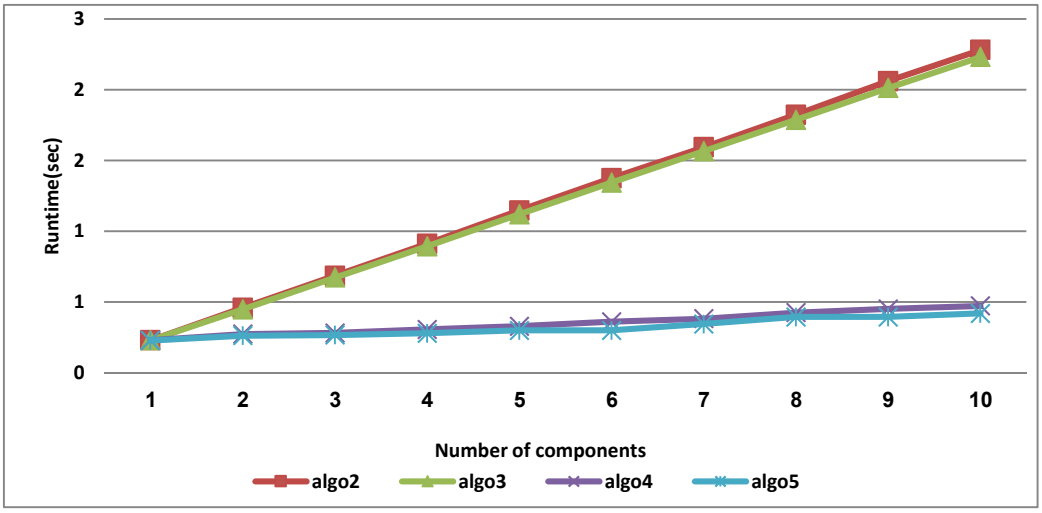


Figure IV.4 : Comparaison entre les quatre notre algorithmes (***Algo2-Algo5***) en utilisant la décomposition en termes de temps d'exécution sur des bases de données synthétiques

D'une part, on constate que le temps nécessaire pour décomposer le graphe en composantes connexes est presque négligeable. En effet, on constate que la courbe de l'algorithme séquentiel (resp. parallèle) ***Algo2*** (resp. ***Algo4***) incluant le pas de décomposition est très proche de celle de l'algorithme séquentiel (resp. parallèle) ***Algo3***. (resp. ***Algo5***) qui reçoit directement en entrée le graphe décomposé.

Par contre, on peut observer que les algorithmes parallèles ***Algo4*** et ***Algo5*** sont nettement plus rapides que les algorithmes séquentiels ***Algo2*** et ***Algo3***.

En résumé, nous pouvons confirmer l'efficacité de l'utilisation de la décomposition avant la prédiction de lien. En effet, cela réduit considérablement la quantité de calculs nécessaires et conduit par conséquent à une diminution remarquable du temps d'exécution.

De plus, la parallélisation du processus de prédiction de lien, rendue possible grâce à l'utilisation de composants connexes au lieu d'un réseau entier en entrée, nous apporte un gain supplémentaire en termes de temps d'exécution. Plus le réseau est grand, c'est-à-dire qu'il a un plus grand nombre de composants et que la répartition des nœuds à travers les composants est plus homogène, plus l'effet de la parallélisation est important dans la réduction du temps d'exécution.

IV.3.2.3. Comparaison globale entre tous les algorithmes

Dans la troisième expérience, nous avons appliqué les cinq algorithmes (***Algo1-Algo5***) sur quatre bases de données réels : NSC, CGS, HTC et GRQ. De plus, nous avons utilisé pour chaque base de données quatre mesures de score : l'indice de ***Jaccard***, le Voisin commun (***CN***), l'indice d'Adamic Adar (***AA***) et l'indice d'allocation des ressources (***RA***). La mise en œuvre de ces mesures se trouve dans la bibliothèque ***networkx*** [2].

Les résultats sont présentés dans les graphes de la ***Figure IV.5***. D'après ces graphiques :

- Nous pouvons remarquer que les quatre variantes de notre approche proposée utilisant la décomposition (***Algo2-Algo4***) surpassent nettement l'algorithme de base qui n'utilise pas la décomposition (***Algo1***) avec un écart important, surtout dans le cas de grands graphes avec un nombre important de composantes.
- Il est également à noter que le taux de gain en temps de calcul entre l'algorithme de référence ***Algo1*** et les autres algorithmes proposés varie en fonction de l'évolution du taux de variance (resp. ***Gini***) : plus la variance (resp. ***Gini***) est faible (resp. forte) , plus le taux de gain est élevé (resp. faible) et inversement.

VI-3.3. Etude comparative entre les différentes mesures de similarité :

Cette section est destinée à évaluer et établir une comparaison objective entre les différentes mesures de similarités de notre étude de l'état de l'art, la comparaison est basée sur des critères pratiques : le temps d'exécution, la précision moyenne et la mesure AUC moyenne.

Dans notre évaluation, nous avons suivi directement les étapes de notre architecture proposée pour gagner plus de temps dans les expériences pratiques,

VI-3.3.1. La préparation des bases de données :

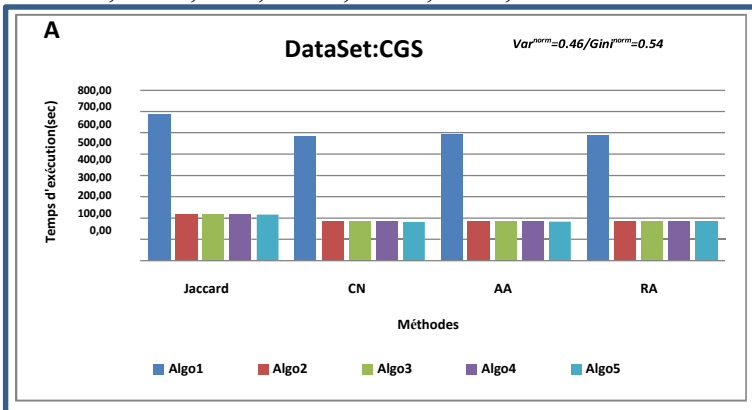
Pour arriver à un résultat optimal, les bases de données sont divisées en cinq parties de 20% de liens, et pour chaque itération de validation, nous prenons une partie pour construire la base de test, et les quatre autres parties pour construire la base d'apprentissage, et ainsi de

[2] https://networkx.org/documentation/stable/reference/algorithms/link_prediction.html

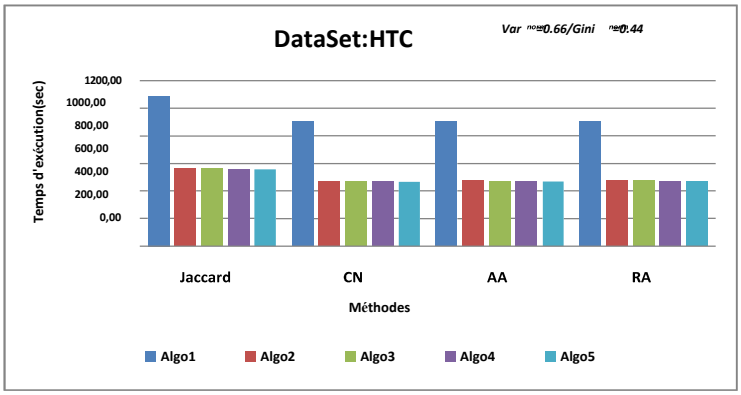
suite. A la fin nous obtenons cinq itérations, les mesures obtenues en chaque itération font généralement l'objet d'une moyenne pour obtenir un score d'évaluation global unique.

Parmi les bases de données disponibles, nous avons choisi les bases suivantes : BUP, CEG, UAL, INF, SMG, EML, NSC, YST et HMT.

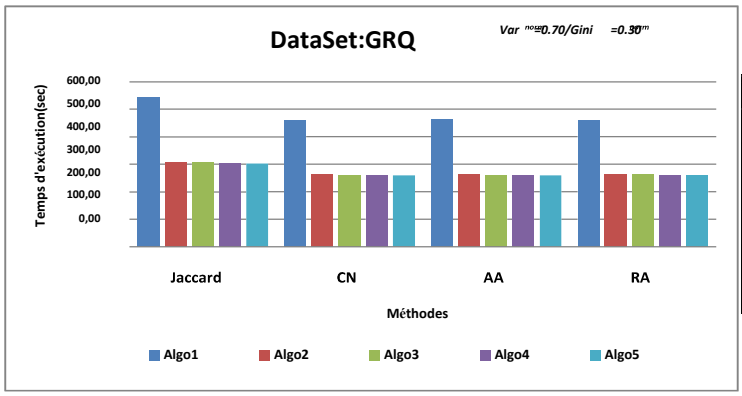
A



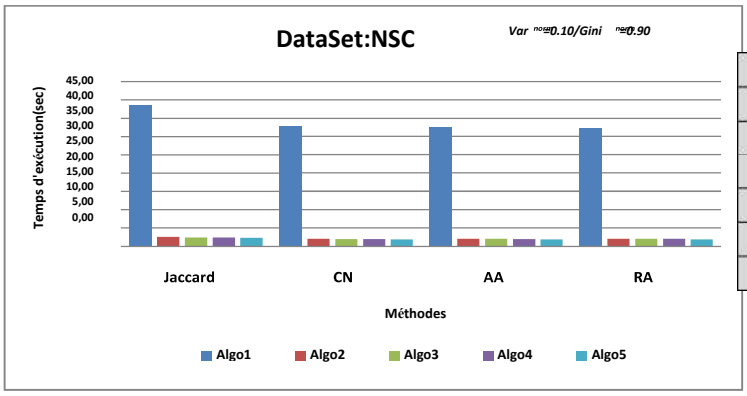
Base de données : CGS				
	Jaccard	CN	AA	RA
Algo1	688.09	583.73	594.12	587.74
Algo2	218.92	186.31	185.49	185.47
Algo3	217.52	184.40	184.17	185.05
Algo4	217.15	183.22	183.04	183.76
Algo5	**216.25**	**181.71**	**182.19**	**182.70**



Base de données : HTC				
	Jaccard	CN	AA	RA
Algo1	1 088.78	906.37	906.91	905.00
Algo2	563.40	469.75	473.90	474.19
Algo3	560.33	468.18	471.14	473.68
Algo4	558.37	467.87	470.65	469.22
Algo5	**556.28**	**466.84**	**469.33**	**468.53**



Base de données : GRQ				
	Jaccard	CN	AA	RA
Algo1	542.48	459.93	461.94	460.67
Algo2	306.73	264.91	262.57	262.63
Algo3	305.51	261.34	261.02	262.05
Algo4	304.21	260.44	260.90	261.56
Algo5	**303.10**	**260.15**	**259.75**	**260.48**



Base de données : NSC				
	Jaccard	CN	AA	RA
Algo1	38.47	32.65	32.44	32.38
Algo2	2.57	2.07	2.08	2.07
Algo3	2.43	2.02	2.03	2.05
Algo4	2.41	1.98	2.03	2.04
Algo5	**2.32**	**1.93**	**1.94**	**1.97**

Figure IV.5 : Comparaison globale entre tous les algorithmes

VI-3.3.2. Le paramétrage des méthodes

Nous avons programmé 47 méthodes, 29 locales, 13 globales et 5 méthodes Quasi-locales.

Les méthodes ayant des paramètres sont évaluées en prenant les valeurs optimales des paramètres qui ont été choisis par leurs propriétaires.

La méthode ***CNC*** est exécutée avec le paramètre ***alpha = 0.8***, la méthode ***CNPA*** est testée avec ***alpha=0.01***, la méthode ***CND*** est teste avec ***beta=0.01***, la méthode ***ADP*** est testée avec ***beta=2.5***, la méthode **Katz** est testée avec ***beta=0.001*** et ***l = 3***, la méthode ***GLHN*** est testée avec ***theta = 0.5***, la méthode ***RWR*** est testée avec ***alpha=0.3*** , la méthode ***FP*** est exécutée avec les paramètres ***alpha = 0.85*** , le nombre ***d'itérations= 10,*** la méthode ***SR*** est testée avec les paramètres ***lambda = 0.8*** et le nombre ***d'itération = 5***, ***LP*** est testée avec ***beta=0.01*** ,les méthodes ***SRW*** , ***LRW*** sont testées avec ***lambda = 0.01*** et le nombre ***d'itération =10***, ***RW*** est testée avec le nombre ***d'itération =10*** , ***RPR*** est testée avec ***lambda = 0.50*** , ***LIT*** est testée avec le nombre ***d'itération =2*** ,en fin ***RP*** est testée avec ***alpha = 1*** et ***beta = 1.***

VI.3.3.3. Les résultats

- Le premier test est destiné à comparer les méthodes suivant le temps de calcul. La comparaison des temps d'exécution est présentée dans le tableau IV-5, qui résume les résultats obtenus

Dans le tableau IV-5, on peut déduire que la méthode ***SAM*** est plus rapide que les autres méthodes. Les trois méthodes de Local Naïve Bayes sont les plus lents avec un grand écart qui atteint le triple

Tableau IV.5 : le temps d'exécution moyen des cinq itérations pour chaque pair de réseau et méthode

N°	Méthode	BUP	CEG	EML	HMT	INF	NSC	SMG	UAL	YST
I.01	CN	0.1573	1.6547	21.6126	99.1335	2.9228	8.3363	16.6764	1.6038	70.1582
I.02	AA	0.1711	1.7802	21.8089	101.0622	3.0564	8.3809	17.1030	1.7551	70.6626
I.03	RA	0.1686	1.7518	21.7892	100.5498	3.0233	8.3840	17.0315	1.7327	70.4502
I.04	RACNI	0.4094	7.6636	46.4970	343.5072	9.5745	9.1735	45.7702	11.7810	105.4652
I.05	JI	0.1845	1.8886	24.6027	110.0009	3.3538	8.6612	19.2066	1.8648	79.7032
I.06	SA	0.1901	1.9079	25.1952	110.6416	3.4497	8.7307	19.5977	1.8984	82.5782
I.07	SO	0.1893	1.8735	24.7100	109.1498	3.3753	8.6814	19.2571	1.8680	80.8798
I.08	HPI	0.1894	1.8906	25.0134	110.9350	3.4102	8.7023	19.5171	1.8781	81.5508
I.09	HDI	0.1884	1.8892	24.9647	109.9505	3.4046	8.7158	19.4220	1.8719	81.3380
I.10	LLHN	0.1876	1.8763	24.6886	109.1658	3.3833	8.6775	19.2503	1.8598	80.8691
I.11.1	IA1	0.2411	3.1518	23.8639	128.8662	3.8402	8.4473	22.2858	3.9802	71.4002
I.11.2	IA2	0.4663	6.1815	46.6374	242.8628	7.5341	10.2234	43.6657	7.7183	130.9161
I.12.1	LNB-AA	0.2977	25.7057	30.7019	948.0200	5.4006	8.5662	381.8259	53.6681	82.9393
I.12.2	LNB-CN	0.2903	25.6387	30.5543	946.6589	5.3706	8.5550	381.7366	53.5389	82.7920
I.12.3	LNB-RA	0.3043	25.7024	30.7863	950.2410	5.4338	8.5688	382.0216	53.6403	82.9968
I.13.1	CAR-AA	0.1707	1.9072	21.6140	102.1458	3.0656	8.3277	17.5961	1.9986	69.0989
I.13.2	CAR-CN	0.3203	3.4601	42.5130	187.6906	5.9182	10.0279	33.4593	3.4582	128.4277
I.13.3	CAR-RA	0.1716	1.8953	21.5300	101.9607	3.0435	8.3230	17.5785	1.9768	69.0761
I.14	LIT	0.2307	2.5238	31.1588	146.0420	4.5835	9.0288	24.5248	2.5356	94.6441
I.15	FSW	0.1908	1.9500	25.5574	112.5189	3.5047	8.7606	20.0183	1.9367	82.3889
I.16	SAM	**0.0345**	**0.2918**	**4.4150**	**25.0871**	**0.5696**	**7.0117**	**3.5546**	**0.3460**	**22.6305**
I.17	CCLP	0.3402	7.5499	28.3696	273.8003	5.3065	8.6547	59.6726	16.6850	76.3810
I.18	TRA	0.3050	4.3229	25.8920	150.2835	4.6957	8.5970	26.3231	6.1552	73.4378
I.19	CNPA	0.1837	1.8840	25.0152	109.7185	3.3947	8.7082	19.3173	1.8691	81.3648
I.20	CN2D	0.3438	3.6070	46.0213	197.1652	6.3003	10.4179	35.8779	3.5481	141.1207
I.21	ADP	0.1636	1.7647	21.4291	99.2846	2.9999	8.3581	16.8116	1.7370	68.8408
I.22	NLC	0.5010	12.4450	33.6910	369.3598	7.0855	8.9512	93.6489	25.8781	82.7629
I.23	LAS	0.3485	3.5493	46.1848	197.1949	6.3309	10.4253	35.7884	3.4742	141.7006
I.24	LNL	0.1786	2.0962	21.9858	107.2254	3.1912	8.3494	18.1728	2.4043	69.3696
II.01	NSP	0.1085	0.6184	15.0527	73.4105	3.3384	10.3769	10.8607	0.7517	74.2856
II.02	KI	0.1255	0.5438	20.5365	120.5780	1.2810	7.5623	15.5677	0.6454	133.4738
II.03	GLHNI	0.0998	0.5939	21.6983	127.2345	1.3905	7.7308	16.3461	0.7217	140.9361
II.04	RW	0.0623	0.5995	21.8054	125.0992	1.4095	7.6842	16.5278	0.7326	138.5310
II.05	RWR	0.0542	0.6006	21.5308	124.0521	1.4006	7.6883	16.3758	0.7281	137.1961
II.06	FP	0.0625	0.5456	20.9502	122.6339	1.2965	7.7715	15.8140	0.6769	135.4926
II.07	SR	0.0654	0.6063	21.8427	125.5448	1.4422	7.7219	16.5756	0.7399	138.7890
II.08	PLM	0.0733	0.6361	22.5374	128.9546	1.4733	8.1365	16.9511	0.7821	142.3268
II.09	ACT	0.0781	0.6622	22.9583	130.0121	1.5448	8.1827	17.2382	0.8134	143.5307
II.10	RFK	0.0593	0.5225	20.8014	121.0653	1.2712	7.9240	15.5748	0.6644	133.4590
II.11	LDI	0.0654	0.5512	21.4004	124.7245	1.3132	8.0050	15.9672	0.6883	137.5306
II.12	NGLI	0.1176	0.6333	15.2110	73.7250	3.3740	10.3846	10.9746	0.7659	74.8371
II.13	RPR	0.0422	0.5354	20.6431	121.3473	1.2691	7.5849	15.6162	0.6494	134.2018
III.01	LPI	0.0543	0.5205	20.5491	121.0105	1.2644	7.5444	15.5186	0.6491	133.3001
III.02	LRW	0.0637	0.6048	21.8291	125.3615	1.4102	7.7058	16.5700	0.7363	138.3934
III.03	SRW	0.0648	0.6155	21.8767	125.3700	1.4507	7.7181	16.5943	0.7388	138.7132
III.04	CND	0.3064	2.8847	38.7023	175.1570	6.7331	12.3326	30.7910	2.9255	141.1548
III.05	CNC	0.3877	2.9586	51.8867	226.4100	9.7721	16.0636	38.9121	3.1757	204.3410

.

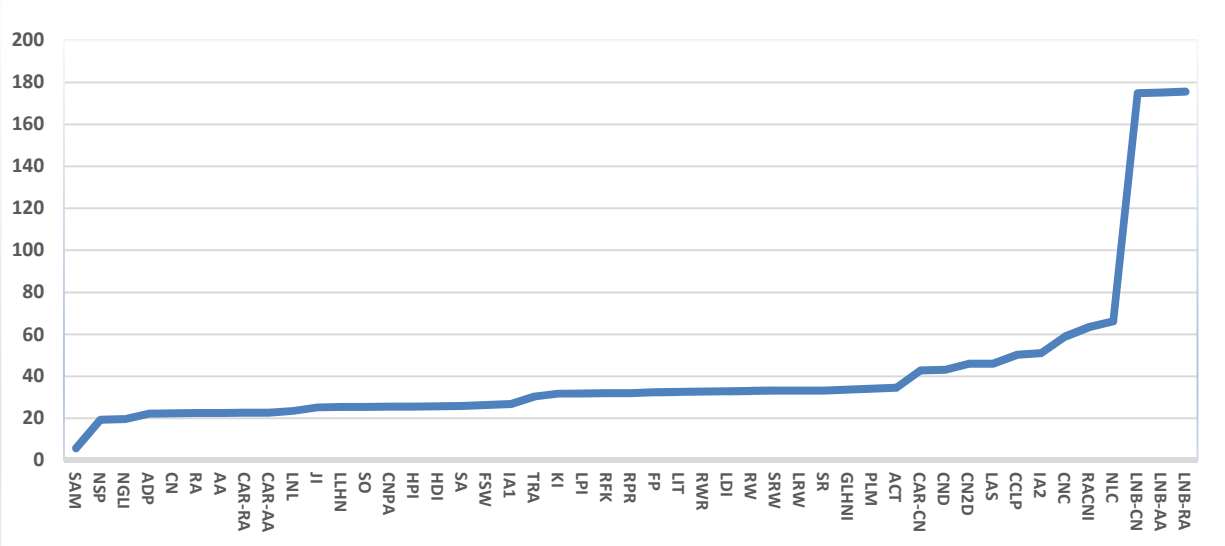


Figure IV.6 : le temps d'exécution moyen de chaque méthode pour tous les réseaux

Tableau IV-6 : Le nombre de fois que chaque méthode apparaît dans le top cinq pour tous les réseaux en fonction de temps de calcul

N°	Méthode	Première	deuxième	troisième	Top 5
I.01	CN				1
I.13.1	CAR-AA				1
I.13.3	CAR-RA			1	
I.16	SAM	9			
I.21	ADP		1		1
I.24	LNL				1
II.01	NSP		3		
II.02	KI		1	1	4
II.04	RW				1
II.05	RWR			1	
II.10	RFK			1	3
II.12	NGLI			3	
II.13	RPR		1	1	3
III.01	LPI		3	1	3

Le tableau IV-6 exprime le nombre de fois que chaque méthode apparaît en première, deuxième et troisième position et dans les cinq premières pour tous les réseaux en fonction de temps d'exécution.

On peut constater que les méthodes globales sont plus présentes dans le top 5 (23 fois), c'est-à-dire qu'elles sont plus rapides. Les méthodes locales sont moins présentes dans le top 5 15 fois). Enfin une seule méthode ***LPI*** Quasi-local apparaît 9 fois dans le top 5.

- Le deuxième test est destiné à comparer les méthodes en fonction de la précision moyenne des cinq itérations de calcul réalisées pour chaque paire méthode et réseau. Les résultats obtenus sont résumés dans le tableau IV-7.

Tableau IV.7 : la Précision moyenne des cinq itérations pour chaque pair de réseau et méthode.

N°	Méthode	BUP	CEG	EML	HMT	INF	NSC	SMG	UAL	YST
I.01	CN	0.1730	0.0845	0.1097	0.1873	0.3167	0.5459	0.0690	0.4148	0.0425
I.02	AA	0.1882	0.0918	0.1151	0.2567	0.3394	0.6778	0.0774	0.4503	0.0405
I.03	RA	0.1923	0.0901	0.0923	0.3532	0.3531	**0.6883**	0.0705	**0.5071**	0.0304
I.04	RACNI	0.1699	0.0914	0.0809	0.1831	0.2800	0.2809	0.0671	0.4458	0.0728
I.05	JI	0.1044	0.0312	0.0410	0.1913	0.3316	0.4769	0.0067	0.0780	0.0080
I.06	SA	0.1125	0.0302	0.0354	0.1883	0.3247	0.4753	0.0067	0.0756	0.0078
I.07	SO	0.1044	0.0312	0.0410	0.1913	0.3316	0.4769	0.0067	0.0780	0.0080
I.08	HPI	0.1489	0.0372	0.0349	0.0934	0.1832	0.4163	0.0324	0.0848	0.0277
I.09	HDI	0.0920	0.0306	0.0432	0.1877	0.3254	0.4737	0.0067	0.0771	0.0082
I.10	LLHN	0.0726	0.0189	0.0188	0.0462	0.1115	0.2485	0.0055	0.0190	0.0065
I.11.1	IA1	0.1823	0.0997	0.1221	0.3258	0.3473	0.6588	0.0780	0.4466	0.0506
I.11.2	IA2	0.1753	0.0968	0.1044	0.3779	0.3528	0.6776	0.0714	0.4644	0.0431
I.12.1	LNB-AA	0.1914	0.0949	0.1203	0.3019	0.3344	0.2774	0.0801	0.4685	0.0490
I.12.2	LNB-CN	0.1813	0.0934	0.1171	0.2391	0.3099	0.2360	0.0773	0.4327	0.0507
I.12.3	LNB-RA	0.1915	0.0909	0.0960	0.3478	0.3603	0.2889	0.0720	0.5049	0.0381
I.13.1	CAR-AA	0.1474	0.0846	0.1193	0.2777	0.3285	0.5680	0.0629	0.4053	0.0738
I.13.2	CAR-CN	0.1470	0.0810	0.1161	0.2239	0.3174	0.5381	0.0614	0.4015	0.0710
I.13.3	CAR-RA	0.1532	0.0909	0.1258	0.3629	0.3455	0.5850	0.0708	0.4398	0.0770
I.14	LIT	0.1227	0.0467	0.0938	0.2645	0.3480	0.5800	0.0143	0.1833	0.0159
I.15	FSW	0.1077	0.0421	0.0740	0.2345	0.3093	0.5649	0.0144	0.1628	0.0199
I.16	SAM	0.1217	0.0300	0.0261	0.1626	0.2767	0.4676	0.0073	0.0485	0.0074
I.17	CCLP	0.1800	0.0930	0.1143	0.2967	0.3042	0.5399	0.0776	0.4407	0.0534
I.18	TRA	0.1836	0.1076	0.1239	0.3522	0.3229	0.6123	0.0893	0.4538	0.0649
I.19	CNPA	0.1702	0.0837	0.0929	0.1474	0.2967	0.4696	0.0584	0.4162	0.0342
I.20	CN2D	0.1538	0.0754	0.1012	0.1829	0.3120	0.4809	0.0598	0.4050	0.0370
I.21	ADP	0.1937	0.0931	0.1176	0.3379	0.3582	0.4796	0.0783	0.4823	0.0425
I.22	NLC	0.1755	0.1015	0.1086	0.3661	0.3100	0.5386	0.0821	0.4654	0.0636
I.23	LAS	0.1217	0.0300	0.0261	0.1643	0.2767	0.4675	0.0073	0.0485	0.0074
I.24	LNL	0.1836	0.1076	0.1239	0.3520	0.3229	0.6123	0.0893	0.4538	0.0649
II.01	NSP	**0.4794**	**0.4537**	**0.3525**	**0.4718**	**0.4754**	0.4663	**0.3675**	0.4841	**0.1972**
II.02	KI	0.1724	0.0870	0.1012	0.1819	0.3009	0.4588	0.0690	0.4139	0.0454
II.03	GLHNI	0.0696	0.0182	0.0173	0.0576	0.1318	0.2605	0.0033	0.0146	0.0042
II.04	RW	0.1142	0.0775	0.0228	0.0964	0.0852	0.3405	0.0408	0.0770	0.0094
II.05	RWR	0.1998	0.1115	0.0436	0.2324	0.2321	0.4606	0.0530	0.2071	0.0168
II.06	FP	0.1384	0.0557	0.0194	0.0373	0.0979	0.1484	0.0304	0.3618	0.0152
II.07	SR	0.0839	0.0216	0.0193	0.0661	0.1337	0.2787	0.0052	0.0210	0.0042
II.08	PLM	0.1782	0.0709	0.0588	0.1961	0.1402	0.0885	0.0366	0.2458	0.0221
II.09	ACT	0.1368	0.0416	0.0128	0.0272	0.0812	0.3204	0.0233	0.3451	0.0061
II.10	RFK	0.1404	0.0548	0.0318	0.1635	0.1744	0.4685	0.0168	0.0898	0.0103
II.11	LDI	0.0979	0.0472	0.0212	0.0731	0.0590	0.0295	0.0129	0.0517	0.0065
II.12	NGLI	0.0934	0.0503	0.0144	0.0569	0.0400	0.4260	0.0301	0.3416	0.0066
II.13	RPR	0.1330	0.0561	0.0446	0.2152	0.1899	0.4066	0.0175	0.1130	0.0143
III.01	LPI	0.1730	0.0877	0.1016	0.1747	0.3006	0.4656	0.0693	0.4139	0.0461
III.02	LRW	0.1665	0.0692	0.0332	0.1952	0.1923	0.4645	0.0289	0.1551	0.0096
III.03	SRW	0.1664	0.0692	0.0331	0.1952	0.1922	0.4645	0.0289	0.1550	0.0096
III.04	CND	0.1735	0.0845	0.1092	0.1871	0.3168	0.5357	0.0689	0.4145	0.0417
III.05	CNC	0.1735	0.0845	0.1092	0.1845	0.3168	0.1402	0.0689	0.4145	0.0416

D'après le tableau IV-7, la méthode ***NSP*** est la méthode qui s'est avérée obtenir les meilleurs résultats en précision moyenne dans nos expériences dans la plupart des réseaux

(***BUP, CEG, EML, HMT, INF, SMG*** et ***YST***), et ensuite la méthode ***RA*** en deux réseaux (***NSC*** et ***UAL***)

Le tableau IV-8 exprime le nombre de fois que chaque méthode apparaît en première, deuxième et troisième position et dans les cinq premières pour tous les réseaux en fonction de précision moyenne :

Dans ce tableau, on peut constater que les méthodes locales sont plus présentes dans le top 5 (35 fois). Elles ont la précision la plus élevée. Ensuite, on trouve les méthodes globales (10 fois). En revanche, on ne trouve aucune méthode Quasi-locale classé parmi les top5.

Tableau IV-8 : Le nombre de fois que chaque méthode apparaît dans le top cinq pour tous les réseaux en fonction de la précision moyenne

N°	Méthode	Première	deuxième	troisième	Top 5
I.02	AA		1		
I.03	RA	2			3
I.04	RACNI				1
I.11.1	IA1				2
I.11.2	IA2		1	1	1
I.12.1	LNB-AA				2
I.12.3	LNB-RA		2		1
I.13.1	CAR-AA			1	
I.13.2	CAR-CN				1
I.13.3	CAR-RA		2		1
I.18	TRA		1	2	1
I.21	ADP			2	1
I.22	NLC			1	2
I.24	LNL			1	2
II.01	NSP	7		1	
II.05	RWR		2		

- Le troisième test est effectué pour comparer les méthodes en fonction de l'***auc*** moyenne des cinq itérations de calcul réalisées pour chaque paire méthode et réseau. Les résultats obtenus sont résumés dans le tableau IV-9.

Dans le tableau IV-9, on peut voir que la méthode ***RWR*** a les meilleurs résultats calculés dans les réseaux ***BUP, CEG, HMT, INF***, et ***SMG***, la méthode ***LDI*** dans le réseau ***EML*** et ***YST***, la méthode ***LPI*** dans le réseau ***NSC***, et enfin la méthode ***ADP*** dans le réseau ***UAL***

Tableau IV.9 : l'AUC moyen des cinq itérations pour chaque pair de réseau et méthode

N°	Méthode	BUP	CEG	EML	HMT	INF	NSC	SMG	UAL	YST
I.01	CN	0.8691	0.8276	0.8234	0.9523	0.9264	0.9058	0.8138	0.9278	0.6850
I.02	AA	0.8781	0.8450	0.8251	0.9553	0.9300	0.9061	0.8224	0.9391	0.6855
I.03	RA	0.8801	0.8485	0.8248	0.9561	0.9305	0.9061	0.8224	0.9439	0.6854
I.04	RACNI	0.8934	0.8324	0.8889	0.9652	0.9515	0.9137	0.8777	0.9279	0.8203
I.05	JI	0.8559	0.7767	0.8215	0.9492	0.9286	0.9059	0.7751	0.8927	0.6837
I.06	SA	0.8621	0.7832	0.8209	0.9501	0.9285	0.9059	0.7777	0.8996	0.6837
I.07	SO	0.8559	0.7767	0.8215	0.9492	0.9286	0.9059	0.7751	0.8927	0.6837
I.08	HPI	0.8682	0.7933	0.8186	0.9484	0.9255	0.9058	0.7866	0.8667	0.6835
I.09	HDI	0.8476	0.7680	0.8215	0.9483	0.9277	0.9058	0.7747	0.8879	0.6837
I.10	LLHN	0.8314	0.7276	0.8161	0.9411	0.9195	0.9056	0.7616	0.7822	0.6828
I.11.1	IA1	0.8780	0.8481	0.8250	0.9559	0.9302	0.9061	0.8227	0.9400	0.6855
I.11.2	IA2	0.8762	0.8450	0.8237	0.9561	0.9297	0.9061	0.8216	0.9418	0.6853
I.12.1	LNB-AA	0.8765	0.8442	0.8249	0.9348	0.9286	0.6335	0.8229	0.9399	0.6854
I.12.2	LNB-CN	0.8717	0.8408	0.8245	0.9334	0.9265	0.6320	0.8221	0.9334	0.6854
I.12.3	LNB-RA	0.8772	0.8448	0.8246	0.9354	0.9295	0.6350	0.8223	0.9435	0.6853
I.13.1	CAR-AA	0.7223	0.7129	0.6593	0.8832	0.8254	0.7554	0.6662	0.8971	0.5792
I.13.2	CAR-CN	0.7215	0.7118	0.6592	0.8825	0.8251	0.7554	0.6660	0.8955	0.5792
I.13.3	CAR-RA	0.7230	0.7140	0.6593	0.8838	0.8255	0.7554	0.6664	0.9001	0.5792
I.14	LIT	0.8615	0.7970	0.8228	0.9504	0.9283	0.9059	0.7842	0.9099	0.6841
I.15	FSW	0.8515	0.7794	0.8217	0.9486	0.9268	0.9059	0.7795	0.9039	0.6841
I.16	SAM	0.8665	0.7905	0.8197	0.9496	0.9276	0.9059	0.7839	0.8863	0.6835
I.17	CCLP	0.8726	0.8451	0.8206	0.9510	0.9230	0.8537	0.8196	0.9291	0.6778
I.18	TRA	0.8783	0.8549	0.8255	0.9567	0.9284	0.9060	0.8270	0.9384	0.6860
I.19	CNPA	0.8629	0.8389	0.8620	0.9579	0.9196	0.9140	0.8646	0.9185	0.7977
I.20	CN2D	0.8610	0.8113	0.8223	0.9510	0.9251	0.9057	0.8037	0.9229	0.6844
I.21	ADP	0.8801	0.8476	0.8251	0.9560	0.9305	0.9059	0.8228	**0.9439**	0.6855
I.22	NLC	0.8691	0.8503	0.8114	0.9507	0.9192	0.8424	0.8198	0.9303	0.6731
I.23	LAS	0.8665	0.7905	0.8197	0.9495	0.9276	0.9059	0.7839	0.8863	0.6835
I.24	LNL	0.8783	0.8549	0.8255	0.9566	0.9284	0.9060	0.8270	0.9384	0.6860
II.01	NSP	0.8455	0.7443	0.8599	0.9423	0.9121	0.9142	0.8047	0.8135	0.7887
II.02	KI	0.8946	0.8507	0.8860	0.9630	0.9528	0.9147	0.8634	0.9180	0.8044
II.03	GLHNI	0.8553	0.7250	0.8646	0.9485	0.9517	0.9146	0.6956	0.7321	0.7701
II.04	RW	0.8691	0.7598	0.8474	0.9290	0.9282	0.9147	0.8529	0.8614	0.7745
II.05	RWR	**0.9234**	**0.8943**	0.8955	**0.9691**	**0.9630**	0.9151	**0.8963**	0.9384	0.8095
II.06	FP	0.8124	0.7126	0.7798	0.8509	0.8223	0.9072	0.7746	0.8672	0.7601
II.07	SR	0.8707	0.7657	0.8708	0.9503	0.9531	0.9146	0.7848	0.7939	0.7403
II.08	PLM	0.8908	0.8480	0.8947	0.9479	0.9439	0.5161	0.8368	0.9407	0.8193
II.09	ACT	0.7456	0.7370	0.7822	0.8786	0.7969	0.9122	0.7913	0.8751	0.7686
II.10	RFK	0.8984	0.8614	0.8856	0.9595	0.9549	0.9150	0.8511	0.9106	0.7964
II.11	LDI	0.8627	0.8517	**0.8968**	0.9342	0.9129	0.5210	0.8533	0.8957	**0.8352**
II.12	NGLI	0.7596	0.7773	0.8129	0.9014	0.7982	0.9123	0.8483	0.8920	0.7907
II.13	RPR	0.8829	0.8462	0.8871	0.9642	0.9577	0.9150	0.8293	0.8946	0.8012
III.01	LPI	0.8962	0.8521	0.8898	0.9660	0.9527	**0.9160**	0.8690	0.9219	0.8182
III.02	LRW	0.9055	0.8561	0.8862	0.9659	0.9582	0.9157	0.8673	0.9236	0.8011
III.03	SRW	0.9055	0.8560	0.8862	0.9658	0.9582	0.9157	0.8672	0.9236	0.8011
III.04	CND	0.8895	0.8316	0.8708	0.9612	0.9492	0.9149	0.8334	0.9191	0.7904
III.05	CNC	0.8895	0.8316	0.8708	0.9464	0.9492	0.8864	0.8334	0.9188	0.7902

Dans le tableau IV-10, on présente la fréquence d'apparition de chaque méthode, en première, deuxième, et troisième position, ainsi que parmi les cinq premières, pour l'ensemble des réseaux, selon l'AUC moyen.

Ce tableau met en évidence que les méthodes locales dominent le top 5, présentant un AUC moyen plus élevé avec 27 occurrences, tandis que les méthodes globales apparaissent 18

fois, équitablement. Les méthodes Quasi-Locale ne figurent pas dans le classement de top 5 quel que soit le réseau utilisé.

Tableau IV-10 : Le nombre de fois que chaque méthode apparaît dans le top cinq pour tous les réseaux en fonction de l'AUC moyen

N°	Méthode	Première	deuxième	troisième	Top 5
I.10	LLHN		1		1
I.12.1	LNB-AA				1
I.12.2	LNB-CN			1	
I.12.3	LNB-RA				1
I.13.1	CAR-AA		4	1	2
I.13.2	CAR-CN	5		1	1
I.13.3	CAR-RA			4	2
I.17	CCLP				1
I.22	NLC				1
II.01	NSP				1
II.03	GLHNI	1			2
II.04	RW				1
II.06	FP	1	1	1	1
II.07	SR			1	
II.08	PLM	1			
II.09	ACT	1	1		2
II.11	LDI		1		
II.12	NGLI		1		1

VI.4. Conclusion

Dans ce chapitre, nous avons présenté une approche efficace pour la prédiction de liens dans les grands réseaux complexes qui part de l'ensemble des composantes connexes d'un réseau au lieu du réseau complet lui-même. L'idée clé de notre proposition est que dans tous les algorithmes basés sur des mesures locales ou sur le chemin, il suffit d'évaluer les liens reliant les nœuds internes d'une même composante.

Nous avons montré que cela nous permet d'économiser considérablement en termes de temps d'exécution en réduisant le nombre de calculs nécessaires.

De plus, nous avons également montré que l'utilisation de la décomposition permet la parallélisation du processus de prédiction de liens en répartissant les composantes sur plusieurs processeurs. Cela peut nous apporter un gain supplémentaire en termes de temps d'exécution, en particulier pour les grands réseaux ayant un grand nombre de composantes et où la répartition des nœuds sur les composantes est homogène.

Dans ce chapitre, nous avons présenté une comparaison objective basée sur des mesures d'évaluation rigoureuses, tel que le temps de calcul, la précision moyenne et l'AUC moyen. Enfin, la variété des méthodes dans le haut du classement montre que la performance de chaque technique dépend fortement des propriétés structurelles du réseau. Ceci souligne l'importance d'analyser les propriétés du réseau avant de choisir une technique de prédiction de lien particulière.

Conclusion Générale

Nous nous sommes intéressés dans cette thèse à l'analyse des réseaux complexes modélisés par des graphes non orientés et plus particulièrement à l'exploration des réseaux complexes pour la prédiction de liens.

Nous avons commencé cette thèse par une étude approfondie de l'état de l'art sur les réseaux complexes, de ce qui a été fait dans ce domaine de recherche scientifique, les différents types des réseaux, ainsi que leurs représentations graphiques et algorithmiques, avec les propriétés importantes. Ensuite nous avons présenté et analysé différentes méthodes destinées à la tâche de prédiction de lien qui sont classées en approches locales, globales et quasi-locales. En nous appuyant sur cette étude critique de l'état de l'art, nous avons proposé une nouvelle approche très efficace qui nous permet d'optimiser les calculs des scores pour les futurs liens prédits en termes de temps d'exécution. Cette approche nous permet d'améliorer la démarche de prédiction de liens en accélérant significativement l'accès aux résultats finaux.

L'approche proposée est basée sur la décomposition du graphe en composantes connexes. L'idée maîtresse de notre proposition est que dans tous les algorithmes basés sur des mesures locales et sur le chemin, il suffit d'évaluer les liens reliant les nœuds internes d'un même composant. En nous appuyant sur cette idée, nous nous avons proposé dans cette thèse une nouvelle architecture pour notre approche composée de six principales étapes à suivre pour réaliser la prédiction de liens. Cette architecture introduit notamment les notions de décomposition et du parallélisme. On peut résumer les principaux apports de cette architecture dans les points suivants :

- Nous avons pu minimiser le nombre de liens qui nécessitent le calcul d'un score. En effet, on a pu minimiser ce nombre jusqu'à 93%. Ce gain est d'autant plus important pour les bases de données de grande taille et ayant plusieurs composants connexes. Nous avons formellement étudié ce gain, ce qui nous a permis de confirmer que plus la répartition des nœuds dans les composantes connexes du réseau est homogène, plus le gain est important.
- Nous avons montré à l'aide d'une étude expérimentale rigoureuse que cela permet de gagner considérablement en temps d'exécution nécessaire pour la prédiction de liens.
- Nous avons montré que l'utilisation de la décomposition rend possible la parallélisation du processus de prédiction de liens en répartissant les composants sur plusieurs processeurs. Cela peut nous apporter un gain supplémentaire en termes de temps d'exécution.
- Enfin, la décomposition nous permet d'éviter les erreurs liées à la gestion de mémoire à cause de la taille énorme des bases de données. Au lieu de travailler sur

la totalité de la base, on travaille séparément sur les composantes qui sont généralement de taille nettement plus réduite que le réseau complet.

Pour évaluer la performance de notre approche, nous avons proposé deux types de comparaisons : descriptive et objective dans le but de trouver l'approche optimale pour une implémentation quelconque de méthode de similarité. Ces études comparatives sont effectuées avec l'utilisation des 22 types de base de données représentant différents réseaux complexes du monde réel, provenant de différentes sources et domaines d'application.

La comparaison descriptive consiste en l'utilisation d'outils mathématiques pour évaluer théoriquement le gain attendu de l'exploitation de la décomposition dans les algorithmes de prédiction de liens. Ce gain est exprimé en termes du nombre de scores calculés et il est mesuré par rapport à la distribution des nœuds sur les composants connexes à l'aide des notions de variance et de l'indice Gini. La comparaison objective est basée sur la mise en œuvre de cinq scénarios ou variantes d'algorithmes différents. La comparaison de ces variantes a clairement montré que : d'un côté la décomposition a un impact positif très important sur le temps de calcul puisque les variantes utilisant la décomposition sont largement plus rapides que l'algorithme de base qui utilise le réseau complet ; et d'un autre côté un gain additionnel en termes de temps d'exécution est obtenu par le traitement parallèle rendu possible également grâce à la décomposition.

Ce travail, nous ouvrent de nombreuses perspectives de travaux futurs liés à la problématique de la prédiction de liens :

- Nous souhaitons développer une nouvelle mesure de prédiction de liens, basée sur le renforcement des mesures à base de voisinage par des informations d'ordre global.
- Nous envisageons également le développement d'une méthode de combinaison d'informations locales et globales pour l'évaluation des liens inter-composantes qui sont complètement négligés dans les méthodes locales et les méthodes basées sur le chemin, mais qui peuvent avoir un sens dans certains contextes pratiques.
- Nous prévoyons de développer une bibliothèque de prédiction rapide de liens basée sur notre idée de décomposition du réseau et incluant toutes les informations locales et les indices de similarité basés sur le chemin.
- Un autre travail futur est d'étudier la question de l'auto-paramétrage des fonctions de similarité en fonction de la topologie des réseaux complexes

Références

Adamic, L.A., Adar, E., 2003. Friends and neighbours on web. J Soc Netw 25, 211–230.

Ahmad, I., Akhtar, M.U., Noor, S., Shahnaz, A., 2020. Missing Link Prediction using Common Neighbor and Centrality based Parameterized Algorithm. Sci Rep 10, 1–9.

Aouay, S., Jamoussi, S., Gargouri, F., 2014. Feature based link prediction, in: 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA). Presented at the 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), pp. 523–527. https://doi.org/10.1109/AICCSA.2014.7073243

Ashraf, A., Budka, M., Musial, K., 2018. Newton's Gravitational Law for Link Prediction in Social Networks. pp. 93–104. https://doi.org/10.1007/978-3-319-72150-7_8

Bai, S., Li, L., Cheng, J., Xu, S., Chen, X., 2018. Predicting Missing Links Based on a New Triangle Structure. Complexity 2018, e7312603. https://doi.org/10.1155/2018/7312603

Barabási, A.-L., Albert, R., 1999. Emergence of Scaling in Random Networks. Science 286, 509–512. https://doi.org/10.1126/science.286.5439.509

Barabási, A.-L., Oltvai, Z.N., 2004. Network biology: understanding the cell's functional organization. Nat. Rev. Genet. 5, 101–113. https://doi.org/10.1038/nrg1272

Batagelj, V., Mrvar, A., 2006. Pajek Datasets. Httpvladofmfuni-Ljsipubnetworksdata.

Blondel, V.D., Gajardo, A., Heymans, M., Senellart, P., Van Dooren, P., 2004. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. SIAM Rev. 46, 647–666.

Boguña, M., Pastor-Satorras, R., Diaz-Guilera, A., Arenas, A., 2004. Models of social networks based on social distance attachment. Phys. Rev. E 70, 056122.

Brandes, U., 2001. A faster algorithm for betweenness centrality. J. Math. Sociol. 25, 163–177. https://doi.org/10.1080/0022250X.2001.9990249

Breiman, L., Friedman, J., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees, The Wadsworth Statistics and Probability Series. Wadsworth Int. Group Belmont Calif. 9.

Burda, Z., Duda, J., Luck, J.-M., Waclaw, B., 2009. Localization of the maximal entropy random walk. Phys. Rev. Lett. 102, 160602.

Cannistraci, C., Alanis-Lobato, G., Ravasi, T., 2013. From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. Sci. Rep. https://doi.org/10.1038/srep01613

Cao, S., Lu, W., Xu, Q., 2016. Deep Neural Networks for Learning Graph Representations. Proc. AAAI Conf. Artif. Intell. 30. https://doi.org/10.1609/aaai.v30i1.10179

Cao, S., Lu, W., Xu, Q., 2015. GraRep: Learning Graph Representations with Global Structural Information. Proc. 24th ACM Int. Conf. Inf. Knowl. Manag. 891–900. https://doi.org/10.1145/2806416.2806512

Chakrabarty, B., Parekh, N., 2016. NAPS: Network Analysis of Protein Structures. Nucleic Acids Res. 44, W375–W382. https://doi.org/10.1093/nar/gkw383

Chebotarev, P., Shamis, E., 2006. The matrix-forest theorem and measuring relations in small social groups. ArXiv Prepr. Math0602070.

Chen, J., Wang, X., Xu, X., 2021. GC-LSTM: Graph Convolution Embedded LSTM for Dynamic Link Prediction. https://doi.org/10.48550/arXiv.1812.04206

Chua, H.N., Sung, W.-K., Wong, L., 2006. Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions. Bioinformatics 22, 1623–1630. https://doi.org/10.1093/bioinformatics/btl145

Clauset, A., Moore, C., Newman, M.E.J., 2008. Hierarchical structure and the prediction of missing links in networks. Nature 453, 98–101. https://doi.org/10.1038/nature06830

D. Bu, L.C. et al, Y. Zhao, 2003. Topological structure analysis of the protein-protein interaction network in budding yeast. Nucleic Acids Res 31, 2443–2450.

Daminelli, S., Thomas, J.M., Durán, C., Cannistraci, C.V., 2015. Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks. New J Phys 17, 113037.

Daniya, T., Geetha, M., Dr, S.K.K., 2020. Classification and regression trees with gini index. Adv. Math. Sci. J. 1857–8438.

Do, K., Tran, T., Venkatesh, S., 2018. Graph Transformation Policy Network for Chemical Reaction Prediction. https://doi.org/10.48550/arXiv.1812.09441

Dong, Y., Ke, Q., Wang, B., Wu, B., 2011. Link Prediction Based on Local Information, in: 2011 International Conference on Advances in Social Networks Analysis and Mining. pp. 382–386. https://doi.org/10.1109/ASONAM.2011.43

Dorogovtsev, S.N., Mendes, J.F., 2002. Evolution of networks. Adv Phys 51, 1079–1187.

Erdos, P., Rényi, A., 1960. On the evolution of random graphs. Publ Math Inst Hung Acad Sci 5, 17–60.

Estrada, E., 2012. The structure of complex networks: theory and applications. Oxford University Press.

Fouss, F., Pirotte, A., Renders, J., Saerens, M., 2007. Random-Walk Computation of Similarities between Nodes of a Graph with Application to Collaborative Recommendation. IEEE Trans. Knowl. Data Eng. 19, 355–369. https://doi.org/10.1109/TKDE.2007.46

Freeman, L.C., 1978. Centrality in social networks conceptual clarification. Soc. Netw. 1, 215–239. https://doi.org/10.1016/0378-8733(78)90021-7

Girvan, M., Newman, M.E.J., 2002. Community structure in social and biological networks. Proc. Natl. Acad. Sci. 99, 7821–7826. https://doi.org/10.1073/pnas.122653799

G.Jeh, Widom, J., 2002. SimRank:a measure of structural-context similarity, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 271–279.

Grandjean, M., 2014. La connaissance est un réseau. Perspective sur l'organisation archivistique et encyclopédique. Cah. Numér. 10, 37–54. https://doi.org/10.3166/LCN.10.3.37-54

Grover, A., Leskovec, J., 2016. node2vec: Scalable Feature Learning for Networks. https://doi.org/10.48550/arXiv.1607.00653

Gu, W., Gao, F., Lou, X., Zhang, J., 2019. Link Prediction via Graph Attention Network. https://doi.org/10.48550/arXiv.1910.04807

Guimera, R., Danon, L., Diaz-Guilera, A., Giralt, F., Arenas, A., 2003. Self-similar community structure in a network of human interactions. Phys. Rev. E 68, 065103(R).

Guns, R., 2014. Link Prediction, in: Ding, Y., Rousseau, R., Wolfram, D. (Eds.), Measuring Scholarly Impact: Methods and Practice. Springer International Publishing, Cham, pp. 35–55. https://doi.org/10.1007/978-3-319-10377-8_2

Guns, R., Rousseau, R., 2014. Recommending research collaborations using link prediction and random forest classifiers. Scientometrics 101, 1461–1473. https://doi.org/10.1007/s11192-013-1228-9

Hasan, M.A.A., Zaki, M.J., 2011. A Survey of Link Prediction in Social Networks. Soc. Netw. Data Anal. 2011 243–275.

Heckerman, D., Meek, C., Koller, D., 2004. Probabilistic Entity-Relationship Models, PRMs, and Plate Models.

Ibrahim, N.M.A., Chen, L., 2015. Link prediction in dynamic social networks by integrating different types of information. Appl Intell 42, 738–750.

Isella, L., Stehlé, J., Barrat, A., Cattuto, C., Pinton, J.F., Broeck, W.V. d, 2011. What's\ in a crowd? Analysis of face-to-face behavioral networks. J Theor Biol 271, 166–180.

Jaccard, P., 1901. Etude comparative de la distribution florale une portion des Alpes et du Jura. Bull Soc Vaud Sci Nat 37, 547–579.

Kamath, P.S., Wiesner, R.H., Malinchoc, M., Kremers, W., Therneau, T.M., Kosberg, C.L., D'Amico, G., Dickson, E.R., Kim, W.R., 2001. A model to predict survival in patients with end-stage liver disease. Hepatology 33, 464–470.

Kastrin, A., Rindflesch, T.C., Hristovski, D., 2016. Link Prediction on a Network of Co-occurring MeSH Terms: Towards Literature-based Discovery. Methods Inf. Med. 55, 340–346. https://doi.org/10.3414/ME15-01-0108

Katz, L., 1953. A new status index derived from sociometric analysis. Psychometrika 18, 39–43.

Kipf, T.N., Welling, M., 2017. Semi-Supervised Classification with Graph Convolutional Networks. https://doi.org/10.48550/arXiv.1609.02907

Kipf, T.N., Welling, M., 2016. Variational Graph Auto-Encoders. https://doi.org/10.48550/arXiv.1611.07308

Kohavi, R., Provost, F., 1998. Glossary of terms. Special issue of applications of machine learning and the knowledge discovery process. Mach Learn 30.

Kumar, A., Singh, S.S., Singh, K., Biswas, B., 2020. Link prediction techniques, applications, and performance: A survey. Phys. Stat. Mech. Its Appl. 553, 124289.

Kunegis, J., 2013. KONECT - The Koblenz Network Collection, in: Proc. Int. Conf. on World Wide Web Companion. pp. 1343–1350.

Laishram, R., 2015. Link Prediction in Dynamic Weighted and Directed Social Network using Supervised Learning.

Lei, K., Qin, M., Bai, B., Zhang, G., Yang, M., 2019. GCN-GAN: A Non-linear Temporal Link Prediction Model for Weighted Dynamic Networks. IEEE INFOCOM 2019 - IEEE Conf. Comput. Commun. 388–396. https://doi.org/10.1109/INFOCOM.2019.8737631

Leicht, E.A., P.Holme, Newman, M.E.J., 2006. Vertex similarity in networks. Phys Rev 73, 026120.

Leskovec, J., Kleinberg, J., Faloutsos, C., 2007. Graph evolution: densification and shrinking diameters. ACM Trans Knowl Discov Data TKDD 1, 2-es.

Li, S., Huang, J., Liu, J., Huang, T., Chen, H., 2020. Relative-path-based algorithm for link prediction on complex networks using a basic similarity factor. Chaos Interdiscip. J. Nonlinear Sci. 30, 013104.

Liben-Nowell, D., 2005. An algorithmic approach to social networks (PhD Thesis). Massachusetts Institute of Technology.

Liben-Nowell, D., Kleinberg, J., 2007a. The link-prediction problem for social networks. J. Am. Soc. Inf. Sci. Technol. 58, 1019–1031.

Liben-Nowell, D., Kleinberg, J., 2007b. The link-prediction problem for social networks. J. Am. Soc. Inf. Sci. Technol. 58, 1019–1031.

Lichtenwalter, R., Lussier, J., Chawla, N., 2010. New perspectives and methods in link prediction. Presented at the Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 243–252. https://doi.org/10.1145/1835804.1835837

Liu, G., Li, J., Wong, L., 2008. Assessing and Predicting Protein Interactions Using Both Local and Global Network Topological Metrics. Genome Inform. 21, 138–149. https://doi.org/10.11234/gi1990.21.138

Liu, W., Lü, L., 2010. Link prediction based on local random walk. EPL Europhys. Lett. 89, 58007. https://doi.org/10.1209/0295-5075/89/58007

Liu, Z., Zhang, Q.-M., Lü, L., Zhou, T., 2011. Link prediction in complex networks: A local naïve Bayes model. EPL Europhys. Lett. 96, 48007. https://doi.org/10.1209/0295-5075/96/48007

Lu, L., Jin, C.H., Zhou, T., 2009. Similarity index based on local paths for link prediction of complex networks. Phys. Rev. E 80, 046122.

Lü, L., Zhou, T., 2011. Link prediction in complex networks: A survey. Phys Stat Mech Its Appl 390, 1150–1170.

Marsden, P., 2006. Generalized Blockmodeling, P. Doreian, V. Batagelj, A. Ferligoj. Cambridge University Press, New York (2005), (xv + 384 pp). Soc. Netw. 28, 275–282. https://doi.org/10.1016/j.socnet.2006.02.001

Martínez, V., Berzal, F., Cubero, J.-C., 2016a. Adaptive degree penalization for link prediction. J. Comput. Sci. 13, 1–9.

Martínez, V., Berzal, F., Cubero, J.-C., 2016b. A survey of link prediction in complex networks. ACM Comput. Surv. CSUR 49, 1–33.

McAuley, J., Leskovec, J., 2012. Learning to Discover Social Circles in Ego Networks, in: Advances in Neural Information Processing Systems. pp. 548–556.

McCune, B., Grace, J.B., Urban, D.L., 2002. Analysis of ecological communities. MjM Software Design, Gleneden Beach.

Menon, A., Elkan, C., 2011. Link Prediction via Matrix Factorization, Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II. https://doi.org/10.1007/978-3-642-23783-6_28

Mitzenmacher, M., 2004. A Brief History of Generative Models for Power Law and Lognormal Distributions. Internet Math. 1. https://doi.org/10.1080/15427951.2004.10129088

Mumin, D., Shi, L.-L., Liu, L., 2022. An efficient algorithm for link prediction based on local information: Considering the effect of node degree. Concurr. Comput. Pract. Exp. 34, e6289. https://doi.org/10.1002/cpe.6289

Newman, M.E., 2006. Finding community structure in networks using the eigenvectors of matrices. Phys Rev E 74, 036104.

Newman, M.E., 2003. The structure and function of complex networks. SIAM Rev 45, 167–256.

Newman, M.E., 2001. The structure of scientific collaboration networks. Proc. Natl. Acad. Sci. 98, 404–409.

Newman, M.E.J., 2010. Networks: An Introduction. https://doi.org/10.1093/acprof:oso/9780199206650.001.0001

Ou, M., Cui, P., Pei, J., Zhang, Z., Zhu, W., 2016. Asymmetric transitivity preserving graph embedding, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1105–1114.

Pachaury, S., Kumar, N., Khanduri, A., Mittal, H., 2018. Link Prediction Method Using Topological Features and Ensemble Model, in: 2018 Eleventh International Conference on Contemporary Computing (IC3). Presented at the 2018 Eleventh International Conference on Contemporary Computing (IC3), pp. 1–6. https://doi.org/10.1109/IC3.2018.8530624

Papadimitriou, A., Symeonidis, P., Manolopoulos, Y., 2012. Fast and accurate link prediction in social networking systems. J. Syst. Softw. 85, 2119–2132.

Perozzi, B., Al-Rfou, R., Skiena, S., 2014. DeepWalk: Online Learning of Social Representations [WWW Document]. arXiv.org. https://doi.org/10.1145/2623330.2623732

P.Massa, Salvetti, M., Tomasoni, D., 2009. Bowling alone and trust decline in social network sites, in: Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC'09). pp. 658–663.

Rattigan, M.J., Jensen, D., 2005. The case for anomalous link discovery. ACM SIGKDD Explor. Newsl. 7, 41.

Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N., Barabasi, A.L., 2002. Hierarchical organization of modularity in metabolic networks. Science 297, 1551–1555.

Reza, Z., Huan, L., 2009. Social Computing Data Repository at ASU, Arizona State University, School of Computing, Informatics and Decision Systems Engineering.

Ribeiro, L.F.R., Savarese, P.H.P., Figueiredo, D.R., 2017. struc2vec: Learning Node Representations from Structural Identity, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 385–394. https://doi.org/10.1145/3097983.3098061

S. Peri, R.A., J.D. Navarro et al., 2003. Development of human protein reference database as an initial platform for approaching systems biology in humans. Genome Res 13, 2363–2371.

Sabidussi, G., 1966. The centrality index of a graph. Psychometrika 31, 581–603. https://doi.org/10.1007/BF02289527

Samad, A., Qadir, M., Nawaz, I., 2019. SAM: a Similarity Measure for Link Prediction in Social Network, in: 13th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS'2019). pp. 1–9.

Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G., 2009. The Graph Neural Network Model. IEEE Trans. Neural Netw. 20, 61–80. https://doi.org/10.1109/TNN.2008.2005605

Scott, J., 2017. Social Network Analysis. SAGE.

Scripps, J., Tan, P.N., Esfahanian, A.H., 2009. Measuring the effects of preprocessing decisions and network forces in dynamic network analysis, in: In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 747–756.

Shibata, N., Kajikawa, Y., Sakata, I., 2012. Link prediction in citation networks. J. Am. Soc. Inf. Sci. Technol. 63, 78–85.

Sorensen, T., 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. København, I kommission hos E. Munksgaard.

Spielman, D., 2012. Spectral graph theory. Comb. Sci. Comput. 18.

Sui, X., Lee, T., Whang, J.J., Savas, B., Jain, S., Pingali, K., Dhillon, I., 2013. Parallel clustered low-rank approximation of graphs and its application to link prediction, in: International Workshop on Languages and Compilers for Parallel Computing. pp. 76–95.

Sun, Q., Hu, R., Yang, Z., Yao, Y., Yang, F., 2017. An improved link prediction algorithm based on degrees and similarities of nodes, in: 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS). Presented at the 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), IEEE, Wuhan, China, pp. 13–18. https://doi.org/10.1109/ICIS.2017.7959962

Tan, F., Xia, Y., Zhu, B., 2014. Link Prediction in Complex Networks: A Mutual Information Perspective. CoRR abs/1405.4341.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q., 2015. LINE: Large-scale Information Network Embedding, in: Proceedings of the 24th International Conference on World Wide Web. pp. 1067–1077. https://doi.org/10.1145/2736277.2741093

Taskar, B., Abbeel, P., Koller, D., 2012. Discriminative Probabilistic Models for Relational Data. https://doi.org/10.48550/arXiv.1301.0604

Tong, H., Faloutsos, C., Pan, J.-Y., 2006. Fast random walk with restart and its applications, in: Sixth International Conference on Data Mining (ICDM'06). IEEE, pp. 613–622.

Valverde-Rebaza, J., Valejo, A.D., Berton, L., Faleiros, T., Lopes, A., 2015. A Naïve Bayes model based on overlapping groups for link prediction in online social networks. https://doi.org/10.1145/2695664.2695719

Vanunu, O., Sharan, R., 2008. A propagation-based algorithm for inferring gene-disease associations, in: German Conference on Bioinformatics. Gesellschaft für Informatik e. V.

Wang, C., 2016. Zhu, 2016 Wang D., Cui P., Zhu W., Structural deep network embedding, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, ca, Usa. pp. 1225–1234.

Wang, H., Le, Z., 2020. Seven-layer model in complex networks link prediction: A survey. Sensors 20, 6560.

Wang, H., Wang, Jia, Wang, Jialin, Zhao, M., Zhang, W., Zhang, F., Xie, X., Guo, M., 2018. GraphGAN: Graph Representation Learning With Generative Adversarial Nets. Proc. AAAI Conf. Artif. Intell. 32. https://doi.org/10.1609/aaai.v32i1.11872

Wang, T., He, X.S., Zhou, M.Y., Z.-Q, Z.Q.F., 2017. Link prediction in evolving networks based on popularity of nodes. Sci Rep 7, 1–10.

Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of small-world'networks. Nature 393, 440–442.

Wu, J., 2018. A generalized tree augmented naive Bayes link prediction model. J Comput Sci 27, 206–217.

Wu, Z., Lin, Y., Wan, H., Jamil, W., 2016a. Predicting top-L missing links with node and link clustering information in large-scale networks. J. Stat. Mech. Theory Exp. 2016, 083202.

Wu, Z., Lin, Y., Wang, J., Gregory, S., 2016b. Link prediction with node clustering coefficient. Phys. Stat. Mech. Its Appl. 452, 1–8.

Xiao, X., Zhang, Z., 2015. Web-Age Information Management. Springer.

Yang, J., Yang, L., Zhang, P., 2015. A New Link Prediction Algorithm Based on Local Links, in: Xiao, X., Zhang, Z. (Eds.), Web-Age Information Management, Lecture Notes in Computer Science. Springer International Publishing, Cham, pp. 16–28. https://doi.org/10.1007/978-3-319-23531-8_2

Yang, J., Zhang, X.D., 2016. Predicting missing links in complex networks based on common neighbors and distance. Sci Rep 6, 1–10.

Yang, X.-H., Yang, X., Ling, F., Zhang, H.-F., Zhang, D., Xiao, J., 2018. Link prediction based on local major path degree. Mod. Phys. Lett. B 32, 1850348.

Ying, D., Ronald, R., Dietmar, W., 2014. Measuring scholarly impact : methods and practice. Springer.

Yu, B., Chen, C., Wang, X., Yu, Z., Ma, A., Liu, B., 2021. Prediction of protein–protein interactions based on elastic net and deep forest. Expert Syst. Appl. 176, 114876. https://doi.org/10.1016/j.eswa.2021.114876

Yu, K., Chu, W., Yu, S., Tresp, V., Xu, Z., 2006. Stochastic Relational Models for Discriminative Link Prediction, in: Advances in Neural Information Processing Systems. MIT Press.

Zaki, M.J., Meira, W., 2014. Data Mining and Analysis: Fundamental Concepts and Algorithms. Cambridge University Press.

Zeng, S., 2016. Link prediction based on local information considering preferential attachment. Phys. Stat. Mech. Its Appl. 443, 537–542. https://doi.org/10.1016/j.physa.2015.10.016

Zhang, J., Zhang, Y., Yang, H., Yang, J., 2014. A link prediction algorithm based on socialized semi-local information. J. Comput. Inf. Syst. 10, 4459–4466. https://doi.org/10.12733/jcis10454

Zhou, J., Huang, D., Wang, H., 2017. A dynamic logistic regression for network link prediction. Sci. China Math. 60, 165–176. https://doi.org/10.1007/s11425-015-0807-8

Zhou, T., 2021. Progresses and challenges in link prediction. iScience 24, 103217. https://doi.org/10.1016/j.isci.2021.103217

Zhou, T., Lü, L., Y.C.Zhang, 2009. Predicting missing links via local information. Eur Phys J B71, 623–630.