Mesures de proximité appliquées à la détection de communautés dans les grands graphes de terrain

Maximilien Danisch

UPMC
SORBONNE UNIVERSITÉS

THÈSE

dirigée par Jean-Loup Guillaume et Bénédicte Le Grand

présentée pour obtenir le grade de

DOCTEUR EN SCIENCES
DE L'UNIVERSITÉ PIERRE ET MARIE CURIE

spécialité Informatique

Mesures de proximité appliquées à la détection de communautés dans les grands graphes de terrain

---

Maximilien DANISCH

Soutenue publiquement le 15 Juin 2015 devant le jury composé de

Rapporteurs :	David Auber	Maître de Conférences HDR, Université de Bordeaux
	Céline Robardet	Professeur, Université de Lyon
Examinateurs :	Alain Barrat	Directeur de Recherche, CNRS
	Rushed Kanawati	Maître de Conférences, Université Paris Nord
	Matthieu Latapy	Directeur de Recherche, CNRS
	Bivas Mitra	Professeur, Indian Institute of Technology Kharagpur
Directeur :	Bénédicte Le Grand	Professeur, Université Panthéon-Sorbonne
Encadrant :	Jean-Loup Guillaume	Professeur, Université de La Rochelle

À ma mère,

Remerciements

Je tiens tout d'abord à remercier Jean-Loup Guillaume et Bénédicte Le Grand, qui m'ont encadré tout au long de ma thèse. Leur aide a été déterminante dans l'aboutissement de mes travaux. Leur façon de penser, leur gentillesse, leur disponibilité et leur force de travail en font des exemples à suivre pour moi. Je tiens à mettre en avant notre compatibilité à tous les trois, aussi bien sur le plan professionnel que sur le plan émotionnel. De cette osmose a pu naître un maximum de créativité et grâce à elle je me sens prêt à être un chercheur indépendant. Cette osmose est bien évidemment due à un effort d'adaptation de leur part et pour cela je les remercie. La fin de cette thèse n'est évidement pas la fin de notre amitié ni de notre collaboration.

Je tiens à exprimer toute ma gratitude à David Auber et à Céline Robardet qui ont accepté d'être les rapporteurs de mon travail, ainsi qu'à Alain Barrat, Rushed Kanawati, Matthieu Latapy et Bivas Mitra qui ont bien voulu faire partie de mon jury. Conscient du travail que cela représente, j'ai fait mon maximum pour préparer un manuscrit et une présentation clairs et intéressants.

La bonne humeur de l'équipe Complex Networks, sa dynamique et les discussions poussées que j'ai eues avec mes collègues (qu'elles soient scientifiques, personnelles, politiques ou métaphysiques) ont également été des ingrédients essentiels au succès de ma thèse. Je tiens ainsi à remercier la chef d'équipe Clémence Magnien et les autres permanents de l'équipe : Matthieu Latapy, Fabien Tarissan et Lionel Tabourier. Ainsi que toute la "work force" actuelle : Noé Gaumont, Jordan Viard, Thibaud Arnoux, Damien Nogues, Raphaël Tackx, Louisa Harutyunyan et Keun-Woo Lim. Et celle que j'ai croisée pendant ma thèse : Sergey Kirgizov, Qinna Wang, Daniel Bernardes, Alice Albano, Sébastien Heymann, Raphaël Fournier-S'niehotta, Romain Campigotto, François Queyroi, Yann Jacob, Oana Balalau, Aurélie Faure de Pebeyre, Amélie Medem, Elie Rotenberg, Emilie Coupechoux, Massoud Seifi et Thomas Aynaud. Pour les mêmes raisons, j'aimerais également remercier les membres de notre équipe voisine, l'équipe Phare, en particulier : Dallal Belabed, Fouad Guenane, Mario Zancanaro et Yacine Benchaib, ainsi que les membres de ma deuxième équipe, le CRI de Paris 1, notamment Ali Jaffal, Elena Epure, Danny Munera et Arnaud de Myttenaere.

Je remercie également Véronique Varenne pour sa bonne humeur et son don de faire paraître les tâches administratives simples.

Je tiens à remercier l'équipe CNeRG de l'IIT Kharaghpur en Inde particulièrement Soumajit Pramanik, Bivas Mitra et le chef d'équipe Niloy Ganguly qui m'ont accueilli très

chaleureusement. Grâce à eux, j'ai découvert une culture différente, notamment en ce qui concerne le travail, et j'en ressors extrêmement enrichi.

Je tiens également à remercier mes co-auteurs Raphaël Fournier-S'niehotta, Darko Obradovic, Nicolas Dugué et Anthony Perez, Raphaël Tackx et Fabien Tarissan, Qinna Wang, ainsi que Soumajit Pramanik et Bivas Mitra, nos discussions et notre bonne entente ont mené à des travaux publiés et j'espère bien renouveler l'expérience.

Magalie Guillaume a relu des parties de cette thèse et je l'en remercie. Jean-Pierre Mercier et Aurélie Faure de Pebeyre ont relu l'intégralité de ce manuscrit et cela durant les quelques jours restant avant la date butoir. Conscient que cela a été éprouvant, ce geste m'a beaucoup touché et je tiens à les remercier en gravant par ces quelques lignes la reconnaissance que je leur témoigne.

Ma professeur de mathématique de terminale Mme Martine Robillard m'a sorti d'un état de "pseudo-échec scolaire" en arrivant, grâce à sa passion pour l'enseignement, à m'inciter à travailler et cela a vraiment révélé mes capacités. À présent, j'aime ce que je fais et je me sens utile, tout cela c'est grâce à elle. Je lui serai éternellement reconnaissant et j'espère bien pouvoir reproduire ce qu'elle a fait pour moi avec mes étudiants. Je tiens également à remercier les enseignants de prépa qui m'ont marqué et ont fortement contribué à m'amener au niveau où je suis : M. Melin, M. Lewis, Mme Meunier et "last but not least" M. Sallen.

Je remercie aussi Hernan A. Makse et Tony Jebara avec qui j'ai effectué des stages longs de recherche avant ma thèse. En plus de me proposer des sujets très intéressants, ils ont pris le temps de partager avec moi leur vision de la recherche et de leur domaine respectif. Cela ma permis d'aborder la thèse avec plus de confiance et de recul.

Enfin, il m'est impossible de ne pas remercier ma mère, Irène Danisch, qui m'a élevé seule et m'a toujours soutenu. Je l'aime plus que tout et je tiens à l'écrire ici, car malheureusement je n'arrive pas toujours à le montrer. C'est elle qui m'a appris, lorsque j'étais encore petit, que c'est en tombant et en se relevant que l'on apprend. Cette idée a depuis façonné mon esprit et est au centre de tout ce que j'entreprends.

Bien que cela soit très étrange, l'univers semble exister. L'humanité semble infinitésimale comparée à sa grandeur et je suis infinitésimal comparé à celle de l'humanité. Ainsi, les contributions de cette thèse sont minuscules, j'en suis navré et je ne peux que remercier l'univers de me laisser y contribuer.

Table des matières

Chapitre 1

Introduction

1.1 Contexte

Les données sont partout. En effet, que ce soit en ligne ou hors ligne, chaque personne effectue des actions produisant des données qui peuvent être collectées. Les exemples sont innombrables : historique des pages web visitées, activité sur les réseaux sociaux (Facebook, Twitter...), opérations effectuées par carte bancaire, appels téléphoniques, déplacements en transport en commun, et même tout déplacement enregistrable par l'intermédiaire d'un smartphone. En plus de ces données générées par les activités quotidiennes d'un individu, il existe également d'autres types de données pouvant être collectées, qu'elles soient générées par des humains ou bien présentes dans la nature. Là encore, les exemples sont innombrables : activité neuronale d'un cerveau, ensemble des réactions chimiques à l'intérieur d'un corps humain, suites de mots dans un livre, réseau et trafic internet, réseau et trafic routier, rayonnements électromagnétiques provenant de l'espace, etc.

Avec le développement des ressources informatiques, certaines de ces données sont enregistrables et peuvent être traitées : elles peuvent être fouillées et des connaissances peuvent en être extraites. Grâce à ces données, on peut également faire de l'apprentissage et prédire des évènements. On peut aussi faire de la recommandation ou détecter des anomalies.

Cependant, lors du traitement de ces données par un algorithme, leur volume important constitue un problème, aussi bien pour leur stockage en mémoire que pour le temps de calcul de l'algorithme. C'est le phénomène connu sous le nom (ou le *buzzword*) de *big data*. Un autre problème trouve sa source dans le format varié de ces données : vecteurs d'attributs, textes, graphes, etc. C'est le phénomène connu sous le nom de *rich data*. Un formalisme intéressant pour tenter de pallier ces problèmes est la représentation (ou la conversion) de ces données sous la forme d'un graphe, c'est-à-dire d'un ensemble de nœuds connectés entre eux par des liens. Ces graphes s'avèrent, en général, être creux (ou clairsemés), c'est-à-dire qu'ils présentent un nombre de liens total dans le graphe petit par rapport au nombre de liens potentiel. Une grande partie des données citées précédemment sont d'ailleurs naturellement représentables selon ce formalisme. Typiquement, Facebook, où

des profils sont associés par des liens d'amitié, Internet où des ordinateurs sont liés par des connexions physiques, le cerveau où des neurones sont connectés par des synapses ou bien des protéines réagissant chimiquement entre elles. Remarquons que même si les données ne sont pas directement représentables sous la forme d'un graphe, on peut les y convertir : par exemple, on peut lier deux textes en fonction de leur similarité (évaluée par exemple via le nombre de mots-clefs qu'ils partagent). Le fait, observé empiriquement, que le nombre de liens soit faible par rapport au nombre total de liens pouvant exister permet de concevoir des algorithmes rapides, donnant une réponse instantanément ou en quelques jours. Notons que si les graphes étaient plus denses, le simple fait d'examiner chaque lien serait déjà trop long pour les plus grands graphes (par exemple Facebook, avec plus d'un milliard de profils, pourrait posséder de l'ordre d'un trillion de liens) avec les puissances de calcul actuelles.

Ces graphes extraits de données réelles sont appelés graphes de terrain ou réseaux complexes et ont souvent des propriétés similaires, par exemple :

— Ils sont creux globalement (comme nous l'avons vu précédemment), mais sont denses localement, c'est-à-dire qu'ils contiennent beaucoup de triangles. Dans les réseaux sociaux, ceci correspond au fait que "les amis de mes amis sont mes amis".
— La distance moyenne entre deux nœuds est faible, c'est-à-dire que le nombre moyen de liens que l'on doit suivre pour aller d'un nœud à un autre est peu élevé.
— La distribution du nombre de connexions qu'a un nœud est hétérogène : il existe ainsi des nœuds très connectés, appelés hubs, ainsi qu'une multitude de nœuds peu connectés.

L'explosion de la taille et du nombre de ces réseaux complexes a créé le besoin d'algorithmes rapides et génériques fonctionnant sur n'importe lequel de ces réseaux afin de les étudier. L'une des problématiques privilégiées liées à leur compréhension concerne leur organisation ; en effet, le nombre de nœuds étant très important, il est intéressant de rassembler ces noeuds dans des groupes afin de mieux comprendre la structure générale du réseau. Il s'agit en quelque sorte d'en faire une synthèse à gros grains (*coarse-graining*) afin d'obtenir une description mésoscopique du système. La *détection de communautés* est un premier pas vers cette description mésoscopique des réseaux complexes. Une communauté est généralement définie de manière informelle comme un groupe de nœuds fortement liés les uns aux autres et peu liés vers l'extérieur, ou plus généralement comme un groupe de nœuds qu'il est pertinent de réunir.

Une autre problématique est de savoir dans quelle mesure deux nœuds sont topologiquement proches. La distance (nombre de sauts en suivant des liens) est trop restrictive, puisqu'elle est toujours faible (comme nous l'avons vu précédemment) et ne prend que des valeurs entières ; elle n'est donc pas adaptée. L'élaboration de *mesures de proximité* plus complexes est donc nécessaire.

Ces deux notions structurelles de *communauté* et de *mesure de proximité* sont fortement intriquées et constituent le sujet principal de cette thèse.

1.2 Communauté et proximité : applications

Les applications directes de la détection de communautés et des mesures de proximité dans les réseaux sont innombrables. Nous citons ici les plus courantes.

Le premier grand groupe d'applications est bien évidement tout ce qui consiste en l'organisation des données, par exemple la constitution de listes d'amis sur les réseaux sociaux. La classification de documents est un autre exemple, en particulier s'il existe une structure en réseau intrinsèque, comme des documents incorporant des citations (e.g. des pages Wikipédia, des articles scientifiques, des articles de loi...), mais pas seulement. La structure en réseau peut également être fabriquée après analyse de texte.

La recommandation est également une application importante. Il peut s'agir de recommandation d'amis sur les réseaux sociaux ou de produits sur des sites de vente en ligne. Une problématique importante, notamment en ce qui concerne les moteurs de recherche, est de proposer des recommandations non seulement pertinentes, mais aussi variées. La détection de communautés et l'identification de pages importantes au sein de ces communautés y jouent de manière évidente un rôle central.

La prédiction, application très similaire à la recommandation, mais néanmoins intrinsèquement différente est la dernière application que nous présentons bien qu'elle ne soit pas, loin de là, l'application la moins importante et la moins lucrative. On peut prédire la fonction de protéines inconnues dans un réseau biologique : si une certaine protéine inconnue apparaît dans la communauté de protéines connues, alors ceci nous donne des éléments pour deviner sa fonction. On peut également prédire quels clients vont changer d'opérateur téléphonique : si la communauté d'un client d'intérêt (les personnes importantes pour le client) change d'opérateur téléphonique, alors on peut s'attendre à ce qu'il en change également. On peut également prédire des relations d'amitié sur les réseaux sociaux : si deux personnes appartiennent à la même communauté, mais ne sont pas amies, alors elles ont de grandes chances de le devenir prochainement. Il pourrait alors être utile de recommander un lien d'amitié à ces utilisateurs et c'est en cela que la prédiction rejoint la recommandation.

1.3 Idée générale de la thèse

Dans la littérature, la détection de communautés prend plusieurs formes ; on peut en effet chercher :

- une partition en communautés, où un nœud appartient à une communauté et une seule ;
- une structure en communautés recouvrantes, où un nœud peut appartenir à une, à plusieurs ou à aucune communauté ;
- la communauté, ou les communautés, d'un nœud en particulier.

Le premier problème, c'est-à-dire le partitionnement d'un réseau en une structure communautaire, a été très largement étudié ces dernières années et compte plusieurs centaines d'articles et donc de méthodes sur le sujet. Ceci est attesté, en partie, par l'article de

Fortunato présentant l'état de l'art sur les communautés (principalement vues comme une partition) en 2010 [64]. La méthode de Louvain citée plus de 2800 fois [32] en est un exemple. Ce problème de partitionnement ayant été bien exploré (bien qu'il suscite toujours l'intérêt comme en témoigne l'article récent [39]), nous nous concentrerons sur les deux autres, c'est-à-dire (i) celui de la couverture et (ii) celui de la détection des communautés d'un nœud en particulier. J'ajouterais que la recherche de communautés recouvrantes est plus naturelle que celle d'une partition, car un nœud appartient en général à plusieurs communautés que l'on qualifie alors de "recouvrantes". Par exemple, dans un réseau social, chaque personne appartient à plusieurs groupes : amis, collègues, familles. Ces deux derniers problèmes semblent donc plus conformes à la réalité du terrain. Nous travaillerons également sur un autre problème, un peu moins classique, qui est (iii) celui de la complétion d'un ensemble de nœuds en une communauté.

La résolution de ces problèmes est en très grande majorité effectuée, à l'heure actuelle, à travers l'optimisation d'une fonction de qualité prenant en entrée un graphe et un ensemble de nœuds dans ce graphe et retournant une valeur quantifiant la pertinence du groupe de nœuds en tant que communauté. L'optimisation est effectuée, en général, de manière gloutonne ou stochastique car ce sont des méthodes rapides et le temps est très souvent le facteur limitant dans les grands graphes. Bien que cette approche permette dans de nombreux cas d'obtenir un résultat satisfaisant, elle souffre de deux problèmes :

1. la conception d'une fonction de qualité est compliquée et aucune fonction proposée dans la littérature ne fait l'unanimité,
2. l'optimisation peut conduire à des minima locaux peu pertinents.

Nous proposons dans cette thèse une approche alternative qui permet d'éviter ces deux problèmes ; nous l'appelons **approche mesure de proximité** et elle est opposée à cette classique **approche fonction de qualité**. Son principe général de fonctionnement peut être décrit comme suit. Étant donné un nœud d'intérêt dans le graphe, on calcule la *proximité* de chaque nœud du graphe à ce nœud d'intérêt. Ensuite, si un petit groupe de nœuds obtient une proximité très élevée à ce nœud d'intérêt et que tous les autres nœuds du graphe ont une proximité très faible, alors on peut directement conclure que le petit groupe de nœuds est la communauté du nœud d'intérêt. Cette idée montre le lien fort qui existe entre mesure de proximité et structure communautaire et nous la déclinerons afin de proposer des solutions aux trois problèmes posés ci-dessus. Dans le cadre du suivi temporel des communautés identifiées, nous montrerons également que les communautés détectées à l'aide d'une approche à base de proximité sont en général faciles à suivre. En effet, contrairement à une approche fonction de qualité, l'approche mesure de proximité est déterministe, peu sensible aux perturbations et les communautés détectées sont caractérisées par un nœud (ou un petit ensemble de nœuds) ce qui permet de les étiqueter et donc de faciliter le suivi de leur évolution.

1.4 Plan du manuscrit

La suite de ce manuscrit est organisée de la façon suivante.

Partie I :

Nous commencerons, dans le chapitre 2, par un état de l'art sur la détection de communautés recouvrantes et de communautés centrées sur un nœud (dites ego-centrées). Puis, dans le chapitre 3, nous mettrons en évidence le lien fort entre structure communautaire et mesure de proximité et dresserons un état de l'art sur les mesures de proximité. Nous proposerons également dans ce chapitre deux nouvelles mesures de proximité, l'une sans paramètre et l'autre paramétrée.

Partie II :

Nous exploiterons ensuite ce lien pour proposer :
— dans le chapitre 4, un algorithme qui permet de trouver toutes les communautés auxquelles un nœud donné appartient ;
— dans le chapitre 5, un algorithme qui permet de compléter un ensemble de nœuds en une communauté ;
— dans le chapitre 6, un algorithme qui permet de trouver des communautés recouvrantes dans un réseau (et potentiellement toutes dans certains cas).

Partie III :

Les chapitres 7, 8 et 9 présentent des travaux réalisés en ouverture de la thèse dédiés respectivement à l'étude du capitalisme social sur Twitter, à une méthode pour générer des graphes respectant exactement une certaine distribution des degrés et à la détection de requêtes à caractère pédophile effectuées dans les réseaux pair-à-pair.

L'annexe A présente les jeux de données utilisés au cours de la thèse.

1.5 Validation des résultats

La validation des résultats que nous allons montrer est une étape essentielle, parfois négligée dans la littérature, et qui mérite d'être ici soulignée. Il s'agit de vérifier que les communautés détectées sont correctes, c'est-à-dire qu'elles correspondent à la réalité du terrain.

Nous axerons notre travail principalement sur la topologie des réseaux étudiés. Par exemple, l'un des jeux de données sur lequel nous allons travailler est le réseau Wikipédia : l'ensemble des pages du site web Wikipédia (constituant les nœuds du réseau) et les liens hypertextes entre elles. Bien sûr, les pages sont constituées de texte et l'analyse de ce texte peut être d'une grande aide pour le groupement de pages en catégories (ou communautés). Ce groupement de pages est d'ailleurs effectué manuellement par les utilisateurs eux-mêmes en utilisant justement ce texte et donne donc lieu à des catégories Wikipédia que nous pouvons voir comme des *communautés sémantiques*. Cependant, nous ignorerons

cette structure pour nous concentrer sur la topologie du réseau et donc obtenir des *communautés topologiques*. Nous utiliserons ainsi souvent la confrontation de ces communautés topologiques et de ces communautés sémantiques pour la validation des méthodes que nous proposerons. En effet, ces communautés sémantiques sont, a priori, indépendantes de la topologie du réseau.

Nous disposons également d'autres réseaux réels ainsi que de réseaux générés dont on connaît les communautés (vérité de terrain) auxquelles nous pourrons comparer les résultats obtenus ; ces réseaux sont présentés dans l'annexe A. Nous essaierons, à chaque fois que cela est possible, de mettre en place des méthodes de validation automatiques plutôt qu'une validation consistant en l'examen manuel des résultats obtenus. Ce second type de validation (validation "à la main"), bien qu'il soit parfois le seul type de validation effectué n'est pas quantifiable et se révèle parfois trompeur. Nous utiliserons également la visualisation pour les graphes de petite taille.

Nous avons mis à disposition tous les codes et les jeux de données utilisés à l'adresse suivante : `http://bit.ly/maxdan94`.

Première partie

Communautés et mesures de proximité

Chapitre 2

État de l'art sur la détection de communautés ego-centrées

2.1 Introduction

La détection de communautés dans les réseaux a été très largement étudiée du fait des très nombreuses applications qu'elle engendre. Plusieurs articles dressant l'état de l'art sur des points spécifiques de ce vaste sujet en témoignent : [131] et [64] voient principalement la détection de communautés comme un partitionnement du réseau ; [162] est dédié aux communautés recouvrantes ; [23] traite de la détection et du suivi temporel des communautés dans les réseaux dynamiques ; [77] s'intéresse aux méthodes permettant de trouver la structure communautaire globale d'un réseau en passant par la détection de communautés locales à certains nœuds ou ensembles de nœuds. Cependant, un grand ensemble de travaux n'a pas été couvert par ces articles d'état de l'art. Il s'agit des travaux visant à *trouver une communauté à partir d'un nœud donné*. Une telle communauté est parfois appelée communauté égo-centrée sur un nœud ou communauté locale à un nœud.

La définition de ce problème est cependant floue et différentes variations peuvent être classées dans ce domaine général. On peut ainsi se poser plusieurs questions sur l'objectif réel des méthodes de calcul de communautés locales ou ego-centrées :

- S'agit-il de trouver une communauté contenant un nœud donné, sachant que le nœud peut appartenir à plusieurs communautés (amis, famille, collègues comme dans un réseau social), auquel cas on peut chercher à trouver toutes les communautés de ce nœud d'intérêt ? S'agit-il plutôt d'une communauté vue comme un unique groupe de nœuds importants pour le nœud en question ?
- Le groupe doit-il contenir le nœud en question, ou bien doit-il seulement être "proche" du nœud en question ? Autrement dit, le nœud en question est-il important pour la communauté détectée ou bien constitue-t-il seulement un point d'accroche pour lancer un algorithme (auquel cas la communauté trouvée n'a potentiellement rien à voir avec ce nœud de départ) ?
- Quelle est la relation entre la (ou une) communauté locale d'un nœud et le nœud

lui-même ? En particulier, si l'on calcule la (les) communauté(s) en partant d'un autre nœud de cette communauté, retrouve-t-on la même communauté ? Ces communautés locales ont-elles une valeur globale ou bien sont-elles intrinsèquement liées à un nœud en particulier ?

Nous allons voir que la plupart des méthodes peuvent souvent être adaptées pour passer d'un objectif à l'autre. Nous proposons donc de dresser l'état de l'art de tous ces travaux et d'en tirer des méthodes générales tout en discutant, à chaque fois, le type des communautés trouvées.

Nous allons également étudier les méthodes qui cherchent une communauté à partir d'un ensemble de nœuds et non d'un seul. Ce problème peut sembler une simple extension du problème précédent, mais nous verrons que quand il s'agit de plusieurs nœuds, le but est souvent de compléter l'ensemble de nœuds en une unique communauté, qui représente alors le groupe des nœuds importants pour l'ensemble de nœuds de départ.

La plupart des méthodes sont basées sur une fonction de qualité qui évalue la pertinence d'un ensemble de nœuds en tant que communauté (souvent indépendamment du (ou des) nœud(s) d'intérêt), laquelle est optimisée en partant de la communauté triviale constituée seulement du nœud d'intérêt. D'autres méthodes sont, en général, basées sur une définition formelle de ce qu'est une communauté et cherchent à trouver un ou des ensembles vérifiant cette définition.

Dans un premier temps nous présentons, dans la section 2.2, les différentes fonctions de qualité utilisées pour quantifier la pertinence d'un ensemble de nœuds en tant que communauté. Ensuite, dans la section 2.3, nous présentons les différentes façons de les optimiser en partant d'un nœud. Dans la section 2.4, nous présentons les autres méthodes ne reposant pas directement sur une fonction de qualité, mais plutôt sur une définition ou des propriétés que l'ensemble de nœuds doit satisfaire. Bien que la plupart des méthodes précédentes se généralisent aisément à un ensemble de nœuds de départ, nous détaillons dans la section 2.5 les méthodes spécialement dédiées à un ensemble de nœuds. La section 2.6 reprend les points de [77] pour étudier le passage de ces approches locales à une approche globale : il s'agit de trouver toutes les communautés d'un réseau par l'intermédiaire de la détection des communautés autour de certains nœuds. Finalement nous concluons ce chapitre d'état de l'art dans la section 2.7.

2.2 Fonctions de qualité

2.2.1 Fonctions basées sur le nombre de liens internes et externes

La conductance d'un ensemble de nœuds est définie par :

$$\phi(S) = \frac{cut(S)}{\min(vol(S), vol(\overline{S}))}$$

où $cut(S)$ est le nombre de liens entre S et les autres nœuds du réseau, ensemble noté $\overline{S}$. $vol(S)$ est la somme des degrés des nœuds dans l'ensemble S. La conductance est aussi

appelée coupe normalisée [135]. Dans le cas où le nombre de nœuds S est petit devant le nombre de nœuds dans le réseau, ce qui est généralement le cas quand on parle de communauté, une approximation de la conductance est :

$$\phi_{aprox}(S) = \frac{cut(S)}{vol(S)}$$

Ceci peut également s'écrire sous la forme :

$$\phi_{aprox}(S) = \frac{l_o}{2l_i + l_o}$$

où l_o (respectivement l_i) est le nombre de liens sortants (respectivement à l'intérieur) de S.

Remarquons que $l_i = \frac{k_i}{2}$ et $l_o = k_o$, où k_o (respectivement k_i) est la somme des degrés sortants (respectivement internes) de S. Ainsi, on trouve parfois la conductance approchée écrite sous la forme $\phi_{aprox}(S) = \frac{k_o}{k_i+k_o}$. Ce changement de variables étant classique et dépendant de la préférence des auteurs, nous garderons la notation (l_i, l_o) dans les autres fonctions de qualité que nous allons présenter à des fins d'uniformisation.

La conductance compare le nombre de liens internes et externes de la communauté et plus la conductance d'un ensemble de nœuds est faible, plus cet ensemble est censé être une bonne communauté. La conductance a été et reste une fonction de qualité de choix pour évaluer la pertinence d'un ensemble de nœuds en tant que communauté. Elle a été très étudiée et, comme [137] a montré que trouver une coupe dont la conductance est plus faible qu'un certain seuil est NP-difficile, beaucoup d'heuristiques ont été développées afin de l'optimiser, c'est-à-dire de trouver un ensemble de nœuds de conductance minimum. Dans le cadre de communauté(s) centrée(s) sur un nœud d'intérêt (ou des nœuds d'intérêts), on peut chercher un ensemble de nœuds minimisant la conductance et contenant ce (ou ces) nœud(s) d'intérêt. Nous allons présenter ces algorithmes d'optimisation dans la section 2.3.

Une fonction de qualité proche est la densité relative du sous-graphe S, notée $rd(S)$ et parfois appelée modularité locale [99]. Elle est définie par le ratio entre le degré interne et externe du sous-graphe S, c'est-à-dire :

$$rd(S) = \frac{2l_i}{l_o}.$$

Une possible variation de cette fonction est :

$$rd_2(S) = \frac{2l_i}{2l_i + l_o}.$$

Notons que $rd_2(S) = f(rd(S))$ avec $f(x) = \frac{1}{1+\frac{1}{x}}$. Or f est une fonction strictement croissante ainsi, en termes de maximisation, les fonctions rd et rd_2 sont équivalentes. Notons également que :

$$\phi_{aprox}(S) = g(rd(S))$$

avec $g(x) = \frac{1}{1+x}$. Or g est une fonction strictement décroissante, ainsi maximiser la densité relative équivaut à minimiser la conductance approximative.

Dans [88], une densité relative paramétrée est proposée, elle est donnée par la formule :

$$f_\alpha(S) = \frac{2l_i}{(2l_i + l_o)^\alpha},$$

où α est un paramètre réel positif qui permet de contrôler le compromis entre le nombre de liens internes et externes. Le paramètre α permet donc de contrôler la taille de la communauté obtenue en sortie.

Ainsi, il ne semble pas y avoir de différence conceptuelle significative entre ces fonctions qui cherchent toutes à maximiser le nombre de liens internes et à minimiser le nombre de liens externes ; seul le compromis entre ces deux valeurs change pour la densité relative en fonction du paramètre α.

Trouver un sous-graphe d'une taille donnée ayant une densité relative supérieure à un seuil donné est NP-complet [137] et il en va donc de même pour les autres fonctions. Si la taille du sous-graphe n'est pas fixée, toute composante connexe (et union de composantes connexes) est un optimum global de la conductance approchée [1] et de la densité relative, mais pas forcément de la densité relative paramétrée dont le maximum dépend du paramètre α même dans ce cas spécifique.

Quand on cherche une communauté et en particulier si l'on cherche une (ou la) communauté contenant un nœud donné, on s'intéressera donc à un optimum local de ces fonctions. Nous présenterons des algorithmes d'optimisation dédiés à cette tâche dans la section 2.3.

Il existe d'autres fonctions qui sont cependant nettement moins utilisées dans le contexte de la détection de communautés. On peut citer par exemple la densité locale qui correspond au nombre de liens à l'intérieur de la communauté divisé par le nombre de liens maximum que la communauté aurait pu contenir :

$$dl(S) = \frac{2l_i}{n(n-1)},$$

avec n le nombre de nœuds dans S. Remarquons que toute clique maximise cette fonction et en particulier un simple lien. Cette fonction est donc difficile à utiliser, mais elle peut être combinée avec d'autres fonctions ou utilisée avec des contraintes additionnelles sur la taille.

Un autre exemple de fonction utilisée moins fréquemment est la distance d'édition [137, 134] (*single cluster editing*). Il s'agit du nombre de liens qu'il faut ajouter à S ou enlever autour de S pour le transformer en une clique isolée :

$$de(S) = \frac{n(n-1)}{2} - l_i + l_o.$$

1. Il faut ici exclure le graphe entier pour la conductance si l'on ne choisit pas son approximation $f(S) = \frac{l_o}{2l_i+l_o}$, mais sa forme originale $f(S) = \frac{cut(S)}{\min(vol(S),vol(\overline{S}))}$. Trouver une coupe dont la conductance est plus faible qu'un certain seuil est bien NP-difficile.

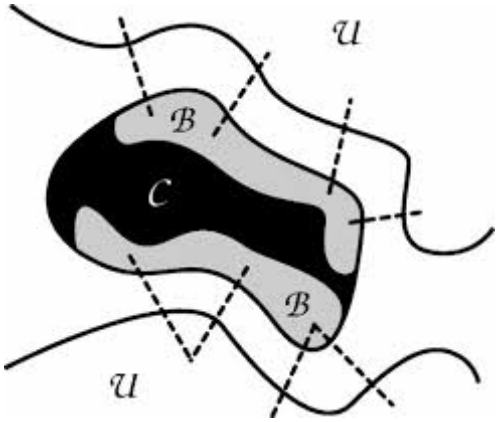


Figure 2.1 – Illustration de la division en trois ensembles : le cœur de la communauté $\mathcal{C}$, la bordure de la communauté $\mathcal{B}$ et les nœuds tout juste à l'extérieur de la communauté $\mathcal{U}$. De plus on note $\mathcal{D} = \mathcal{C} \cup \mathcal{B}$ l'ensemble des nœuds dans la communauté. Cette figure est extraite de [43].

Cette fonction peut peiner à quantifier pertinemment des communautés dans un graphe peu dense du fait du terme en $\frac{n(n-1)}{2}$ qui croît rapidement : l_i et l_o peuvent en effet être négligeables devant ce terme dans un graphe creux si n est un peu trop grand. Ainsi sa minimisation peut tendre à donner de petites communautés peu pertinentes. C'est sûrement pour cela qu'elle est moins utilisée.

2.2.2 Fonctions basées sur les connexions entre le cœur, la bordure et l'extérieur de la communauté

Plusieurs fonctions de qualité ne sont pas basées sur le nombre de liens internes et externes, mais sur le nombre de liens entre les trois ensembles détaillés sur la figure 2.1 :

- l'ensemble $\mathcal{C}$ est le cœur de la communauté ; il est constitué des nœuds ayant tous leurs voisins dans la communauté,
- l'ensemble $\mathcal{B}$ est la bordure de la communauté ; il est constitué des nœuds dans la communauté mais ayant au moins un voisin hors de la communauté,
- et l'ensemble $\mathcal{U}$ est constitué des nœuds hors de la communauté mais ayant au moins un voisin inclus dans la communauté.
- De plus, on note $\mathcal{D} = \mathcal{C} \cup \mathcal{B}$ l'ensemble des nœuds dans la communauté.

La première fonction utilisant cette division en trois ensembles a été introduite dans [43] et est appelée la modularité locale R. Elle est donnée par la formule :

$$R = \frac{I}{T}$$

où I est le nombre de liens entre $\mathcal{D}$ et $\mathcal{B}$. T est la somme des degrés des nœuds dans $\mathcal{B}$.

Il y a ici une différence conceptuelle entre cette fonction de qualité et celles présentées précédemment comparant seulement les nombres de liens internes et externes. Ici, seuls les

liens avec une extrémité dans la bordure $\mathcal{B}$ sont considérés. En effet : "I est le nombre de liens entre $\mathcal{D}$ et $\mathcal{B}$; T est la somme des degrés des nœuds dans $\mathcal{B}$", mais il suffit de remplacer $\mathcal{B}$ par $\mathcal{D}$ pour avoir "I est le nombre de liens à l'intérieur de $\mathcal{D}$; T est la somme des degrés des nœuds dans $\mathcal{D}$" et ainsi retrouver la fonction :

$$rd2(S) = \frac{2l_i}{2l_i + l_o}$$

présentée dans la section précédente. Cette fonction de qualité permet donc de ne pas considérer le centre de la communauté qui peut parfois être très dense (une clique) pour se concentrer sur les nœuds en bordure ; elle peut donc donner des communautés (communauté désigne ici un optimum local des fonctions de qualité) très différentes de celles données par les fonctions précédentes. Cependant, si dans un réseau particulier, chaque nœud a au moins un lien en dehors de la communauté, alors les deux fonctions sont équivalentes (puisque dans ce cas $\mathcal{B} = \mathcal{D}$) et les deux fonctions mènent donc aux mêmes communautés dans ce cas précis.

Encore une fois, toute composante connexe (et union de composantes connexes) est un optimum global de cette fonction de qualité et donc quand on utilise cette fonction, on cherche des optima locaux. Nous présenterons des algorithmes d'optimisation dédiés à cette tâche dans la section 2.3.

Dans [40], les auteurs mettent en évidence un défaut de la fonction de qualité précédente illustré par la figure 2.2 : si le nœud d'intérêt est dans une communauté avec un de ses nœuds adjacent à un chemin, alors la qualité de la communauté mesurée par la fonction précédente sera moindre que la qualité de l'ensemble des nœuds constitué de l'union de la communauté et du chemin. Pourtant le chemin contient des nœuds éloignés du nœud d'intérêt qu'il ne semble pas pertinent d'inclure dans la communauté du nœud d'intérêt. Ce défaut est courant dans les fonctions de qualité, on le retrouve par exemple pour la densité relative, $dr(S) = \frac{l_i}{l_o}$.

Ce défaut vient du fait que les fonctions de qualité courantes cherchent à (i) avoir une relation forte entre les nœuds de la communauté et (ii) avoir une relation faible entre les nœuds intérieurs et les nœuds extérieurs à la communauté. Le premier point est en général évalué par la somme des degrés intérieurs à la communauté et le deuxième par la somme des liens sortants de la communauté. Les auteurs de [40] avancent que l'aspect manquant à ces fonctions de qualité est la densité des connexions et pas seulement leur nombre. En tenant compte de ces remarques, ils proposent donc la fonction de qualité suivante qui permet de pallier ce défaut :

$$L = \frac{L_{\text{in}}}{L_{\text{ex}}},$$

où L_{in} est le degré moyen des nœuds dans le sous-graphe induit par D et L_{ex} est le nombre de liens sortants de la communauté qu'a en moyenne un nœud dans B.

Dans [117], les auteurs mettent en évidence d'autres limites des fonctions de qualité actuelles. Par exemple, lors d'une optimisation gloutonne commençant avec seulement le nœud d'intérêt dans la communauté (approche la plus utilisée), les nœuds avec les degrés

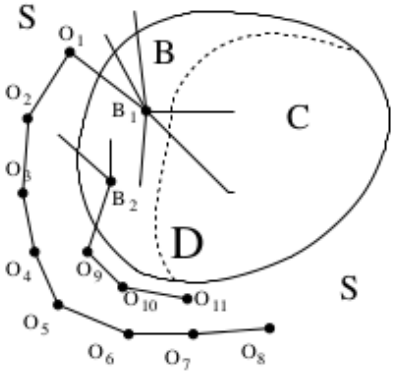


Figure 2.2 – Illustration d'un défaut classique des fonctions de qualité : les longs chemins sont entièrement inclus. Cette figure est extraite de [40].

les plus faibles sont systématiquement ajoutés en premier au début de l'optimisation. Cela introduit un biais et certains maxima ne sont pas accessibles, surtout si le nœud est à la frontière de plusieurs communautés. Les auteurs présentent alors une nouvelle fonction de qualité afin de pallier ce problème ainsi qu'une nouvelle méthode d'optimisation que nous détaillerons plus tard. Leur fonction de qualité est la suivante :

$$T = \frac{T_{\text{in}}}{T_{\text{ex}}}$$

où T_{in} et T_{ex} sont similaires à L_{in} et L_{ex} de [99], mais les contributions de chaque nœud sont pondérées en fonction de la distance au nœud d'intérêt :

$$T_{\text{in}} = \frac{\sum_{i \in \mathcal{D}} \frac{|N_i \cap \mathcal{D}|}{1+d_i}}{|\mathcal{D}|},$$

$$T_{\text{ex}} = \frac{\sum_{i \in \mathcal{D}} |N_i \cap \mathcal{U}|(1+d_i)}{|\mathcal{D}|},$$

où N_i est l'ensemble des voisins du nœud i et d_i est la distance du nœud i au nœud d'intérêt.

Les auteurs exhibent plusieurs graphes jouets problématiques sur lesquels ils arrivent à donner la solution voulue, ce qui n'est pas possible par l'optimisation simple des autres fonctions. Remarquons ici que les deux fonctions de qualité précédentes évaluent une communauté en tant que telle, mais que cette dernière prend en compte la distance au nœud d'intérêt et est intrinsèquement liée à ce nœud d'intérêt.

Remarquons également que cette dernière fonction, bien qu'elle soit née d'une évolution des fonctions basées sur les connexions entre le cœur, la bordure et l'extérieur de la communauté, n'utilise pas cette notion de bordure et compare seulement les liens dans la

communauté et hors de la communauté ; elle pourrait donc être placée dans la partie précédente.

Il y a quelques différences conceptuelles entre ces trois mesures de proximité mais, pour le problème de détection de communauté locale à un nœud, les fonctions introduites plus récemment sont des améliorations palliant des défauts spécifiques.

2.2.3 Fonctions basées sur la densité de triangles

Dans [66], les auteurs suggèrent une fonction de qualité, appelée *cohésion*, basée sur la densité interne et externe de triangles. Leur fonction est construite autour de trois hypothèses :

1. La qualité d'un ensemble de nœuds en tant que communauté ne dépend pas de l'existence d'autres communautés.
2. La qualité d'une communauté n'est pas affectée par des nœuds lointains.
3. Une communauté est un ensemble "dense" au sein duquel une information se propage plus facilement que vers le reste du réseau.

En prenant en considération la nature intrinsèque du recouvrement entre communautés, ils suggèrent d'utiliser les triangles plutôt que les liens pour construire la fonction de qualité. En effet, en considérant que les liens entre nœuds peuvent être de nature différente, par exemple des liens entre des personnes de la même famille ou des liens entre collègues, pour un triangle il y a de grandes chances que les trois liens soient de même nature. Au contraire, des liens n'appartenant à aucun triangle peuvent être considérés comme des liens faibles (*weak ties* en anglais) et servir de ponts entre communautés ou être considérés comme du bruit. Ainsi leur exclusion ne devrait pas changer la qualité d'une communauté.

Pour la même raison, seuls les triangles à l'intérieur de la communauté ou sortant de la communauté devraient affecter la qualité d'un ensemble de nœuds en tant que communauté. Des liens sortant de la communauté ne sont pas forcément vus comme néfastes pour la qualité de la communauté du moment qu'ils ne forment pas de triangles sortant de la communauté.

Étant donné un ensemble de nœuds S et en notant $\triangle_i(S)$ le nombre de triangles inclus dans S et $\triangle_o(S)$ le nombre de nœuds sortant de S, c'est-à-dire ayant deux nœuds dans S et un nœud en dehors, la cohésion $C(S)$ de l'ensemble S est donnée par :

$$C(S) = \frac{\triangle_i(S)}{\binom{|S|}{3}} \times \frac{\triangle_i(S)}{\triangle_i(S) + \triangle_o(S)}$$

Le premier terme correspond à la densité de triangles dans l'ensemble S, tandis que le deuxième terme mesure l'isolement de l'ensemble S en comparant le nombre de triangles internes et externes.

Pour valider leur mesure, les auteurs réalisent une expérience sur Facebook : ils demandent à des utilisateurs de Facebook de noter la pertinence de groupes d'amis ayant différentes valeurs de cohésion. Les auteurs observent une forte corrélation entre les notes

données par les utilisateurs et les valeurs de cohésion. Cette corrélation est plus forte que la corrélation obtenue par d'autres métriques telles que la densité de liens ou la conductance. Remarquons cependant que pour cet article les auteurs se limitent à des communautés consistant en un nœud et une partie de ses voisins (les communautés du nœud d'intérêt contiennent seulement des nœuds qui lui sont adjacents). Les auteurs prouvent également que trouver un ensemble de nœud dont la cohésion est supérieure à un certain seuil est un problème NP-difficile et ils proposent une heuristique pour optimiser la cohésion, appelée C^3, dont nous reparlerons dans la section 2.3.

Dans [125], les auteurs construisent différentes fonctions de qualité en considérant également les triangles plutôt que les liens. Ils montrent en effet que l'optimisation des fonctions de qualité basées sur les liens, comme la conductance, peut donner des communautés qui ne sont pas denses, pauvres en triangles et contenant des ponts entre communautés.

2.3 Optimisation en partant d'un nœud d'intérêt

Nous avons détaillé les principales fonctions de qualité qui évaluent la pertinence d'un ensemble de nœuds en tant que communauté. Il s'agit maintenant de les optimiser en partant d'un nœud d'intérêt afin de trouver :

1. la communauté du nœud d'intérêt,
2. les communautés du nœud d'intérêt ou
3. une communauté plus ou moins proche du nœud d'intérêt.

Les algorithmes d'optimisation peuvent se séparer en deux grandes familles (i) les algorithmes gloutons et stochastiques qui peuvent être adaptés pour trouver les trois types de communautés et (ii) les algorithmes s'appuyant sur une mesure de proximité et cherchant à découvrir le premier type de communauté.

2.3.1 Différentes variantes d'algorithmes gloutons et stochastiques

Étant donné un nœud d'intérêt n_i et une fonction de qualité f, les méthodes d'optimisation gloutonnes peuvent en général être décomposées selon les trois étapes suivantes :

1. **initialisation :** on part de la communauté contenant seulement le nœud d'intérêt : $C = \{n_i\}$,
2. **optimisation :** on ajoute à chaque fois le nœud, voisin de la communauté C, qui optimise le plus la fonction f,
3. **arrêt :** on s'arrête lorsque l'on ne peut plus augmenter la fonction de qualité.

Chacune de ces trois étapes peut être modifiée indépendamment et la plupart des algorithmes sont des variantes de cet algorithme générique. Ils permettent de couvrir les différents types de communautés recherchées (la communauté du nœud d'intérêt, toutes

ses communautés ou une communauté plus ou moins proche du nœud d'intérêt). Nous discutons ici ces différents types de modification.

— **Discussion sur l'initialisation :** il est possible de commencer avec plusieurs nœuds si l'on cherche à compléter un ensemble de nœuds en une communauté. Ou bien d'ajouter seulement certains nœuds en plus du nœud d'intérêt (par exemple certains de ses voisins) si l'on veut trouver une communauté spécifique du nœud d'intérêt ou bien introduire du bruit pour trouver plusieurs communautés différentes, en lançant l'algorithme plusieurs fois avec d'autres "nœuds bruits" de départ. Il est également possible de commencer avec tous les nœuds du graphe dans la communauté et ensuite de les enlever pas à pas de manière à optimiser la fonction de qualité. C'est la variante choisie dans [140] ; elle permet dans certains cas d'arriver à un algorithme optimal, nous détaillerons cette méthode dans la section 2.5. Dans [71], les auteurs montrent formellement que la présence de deux propriétés courantes dans les réseaux : (i) un fort coefficient de clustering et (ii) une distribution en loi de puissance des degrés, implique que les sous-graphes induits par nœud et tous ses voisins ont en général une conductance élevée. Ils montrent de manière empirique que c'est effectivement le cas dans des réseaux réels. Ainsi, commencer la recherche de communauté avec ce type d'ensemble de nœuds (un nœud et ses voisins) semble être une bonne heuristique d'initialisation.

— **Discussion sur le critère d'arrêt :** il est possible de continuer même si aucun nœud ne permet d'augmenter la fonction de qualité, par exemple en ajoutant celui qui la fait le moins diminuer et en s'arrêtant lorsque tous les nœuds du graphe (ou un nombre de nœuds fixé en entrée, si le temps est un facteur limitant) ont été ajoutés. Dans certains cas, on peut observer, en fonction du nombre de nœuds ajoutés, une diminution puis une augmentation de la fonction de qualité. Il est ainsi possible d'éviter de rester coincé dans certains maxima locaux. Dans ce cas, on retourne l'ensemble de nœuds qui a maximisé la fonction de qualité au cours de l'optimisation. Cette variante est utilisé dans [43].

— **Discussion sur l'optimisation :** on peut ajouter tous les nœuds menant indépendamment à une augmentation de la fonction de qualité au lieu d'en ajouter seulement un comme réalisé dans [40].

  Il est possible, comme dans [67], d'ajouter certains nœuds qui font baisser la fonction de qualité dans l'immédiat, mais vont potentiellement la faire croître dans le futur, quitte à revenir en arrière si le résultat n'est pas aussi bon que souhaité.

  Il est également possible d'ajouter non pas le nœud qui améliore le plus la fonction de qualité, mais simplement un nœud aléatoirement choisi parmi ceux qui la font croître. L'algorithme devient alors stochastique et non-déterministe : on peut obtenir plusieurs communautés pour un même nœud. En lançant plusieurs fois l'algorithme il est alors possible d'obtenir différentes communautés mais une étape de nettoyage des communautés trouvées est souvent nécessaire car des communautés très similaires, sans être parfaitement identiques, peuvent être trouvées. Remarquons que l'on peut tirer avantage de cette redondance des communautés trouvées afin de donner en sortie seulement les communautés vraiment significatives (celles

qui apparaissent plusieurs fois) [88].
Même dans le cas glouton, on peut obtenir différentes communautés en lançant plusieurs fois l'optimisation et en introduisant du bruit au préalable (on peut le faire également dans le cas d'un algorithme stochastique, si l'on veut accroître le bruit pour atteindre plus de minima locaux différents). Ce bruit peut être introduit durant l'initialisation en introduisant d'autres sommets en plus du nœud d'intérêt (comme vu précédemment) ou l'on peut modifier légèrement le réseau entre chaque optimisation comme dans [117]. Dans cette approche, les auteurs proposent d'effacer les liens de la communauté précédemment trouvée entre chaque optimisation jusqu'à ce que le nœud d'intérêt n'ait plus de voisin. L'algorithme associé est appelé IOLoCo (Identification of Overlapping Local Communities)
Il est aussi possible d'ajouter une étape où l'on cherche également à enlever un nœud auparavant inclus dans la communauté si cela augmente d'avantage la fonction de qualité (variante utilisée dans [40, 88]). Dans ce cas, des variations légères permettent de chercher les différents types de communautés. (i) Si l'on cherche "la communauté" (resp. les communautés) du nœud d'intérêt, il suffit de lancer une fois (resp. plusieurs fois) l'optimisation en interdisant d'enlever le nœud d'intérêt de la communauté et de retourner la communauté obtenue (resp. les communautés obtenues). (ii) Si l'on cherche simplement une communauté, possiblement sans relation avec le nœud d'intérêt, on peut s'autoriser à enlever le nœud de départ de la communauté. En général, on obtient une communauté proche du nœud de départ, cependant cela n'est pas garanti et il est possible d'obtenir une communauté lointaine, sans relation avec ce nœud.

2.3.2 Algorithme nibble et algorithmes dérivés

Un algorithme, appelé *nibble*, a été développé dans [141] (et étendu dans [142]) pour trouver un ensemble de nœuds avec une connexion interne sensiblement plus grande que sa connexion externe (i.e., trouver une coupe de conductance faible) près d'un nœud d'intérêt. L'algorithme simule une marche aléatoire feignante tronquée (lazy truncated random walk), étant donné un ensemble de nœuds d'intérêt E_i, la marche aléatoire est initialisée avec la distribution $p_0 = \psi_S$ de marcheurs suivante :

$$\psi_S = \begin{cases} d(x)/Vol(S) & \text{si } x \in S \\ 0 & \text{sinon.} \end{cases}$$

Ensuite un pas de la marche à l'itération t peut être décrit de la façon suivante :

1. Calculer la distribution $p_t = Mp_{t-1}$, où $M = \frac{1}{2}(I + AD^{-1})$ est la matrice de transition de marcheur aléatoire feignant (avec A la matrice d'adjacence du graphe, I la matrice identité et D la matrice des degrés [2]).
2. Trier tous les nœuds ayant un p_t non nul par ordre décroissant en fonction des probabilités normalisées : $r_t(v) = p_t(v)/d(v)$.

2. D_{ii} correspond au degré du nœud i, les autres entrées sont à 0.

3. Pour tout nœud v tel que $r_t(v) \leq \varepsilon$ mettre la probabilité $p_t(v)$ à 0.
4. Faire un balayage : pour chaque i séparer les nœuds en deux ensembles : les nœuds classés avant i et les nœuds classés après i et calculer la valeur de la conductance de la coupe du graphe séparant les deux ensembles. Retourner la coupe si sa conductance est plus faible que la valeur de la coupe voulue.

Des résultats théoriques importants sont développés dans le papier [17], en particulier les auteurs prouvent que sous certaines conditions l'algorithme donnera une coupe de bonne conductance (ils obtiennent une borne maximum en fonction de plusieurs paramètres) en un temps linéaire en fonction de la taille de l'ensemble en sortie.

Dans [20], les auteurs généralisent *nibble* avec les trois points suivants.

1. Ils ne cherchent pas l'indice i qui sépare les deux ensembles en faisant un balayage, mais séparent les deux ensembles en mettant tous les nœuds classés avant i dans le premier, tous les nœuds classés après $i+t$ (où t est un paramètre) dans le deuxième. Ceux qui sont entre i et $i+t$ sont placés dans un des deux ensembles de manière à minimiser le nombre de liens dans la coupe en utilisant le théorème MaxFlow-MinCut. Cette heuristique permet empiriquement d'améliorer la conductance de l'ensemble donné en sortie.
2. En choisissant les paramètres de manière plus libre.
3. En retournant en sortie l'ensemble de nœuds qui a donné la conductance la plus faible.

Un algorithme dérivé est développé dans [18] et appelé le *pagerank-nibble*. Il consiste à utiliser le pagerank enraciné (*rooted pagerank* en anglais) sur le (ou les) nœud(s) d'intérêt(s) plutôt qu'une marche aléatoire feignante tronquée.

Le pagerank enraciné sur un nœud i est donné par la solution du système suivant :

$$pr_\alpha(i) = \alpha I_i + (1-\alpha)Tpr_\alpha(i),$$

où $T = AD^{-1}$ est la matrice de transition de marcheur aléatoire et I_i est le vecteur nul, sauf pour l'entrée i où il vaut 1. Cette définition peut être étendu à un pagerank enraciné sur un nombre n de nœuds et le vecteur initial I est alors le vecteur nul sauf pour les n nœuds concernés où il vaut $\frac{1}{n}$.

Les auteurs ne calculent pas le pagerank enraciné exact, mais un pagerank enraciné approché, ce qui permet de rendre le temps d'exécution de l'algorithme linéaire en fonction de la taille de l'ensemble de nœuds en sortie et indépendant de la taille du graphe. Nous détaillerons cette méthode dans le chapitre 3.

Cette approche est également généralisée dans [19] à des graphes dirigés.

Dans [17], les auteurs montrent formellement que, lorsqu'il y a une décroissance forte dans les valeurs du pagerank enraciné (les valeurs étant triées par ordre décroissant), une coupe séparant les nœuds avant la forte décroissance et après la forte décroissance constitue une coupe de faible conductance. Une intuition de ce phénomène est que quand un pas de marche aléatoire est appliqué avec la distribution de marcheur aléatoire donnée par un pagerank enraciné, alors tous les nœuds du graphe ont plus de probabilité d'avoir un

marcheur excepté le nœud graine. Ainsi, il ne peut pas exister beaucoup de liens entre un ensemble de nœud de forte probabilité et un ensemble de nœuds de faible probabilité. En effet, dans le cas contraire, l'ensemble de faible probabilité drainerait les marcheurs de l'ensemble de forte probabilité.

Ces techniques d'optimisation dérivées de *nibble* sont seulement utilisées pour l'optimisation de la conductance, mais remarquons qu'elles pourraient être adaptées et utilisées pour optimiser d'autres fonctions de qualité.

Dans [166], les auteurs mettent en évidence un autre intérêt de ces algorithmes dérivés de *nibble*. En effet, en plus d'obtenir un groupe de nœuds avec une conductance faible, il est important que ce groupe de nœuds soit bien connecté. Or, il est possible d'avoir un groupe de nœuds avec une conductance faible, mais peu connecté à l'intérieur, ou même déconnecté. Les auteurs donnent des garanties théoriques sur le fait qu'en utilisant l'algorithme *pagerank-nibble* de [18] pour trouver un ensemble de faible conductance, les ensembles obtenus, en plus d'avoir une faible conductance, seront bien connectés.

Dans [13], les auteurs proposent une version avec plusieurs marcheurs aléatoires liés entre eux par un fil. L'idée directrice est qu'il est plus difficile pour plusieurs marcheurs aléatoires de sortir simultanément de la communauté par un goulot d'étranglement que pour un marcheur aléatoire unique.

Ces méthodes dérivées de *nibble* permettent de trouver "la" communauté du nœud d'intérêt. Si l'on cherche plusieurs communautés alors elles doivent être adaptées, c'est ce que nous ferons dans le chapitre 4.

2.4 Autres méthodes de détection de communauté à partir d'un nœud

Nous venons de détailler les méthodes basées sur l'optimisation locale d'une fonction de qualité. Remarquons que les communautés trouvées par ces méthodes dépendent en général très fortement du nœud de départ. Ainsi, si l'on trouve une certaine communauté C_x en partant d'un nœud x donné, on ne la trouvera pas forcément à l'identique en partant d'un autre nœud $y \in C_x : C_x \neq C_y$. Ainsi ces communautés n'ont pas de valeur globale. Nous étudions ici les autres méthodes qui ne sont pas basées sur une fonction de qualité, mais plutôt sur une définition formelle de ce qu'est une communauté. Ces communautés ont en général une valeur globale.

La première définition naturelle d'une communauté est la notion de clique maximale [98]. Ainsi les communautés auxquelles appartient le nœud d'intérêt peuvent être simplement toutes les cliques maximales le contenant. Un algorithme immédiat consistant à chercher toutes ces cliques maximales en découle, mais l'énumération des cliques maximales est exponentielle en général, ce qui peut poser problème dans de grands graphes. De plus la notion de clique maximale est très structurante : si un sommet est connecté à tout le groupe sauf un de ses membres il sera exclu de la communauté.

Beaucoup d'autres définitions plus élaborées ont été avancées et pour chacune on peut

proposer des algorithmes simples, mais souvent exponentiels, pour trouver les communautés du nœud d'intérêt.

Les définitions sont par exemple :

— Une n-clique : un ensemble maximal de nœuds tel que la distance entre chaque paire de nœuds est inférieure ou égale à n [97, 14]. Une clique est une 1-clique avec cette définition.

— Un n-clan : un sous-graphe maximal tel que la distance entre chaque paire de nœuds du sous-graphe (calculé sur le sous-graphe lui-même et pas sur le graphe entier comme dans le cas précédent) est inférieure ou égale à n [108].

— Un k-plex : un sous-graphe maximal tel que chaque nœud soit adjacent à tous les autres nœuds du sous-graphe sauf au plus k d'entre eux [132].

— Un k-core : un sous-graphe maximal tel que chaque nœud du sous-graphe soit adjacent à au moins k autres nœuds du sous-graphe [132].

— Des k-cliques percolées : un ensemble maximal de cliques de taille k tel qu'il existe une série de cliques de taille k adjacentes sur k-1 sommets entre chaque paire de cliques de l'ensemble [120].

Un ensemble LS [96], ou communauté forte [126], est un ensemble de nœuds tel que chaque nœud a un degré interne (nombre de voisins dans l'ensemble) plus grand que son degré externe (nombre de voisins en dehors). Cette condition, assez stricte, peut être relaxée en définissant la notion de communauté faible [126] : un ensemble de nœuds tel que la somme de ses degrés internes est plus élevée que la somme de ses degrés externes. Une communauté forte est aussi une communauté faible, mais l'inverse n'est pas vrai.

Une communauté forte peut être trouvée en utilisant le théorème MaxFlow-MinCut. C'est l'approche utilisée dans [63]. Étant donné un *nœud source* (nœud d'intérêt) que l'on suppose dans la communauté et un *nœud puits* en dehors de la communauté, les auteurs suggèrent de séparer le réseau en deux par la coupe minimale entre les deux nœuds. L'ensemble contenant le nœud source est alors une communauté forte. En effet, sinon il existe un nœud tel que son degré interne est plus faible que son degré externe, enlever ce nœud de l'ensemble de nœud contenant le nœud source pour le mettre dans l'autre ensemble conduirait alors à une coupe plus faible, ce qui est absurde. Remarquons que l'on peut utiliser la méthode sans grande modification avec plusieurs nœuds sources et plusieurs nœuds puits. On peut donc, avec cette méthode, compléter un ensemble de nœuds en communauté en utilisant une information de nœuds que l'on sait en dehors de la communauté. On peut également appliquer la méthode plusieurs fois en annotant comme inclus ou exclus de la communauté certains nœuds entre chaque itération. Les auteurs proposent également une version ne nécessitant pas de connaître des nœuds en dehors de la communauté en utilisant un *nœud puits universel*. Le poids du lien entre chaque nœud et le puits universel contrôle la taille de la communauté finale.

Une autre méthode intéressante est celle de Bagrow et Bolt [25]. Ils proposent d'approximer la communauté d'un nœud d'intérêt par l'ensemble des nœuds à distance inférieure ou égale à l du nœud d'intérêt. Leur méthode consiste donc à simplement trouver la valeur la plus pertinente de cet entier l.

Étant donné un nœud d'intérêt j, leur algorithme augmente l tant que :

$$\Delta K_j^l < \alpha.$$

ΔK_j^l correspond au changement du degré extérieur :

$$\Delta K_j^l = \frac{K_j^l}{K_j^{l-1}},$$

avec K_j^l correspondant aux nombres de liens entre les nœuds à distance l de j et les nœuds à distance $l+1$. α est le paramètre de contrôle du critère d'arrêt, un petit α permet d'avoir une communauté plus grande.

Bien sûr cette méthode est loin d'être parfaite : un nœud d'intérêt problématique peut faire que l'ensemble de départ couvre deux communautés ou plus (en effet, il suffit d'un seul lien vers une communauté lointaine pour que le résultat final soit peu pertinent). Cependant, l'ensemble de départ reste une approximation de communauté ou d'ensemble de communautés et pourrait être utilisé en conjonction avec d'autres algorithmes présentés précédemment. Les auteurs présentent également une méthode pour passer de ces communautés locales à une partition du réseau en communautés qui sera détaillée dans la section 2.6.

2.5 Détection de communauté à partir d'un ensemble de nœuds

Nous avons déjà montré que les méthodes cherchant une communauté à partir d'un nœud pouvaient être modifiées pour chercher une communauté à partir d'un ensemble de nœuds (il s'agit du problème de la complétion d'un ensemble de nœuds en une communauté). Cependant, il existe des méthodes complètement dédiées à cette tâche, nous les détaillons ici.

2.5.1 Algorithme nibble pour compléter une communauté

Dans [17], les auteurs s'autorisent à utiliser l'algorithme *nibble* en redémarrant aléatoirement d'un nœud faisant partie d'un ensemble d'intérêt plutôt que d'un seul nœud d'intérêt. Ils montrent alors formellement que tout ensemble de nœuds constitué d'une partie conséquente d'un ensemble de nœuds avec une conductance faible sera complété en un ensemble de nœuds avec une conductance faible : cette observation est très utile lorsque l'on cherche à compléter en une communauté un ensemble de nœuds qui s'en approche.

2.5.2 Sous-graphe de proximité

Dans le but de trouver un petit sous-graphe connexe et étroitement relié à quelques nœuds d'intérêt, plusieurs articles ont développé des fonctions de qualité basées sur la proximité aux nœuds d'intérêt ; nous détaillons ici ces proximités et fonctions de qualité.

Dans [61], les auteurs définissent *le problème de sous-graphe de connexion* (*connection subgraph problem*) : étant donné un graphe G, deux nœuds s et t dans G et un budget b, le but est de trouver un sous-graphe connexe H de taille au plus b contenant s et t qui maximise une fonction de qualité $g(H)$. Ils suggèrent une fonction $g(H)$ basée sur le courant électrique : ils essayent de trouver le sous-graphe H de taille b qui conduit le plus de courant électrique de s à t sous une certaine perte (un paramètre contrôlant la perte est fixé arbitrairement). Les auteurs développent une heuristique basée sur la programmation dynamique ainsi qu'un algorithme d'approximation pour traiter des grands graphes. Pour l'algorithme d'approximation ils sélectionnent un sous-graphe beaucoup plus grand que b (la taille du résultat souhaité), mais beaucoup plus petit que la taille du graphe complet et appliquent leur heuristique dessus.

Dans [84] les auteurs cherchent à trouver un *graphe de proximité*, qui est une généralisation du problème précédent à plus de deux nœuds. Étant donnés deux nœuds s et t, les auteurs présentent la *Cycle-Free Escape Probability* : $CFEP_G(s,t)$ qui est la probabilité qu'un marcheur aléatoire dans le graphe G partant de s arrive à t sans visiter un nœud deux fois (d'où le terme *cycle-free*). Ils essayent ensuite de trouver le sous-graphe H qui maximise $\frac{(CFEP_H(s,t))^\alpha}{n}$ où n est le nombre de nœuds dans H. Ce problème est une relaxation du problème où l'on cherche à trouver le sous-graphe H de taille k tel que $CFEP_H(s,t)$ est maximal et il permet de fixer la taille voulue pour le sous-graphe de manière moins stricte. Pour calculer $CFEP_H(s,t)$, ils utilisent une approximation basée sur le calcul des k chemins les plus courts sur le même sous-graphe H mais avec les arêtes pondérées intelligemment. Ils utilisent ensuite une heuristique pour trouver le meilleur sous-graphe H. Ils généralisent leur approche pour un ensemble de nœuds S plus grand que 2 : trouver le sous-graphe H qui maximise la somme de $CFEP_H(s,t)$ pour toute paire de nœuds dans S.

Dans [149] les auteurs cherchent un sous-graphe connexe de taille b contenant des nœuds qui sont "proches" d'au moins K nœuds parmi les nœuds de l'ensemble donné en entrée. Pour cela, comme dans [61], ils mettent au point une fonction de qualité pour quantifier la pertinence d'un sous-graphe donné en fonction des proximités aux nœuds d'intérêt. Pour cela, ils utilisent les marches aléatoires avec téléportation au nœud d'intérêt (le pagerank enraciné) et normalisent chaque entrée de la matrice de transition par un facteur $\frac{1}{d^\alpha}$, où d est le degré du nœud d'arrivée ; le but est que les nœuds de fort degré ne drainent tous les marcheurs aléatoires. Ils avantagent ainsi les chemins passant par des nœuds de faible degré. Leur méthode a donc deux paramètres : α et la probabilité de se télétransporter au nœud d'intérêt (tous deux fixés arbitrairement à 0.5 dans l'article). Les auteurs combinent alors les scores obtenus par les pageranks enracinés normalisés en faisant ce qu'ils appellent un "K_softAND" : la proximité d'un nœud à l'ensemble de nœuds donné en entrée est donnée par la probabilité que ce nœud ait au moins K marcheurs aléatoires et ils développent une heuristique pour l'optimiser.

2.5.3 Fonctions nœud-monotones

Dans [140], les auteurs cherchent à résoudre le problème suivant : étant donné un graphe $G = (V, E)$, un ensemble de nœuds d'intérêts $Q \subseteq V$, et un nombre d, leur but est de trouver un sous-graphe induit $H = (V_H, E_H)$ de G tel que :

1. V_H contient Q ($Q \subseteq V_H$) ;
2. H est connexe ;
3. $D_H(Q) \leq d$; et
4. $f(H)$ est maximum parmi toutes les solutions possibles pour H.

En posant $D_H(Q, v) = \sum_{q \in Q} d_H(q, v)$ avec $d_H(q, v)$ la distance dans le sous-graphe H entre les nœuds v et q, $D_H(Q) = \max_{v \in H} D_H(Q, v)$. $f(H)$ est une fonction mesurant la qualité du sous-graphe H en tant que communauté (ici on ne considère pas l'extérieur du sous-graphe pour évaluer sa qualité). Ce problème correspond donc à une formulation de la complétion d'un ensemble de nœuds en une communauté. d est un paramètre contrôlant la taille du graphe en sortie. Remarquons qu'au lieu d'utiliser la distance, on peut utiliser une mesure de proximité afin de tenir compte de la redondance des liens et d'avoir une sélection plus discriminante que la distance (à valeur entière et peu élevée dans les réseaux réels).

Pour une fonction $f(H) = \min_{v \in H} g(H,v)$ où g est une fonction nœud-monotone [3] et en ne considérant pas la contrainte 3, les auteurs présentent un algorithme d'optimisation glouton optimal.

Celui-ci commence avec un ensemble contenant tous les nœuds. À chaque itération on enlève le nœud v ayant un $g(H, v)$ minimum jusqu'à ce qu'un des nœuds d'intérêt ait un $g(H, v)$ minimum ou jusqu'à ce que l'ensemble des nœuds d'intérêt soit déconnecté. On retourne le sous-graphe ayant eu le $f(H)$ le plus élevé durant l'optimisation.

Par exemple, pour $g(H, v)$ définie comme le degré du nœud v dans le sous graphe H, $f(H)$ est alors le degré minimum des nœuds du sous graphe H dans le sous-graphe H (ce qui peut effectivement être vu comme une fonction évaluant la qualité du sous-graphe H en tant que communauté).

2.5.4 Trouver des communautés imbriquées

Lorsque l'on souhaite compléter un ensemble de nœuds en une communauté, on doit faire face à un dilemme : (i) sélectionner une communauté dense, mais petite, contenant l'ensemble de nœuds ou (ii) sélectionner une communauté moins dense, mais plus grande. C'est ce qu'avancent les auteurs de [147] et, de manière à éviter ce dilemme, ils proposent de chercher une séquence de *communautés imbriquées*. Ainsi, étant donné un ensemble de nœuds S et un entier k, les auteurs cherchent une séquence de k communautés S_i telles

3. Une fonction f est nœud-monotone non-croissante si pour tout graphe G, pour tout sous-graphe H de G et pour tout nœud v de H, on a $f(H, v) \leq f(G, v)$. Les fonctions nœud-monotones non-décroissantes sont définies de manière similaire.

que ($S = V_1 \subseteq V_1 \subseteq ... \subseteq V_k = V$). Ils font en sorte que la densité décroisse et qu'elle soit le plus uniforme possible à l'intérieur de chaque communauté trouvée. Les auteurs développent alors une fonction de qualité pour évaluer une telle séquence de communautés et proposent de séparer son optimisation en deux sous-problèmes : (i) trier les nœuds et (ii) segmenter les nœuds triés en k ensembles. Le tri est basé sur une décomposition en k-cores du graphe après l'avoir pondéré en fonction du pagerank enraciné sur les nœuds d'intérêt.

2.6 Des communautés locales aux communautés globales

Enfin, un dernier problème consiste à étudier comment le calcul de communautés locales peut permettre de calculer des communautés globales du graphe. Un article d'état de l'art [77] présente ces méthodes ; nous proposons d'en rappeler les points essentiels et d'y ajouter quelques remarques.

Il est en général simple de passer d'une méthode de détection de communautés locales à une méthode de détection de communautés globales. Par exemple, on peut suivre la démarche de [88] où les auteurs proposent de chercher "la communauté naturelle" de chaque nœud du graphe avec leur méthode de détection de communauté locale et de retourner ces communautés afin d'obtenir une couverture du graphe (en communautés recouvrantes). À cause du coût de calcul important engendré par la détection de la communauté de chaque nœud du graphe, les auteurs proposent de se limiter à la recherche de la communauté naturelle des nœuds qui ne sont pas déjà affectés à une communauté.

Afin d'améliorer ce type d'algorithme, il est également possible de faire une sélection plus élaborée des graines initiales et également de nettoyer l'ensemble des communautés locales découvertes en sortie de l'algorithme.

Ces méthodes consistent donc généralement en trois étapes :

1. Sélection des graines : choix des nœuds ou des ensembles de nœuds par l'intermédiaire desquels une (ou des) communauté(s) locale(s) va (vont) être calculée(s).
2. Calcul des communautés locales à partir des graines sélectionnées.
3. Calcul des communautés finales à partir des communautés locales.

Certains algorithmes utilisent des graines qui sont des nœuds uniques (comme [80, 133, 160]), alors que d'autres utilisent des ensembles de nœuds pas forcément connectés (comme [76]) ou induisant un sous-graphe connecté dense (comme [34, 122]). Les méthodes cherchent en général des graines qui sont centrales aux communautés, par exemple les nœuds qui ont une centralité plus élevée que celle de leurs voisins [80], à l'exception de [78] qui cherche également des nœuds à l'intersection de communautés. Le nombre de graines peut être donné en entrée à l'algorithme, mais en général il est déterminé de manière automatique suivant une heuristique.

Dans la plupart des méthodes, les communautés finales sont simplement les communautés locales, mais parfois une étape additionnelle de nettoyage est ajoutée comme dans

[78] (utilisant du clustering d'ensemble [143]) ou dans [25]. Dans cette dernière méthode, les auteurs créent une matrice $[M_{ij}]_{N\times N}$, avec $M_{ij} = 1$ si j est un membre de la communauté locale de i et 0 sinon. Une distance entre les lignes i et j est définie de la façon suivante :

$$Distance(i, j) = n - \sum_{k=1}^{n} \delta(M_{ik}, M_{jk})$$

où $\delta(M_{ik}, M_{jk}) = 1$ si $M_{ik} = M_{jk}$ et 0 sinon.

La matrice est alors triée : pour chaque ligne i (i étant trié par ordre croissant), la ligne $j > i$ avec la distance minimum à la ligne i est déterminée, le contenu des lignes et des colonnes $i+1$ et j est alors échangé. Enfin, un clustering hiérarchique est déterminé à partir de cette matrice triée. Étant donné un paramètre $D_{\min}$, pour chaque $i = 2, \ldots, n$, les nœuds $i-1$ et i sont placés dans le même cluster si et seulement si $Distance(i-1, i) < D_{\min}$. L'histogramme des distances $Distance(i-1, i)$ peut aider à trouver manuellement un $D_{\min}$ pertinent.

2.7 Conclusion

La détection de communautés en partant d'un nœud peut donc être vue sous trois angles différents : la détection de "la communauté" du nœud d'intérêt, la détection de toutes ses communautés ou bien simplement la détection d'une communauté indépendamment du nœud d'intérêt qui est alors utilisé simplement comme un nœud d'accroche. Au contraire, lorsque l'on considère plusieurs nœuds de départ, toutes les méthodes cherchent à compléter cet ensemble de nœuds en une communauté, cela revient donc à chercher "la communauté" de l'ensemble de nœuds.

Cette détection se fait généralement en cherchant des minima locaux d'une fonction de qualité ad hoc qui quantifie la pertinence d'un groupe de nœuds en tant que communauté. De telles fonctions évaluent le plus souvent la topologie locale du réseau, par exemple en comparant le nombre de liens internes et externes à la communauté. L'optimisation est réalisée de façon gloutonne/stochastique ou en utilisant des heuristiques plus élaborées mais restant efficaces en temps de type *nibble*.

Nous remarquons également qu'aucune fonction de qualité ne fait l'unanimité et que l'optimisation, visant à trouver des minima locaux, introduit un biais mal contrôlé et peu étudié. Une fonction de qualité et son type d'optimisation vont souvent de pair et seule la comparaison du résultat final à des communautés "vérité de terrain" peut permettre de juger de la validité de la méthode.

Dans la suite nous proposons une approche orthogonale pour résoudre ces problèmes de détection de communautés en partant d'un nœud et de complétion de communauté. Nous n'allons pas travailler directement avec des fonctions de qualité, mais utiliser des mesures de proximité afin d'isoler des groupes de nœuds pertinents proches du (ou des) nœud(s) d'intérêt. Pour cela, nous allons exploiter le lien fort existant entre communauté et mesure de proximité que nous détaillons dans le prochain chapitre. En particulier, nous serons

dans un premier temps moins ambitieux : plutôt que de vouloir proposer une méthode qui marche “à coup sûr” pour détecter la (ou les) communauté(s) d'un nœud d'intérêt, nous proposons une méthode qui permet de trouver une communauté bien dessinée et cela à partir d'un nœud qui est uniquement dans cette communauté et est important en son sein. Nous allons ensuite améliorer et décliner la méthode pour construire des algorithmes efficaces permettant de résoudre les trois problèmes suivants : (i) trouver des communautés auxquelles un nœud donné appartient, (ii) compléter un ensemble de nœuds en une communauté et (iii) trouver des communautés recouvrantes dans un réseau.

Chapitre 3

Lien entre mesures de proximité et communautés

Dans ce chapitre, nous expliquons tout d'abord le lien qui existe entre mesures de proximité et communautés. Ce lien est, comme nous l'avons évoqué précédemment, très peu utilisé dans la littérature, qui privilégie l'optimisation gloutonne de fonctions de qualité. Nous dressons ensuite un état de l'art des mesures de proximité en nous focalisant sur les plus pertinentes pour la détection de communautés. Finalement nous présentons deux mesures de proximité que nous avons mises au point au cours de la thèse, l'une sans paramètre et l'autre avec paramètre.

3.1 Mesure de proximité pour détecter la communauté locale d'un nœud

Les approches consistant à optimiser des fonctions de qualité sont très largement utilisées pour la détection de communautés. Une fonction de qualité permet d'évaluer dans quelle mesure un certain groupe de nœuds est une "bonne" communauté au sein d'un graphe. Comme nous l'avons décrit dans le chapitre précédent, de telles approches se basent généralement sur une comparaison des connectivités internes et externes à un groupe de nœuds pour évaluer cette "qualité communautaire". Bien qu'elles soient très largement répandues, elles souffrent généralement de deux défauts majeurs :

1. L'optimisation de fonctions de qualité est souvent très complexe du fait de la nature non-convexe de l'espace d'optimisation. Cette optimisation devant rester efficace en temps de manière à pouvoir traiter de grands graphes, elle est souvent effectuée de manière gloutonne, en commençant uniquement avec le nœud d'intérêt dans la communauté et en ajoutant les voisins un par un. Comme l'on ne s'autorise pas à repasser par des zones de moindre qualité pour trouver de meilleures zones, les communautés obtenues peuvent donc correspondre à des minima locaux et sont en

général plus petites que ce qu'elles devraient être. [1]

2. La mise en œuvre de fonctions de qualité pour la détection de communautés est également difficile du fait de l'existence de paramètres d'échelle. Par exemple, la fonction de qualité $\frac{l_{in}}{(l_{in}+l_{out})^{\alpha}}$ étudiant le rapport entre le nombre de liens internes l_{in} à la communauté et le nombre de liens internes et externes $l_{in} + l_{out}$, repose sur un paramètre d'échelle α qui contrôle la taille de la communauté que l'on veut obtenir : elle déforme la fonction de qualité en privilégiant une certaine taille de communauté. Cela pose problème si cette taille privilégiée ne correspond pas au résultat attendu. Trouver la bonne valeur de α est souvent fastidieux (on peut par exemple chercher les communautés pour de nombreuses valeurs α différentes et retenir les communautés qui sont présentes fréquemment). De plus, cette valeur α est parfois arbitrairement fixée à 1 et disparaît alors de la fonction de qualité : $\frac{l_{in}}{(l_{in}+l_{out})}$. α devient alors un paramètre caché et on privilégie implicitement une certaine taille de communauté. Ce problème de paramètre semble très lié au théorème de l'impossibilité de clustering de Kleinberg [82] qui, en quelque sorte, montre qu'il est difficile de faire du clustering sans fixer un paramètre d'échelle.

Nous proposons de pallier les problèmes de l'approche à base de "fonction de qualité" pour la détection de communautés avec une approche à base de "mesure de proximité". Il existe en effet un lien très fort entre la notion de mesure de proximité et celle de communauté. Nous illustrons ce lien avec une mesure de proximité particulière que nous allons détailler plus loin (l'opinion propagée [52]). Nous montrons en particulier qu'une mesure de proximité peut permettre de trouver la (ou les) communauté(s) d'un nœud donné. Nous exploiterons ensuite cette idée sous différentes formes tout au long de la thèse afin de : "trouver toutes les communautés auxquelles un nœud appartient", "compléter un ensemble de nœuds en une communauté" et "trouver toutes les communautés recouvrantes d'un réseau" respectivement dans les chapitres 4, 5 et 6.

La figure 3.1 montre les résultats fournis par une mesure utilisée pour évaluer la proximité de tous les nœuds d'un réseau à un nœud d'intérêt donné. Sur la courbe montrant ces valeurs triées par ordre décroissant, on peut voir trois plateaux séparés par deux

1. Ce problème peut dans une certaine mesure être illustré par le fonctionnement de l'algorithme de Louvain [32]. Cet algorithme cherche à partitionner un graphe en communautés en optimisant de manière gloutonne une fonction de qualité, appelée *modularité* : chaque nœud constitue au départ sa propre communauté. Ensuite, l'algorithme change un par un les nœuds de communauté si cela permet d'augmenter la modularité. En faisant cela il fusionne les communautés initiales (ici réduites à un seul nœud) et arrive rapidement à un maximum local ne permettant plus d'optimiser davantage la modularité en changeant un nœud de communauté. Il construit alors un autre graphe en fusionnant les nœuds appartenant à une même communauté en hypernœud et recommence l'opération dans ce nouveau graphe (ce nouveau graphe est construit de telle manière que fusionner des hypernœuds équivaut à fusionner des communautés dans le graphe initial). Les communautés finales obtenues sont bien plus grosses et optimisent bien mieux la modularité que les communautés obtenues avant la fusion des communautés en hypernœuds. Cette fusion a permis ici de contourner certains de ces minima locaux. Ceci donne des arguments en faveur de l'assertion suivante : les communautés obtenues par "l'optimisation gloutonne simple" d'une fonction de qualité en partant d'un nœud (ou d'un petit ensemble de nœud) donne lieu à des communautés plus petites que ce qu'elles devraient être à cause des minima locaux.

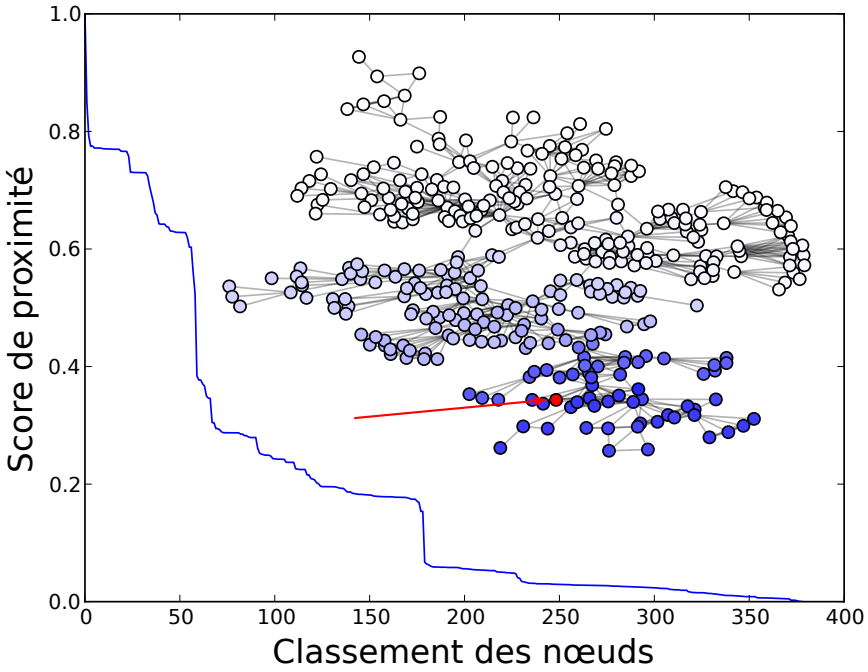


Figure 3.1 – Proximités (nous avons utilisé ici l'opinion propagée [52] que nous détaillerons plus tard dans ce chapitre) des nœuds d'un graphe à un nœud d'intérêt triées par ordre décroissant. Sur la représentation du graphe (avec un layout graphviz [57]), le nœud d'intérêt est indiqué par une flèche et l'intensité de la couleur des nœuds est proportionnelle à leur proximité au nœud d'intérêt. Le graphe utilisé est un réseau de co-auteurs comptant 379 nœuds et 914 liens [115].

décroissances fortes aux environs du 50^{e} et du 180^{e} nœud. En visualisant le graphe, on constate que les deux premiers plateaux correspondent à deux communautés à des échelles différentes, communautés que l'on aurait pu trouver intuitivement. Cela suggère que l'on puisse détecter une communauté si l'on obtient un plateau suivi d'une forte décroissance.

La figure 3.2 montre le résultat de la même expérience menée sur quatre nœuds distincts du réseau *Wikipédia 2008* (cf. annexe A pour une description du jeu de données). Cette courbe est présentée en échelle log-log pour mieux mettre en évidence les transitions. On peut voir, en faisant le lien avec l'expérience précédente (figure 3.1) que, pour la courbe transition claire et la courbe transition lente, cette mesure peut directement traduire la présence de communautés ego-centrées sur le nœud d'intérêt. Cependant, la plupart du temps on obtient des courbes en loi de puissance et en loi de puissance déformée (cf. figure 3.2). Ces lois sans échelle montrent qu'aucune taille caractéristique ne peut être extraite de cette mesure, ce qui signifie que : soit le sommet appartient à plusieurs communautés de taille différente, soit il n'appartient à aucune communauté.

Nous avons étudié manuellement les pages correspondant à ces quatre courbes :

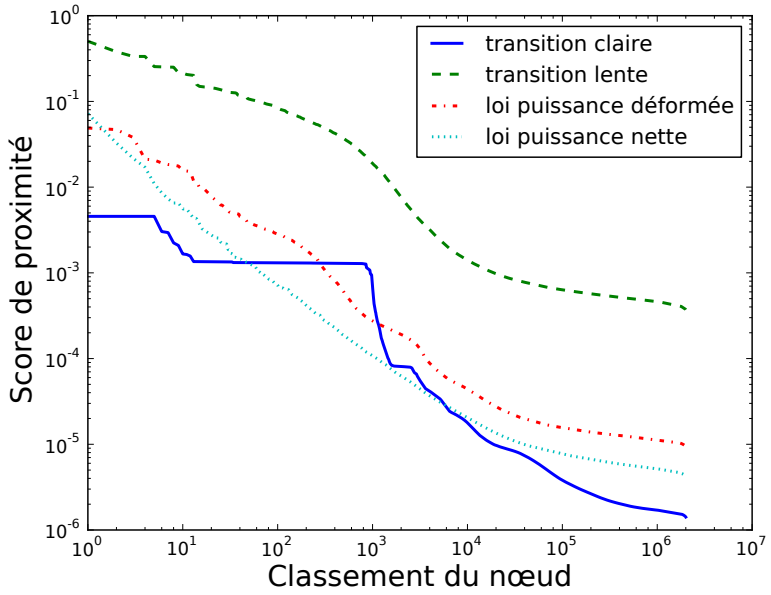


Figure 3.2 – Proximité décroissante calculée avec l'opinion propagée [52] des nœuds du réseau *Wikipédia 2008* à quatre nœuds d'intérêt distincts (échelle doublement logarithmique).

— La courbe *transition claire* correspond à la page "Cotton Township, Switzerland County, Indiana". Comme on peut le voir, les 6 premiers nœuds constituent un plateau. Ces nœuds correspondent à la page "Switzerland County, Indiana" et aux 5 autres pages des *Townships* du *Switzerland County*. On observe ensuite une décroissance sur 7 nœuds très proches sémantiquement de "Township, Switzerland County" et "Indiana". Les 970 nœuds suivants constituent un second plateau et correspondent aux autres *Townships* de l'Indiana (sans exception, l'Indiana comptant 1005 *townships*). On observe ensuite une forte décroissance sur 1000 nœuds composée de pages relatives à l'Indiana et aussi, dans une moindre mesure, à l'Illinois. On observe ensuite un plateau avec 1000 nœuds composés essentiellement des *Townships* de l'Illinois (avec quelques exceptions). La décroissance ondulante vers un plateau final transite doucement vers des pages plus éloignées du sujet en passant par d'autres *Townships* américains et d'autres sujets liés à l'Amérique.
— La courbe *transition lente* correspond à la page "Mafia". Les premiers milliers de pages sont dédiées à des mafiosi et à des sujets connexes au crime organisé. On observe ensuite une transition douce vers des sujets plus éloignés.
— La courbe *loi de puissance déformée* est obtenue avec la page "Mi-Hyun Kim" qui est une joueuse de golf professionnelle de Corée du Sud. La page est principalement liée à des pages sur le golf et à des pages relatives à la Corée du Sud. Les mille premières pages traitent de l'un ou l'autre de ces sujets.
— La courbe *loi de puissance nette* est obtenue pour la page "JNCO", qui est une

transition claire	transition lente	loi puissance déformée	loi puissance nette
73‰	208‰	610‰	109‰

Tableau 3.1 – Proportion de nœuds donnant les différentes courbes. L'étude est menée sur 1000 nœuds choisis au hasard dans le réseau *Wikipédia 2008* en utilisant l'opinion propagée comme mesure de proximité. L'attribution à une classe est ici réalisée à la main.

marque de vêtements. On observe ici une loi de puissance quasi-parfaite débouchant sur un plateau final. Cette page est liée à des pages très différentes les unes des autres, telles que : "Los Angeles", "Jeans", "Hip-hop", "J.C. Penney", "Graffiti", "Kangaroo", "Boxing" et "Nu Metal". Les pages les mieux classées n'appartiennent pas à un thème précis.

On voit donc ici que, dans le même réseau coexistent différents types de communautés :

1. des communautés bien définies, comme celle de Switzerland county ou celle de l'Indiana.
2. des communautés définies de manière plus floue comme celle de Mafia.

Ces communautés peuvent aussi être observées à plusieurs échelles : Switzerland county est ainsi une sous-communauté de celle de l'Indiana.

Dans un même réseau coexistent aussi deux grands types de nœuds :

1. des nœuds qui peuvent par eux-mêmes caractériser une (ou des) communauté(s) comme "Cotton Township, Switzerland County, Indiana" ou "Mafia" ;
2. des nœuds qui appartiennent à plusieurs communautés : quelques-unes comme "Mi-Hyun Kim" ou de nombreuses comme "JNCO" et ne peuvent pas caractériser une communauté à eux seuls.

Bien sûr, la frontière entre ces différents types de communautés ou de nœuds n'est pas clairement identifiée. On peut néanmoins attribuer une classe à un nœud en analysant la décroissance des proximités et ainsi savoir grossièrement s'il appartient à une, quelques-unes ou de nombreuses communautés et voir si la (les) communauté(s) est (sont) clairement dessinée(s) ou plutôt floue(s). Nous avons remarqué que le plus souvent (plus de 70% des cas), une loi sans échelle déformée ou nette est obtenue, c'est-à-dire que la plupart des nœuds appartiennent à plusieurs communautés définies de manière floue, cf. tableau 3.1 où nous avons classifié à la main les courbes obtenues pour 1000 nœuds choisis aléatoirement dans le réseau.

La figure 3.3, montre différents attributs des courbes : la pente maximale, la pente moyenne et l'écart-type de la pente pour les courbes obtenues par ces 1000 nœuds, ainsi qu'une Analyse en Composantes Principales (ACP). Nous voyons que l'écart-type et la pente maximale permettent, tous deux indépendamment, de discriminer de manière correcte les lois en puissance des allures en "plateau / décroissance / plateau" : prendre par exemple un écart-type supérieur à 0.6, ou une pente maximale supérieure à 3, permet d'identifier de manière correcte les courbes en "plateau / décroissance / plateau".

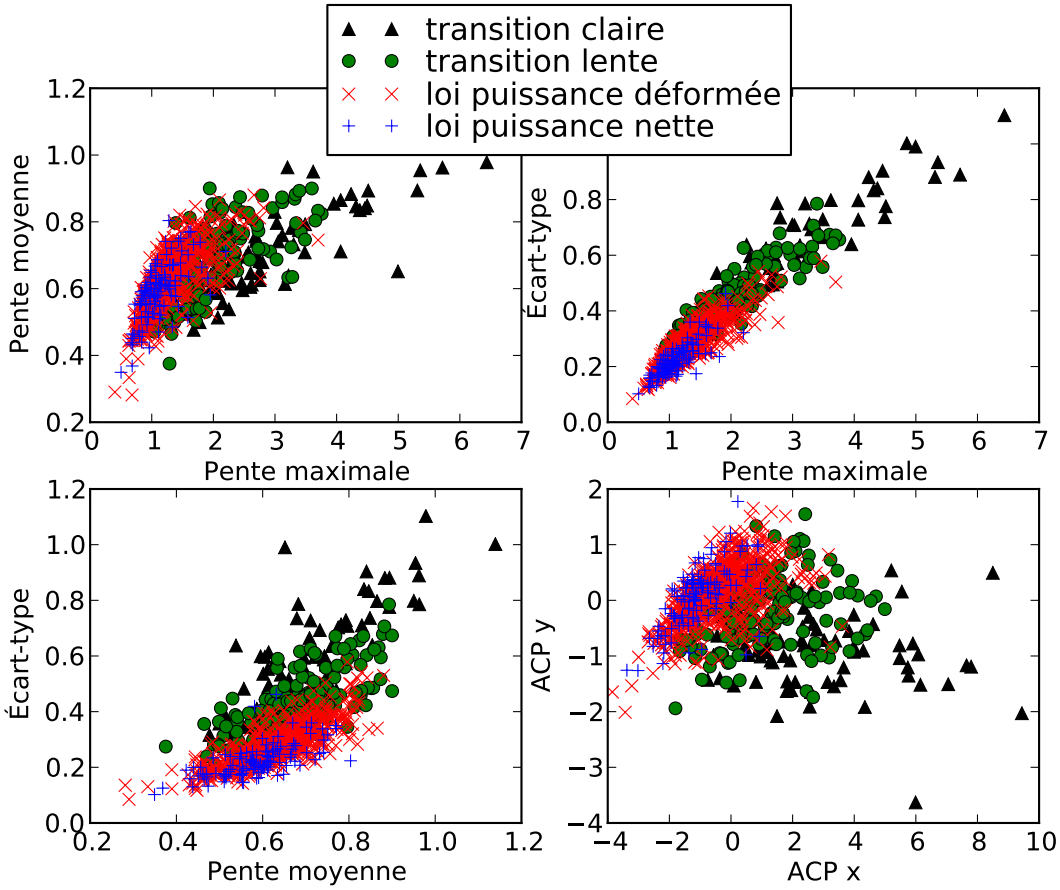


Figure 3.3 – Nœuds donnant les différentes courbes. L'étude est menée sur 1000 nœuds choisis au hasard dans le réseau *Wikipédia 2008* et en utilisant l'opinion propagée comme mesure de proximité. L'attribution à une classe est ici réalisée à la main.

3.1.1 Loi sans échelle : trop de communautés ou pas de communauté ?

Nous avons dit précédemment qu'une courbe d'allure de loi sans échelle signifie que le nœud n'appartient à aucune communauté ou bien à plusieurs communautés. On ne peut alors pas extraire d'échelle caractéristique de la mesure. L'objet de cette section est de justifier cette intuition à travers des expériences réalisées sur des graphes jouets à structure communautaire connue.

Nous avons mis en place un modèle de graphes spécifique pour émuler ce phénomène. Nous générons un graphe composé de quatre graphes d'Erdos-Renyi dont les probabilités internes de connexions sont les suivantes : 0.4, 0.3, 0.2 et 0.1 et les nombres de nœuds sont respectivement : 20, 40, 80 et 160. Les quatre graphes d'Erdos-Renyi se recouvrent sur un unique nœud, notre nœud d'intérêt ; ils représentent quatre communautés de ce nœud, de taille et de densité différentes. Nous mesurons ensuite la proximité de tous les nœuds

du réseau au nœud d'intérêt. La figure 3.4a représente le réseau en coloriant les nœuds en fonction de leur score selon une échelle logarithmique, tandis que la figure 3.4b montre la décroissance des scores de proximité en échelle log-log. Les "plateau / décroissance / plateau"sont ici évidents et la structure communautaire peut être reconstituée à la perfection en coupant aux plus grandes pentes. Nous avons répété l'expérience sur ce même réseau en y ajoutant du bruit : nous avons ajouté chaque lien possible (et enlevé chaque lien existant) avec une probabilité de 0.01 (résultats détaillés figure 3.4c et 3.4d) et avec une probabilité de 0.03 (résultats détaillés dans les figures 3.4e et 3.4f). Nous voyons que l'allure "plateau / décroissance / plateau"s'estompe pour laisser place à une courbe allant vers une loi puissance.

Nous verrons par la suite comment pallier ce problème de loi sans échelle, mais nous proposons d'abord de faire un état de l'art des mesures de proximité et de présenter les avantages et les inconvénients que chaque mesure pourrait avoir pour la détection de communautés.

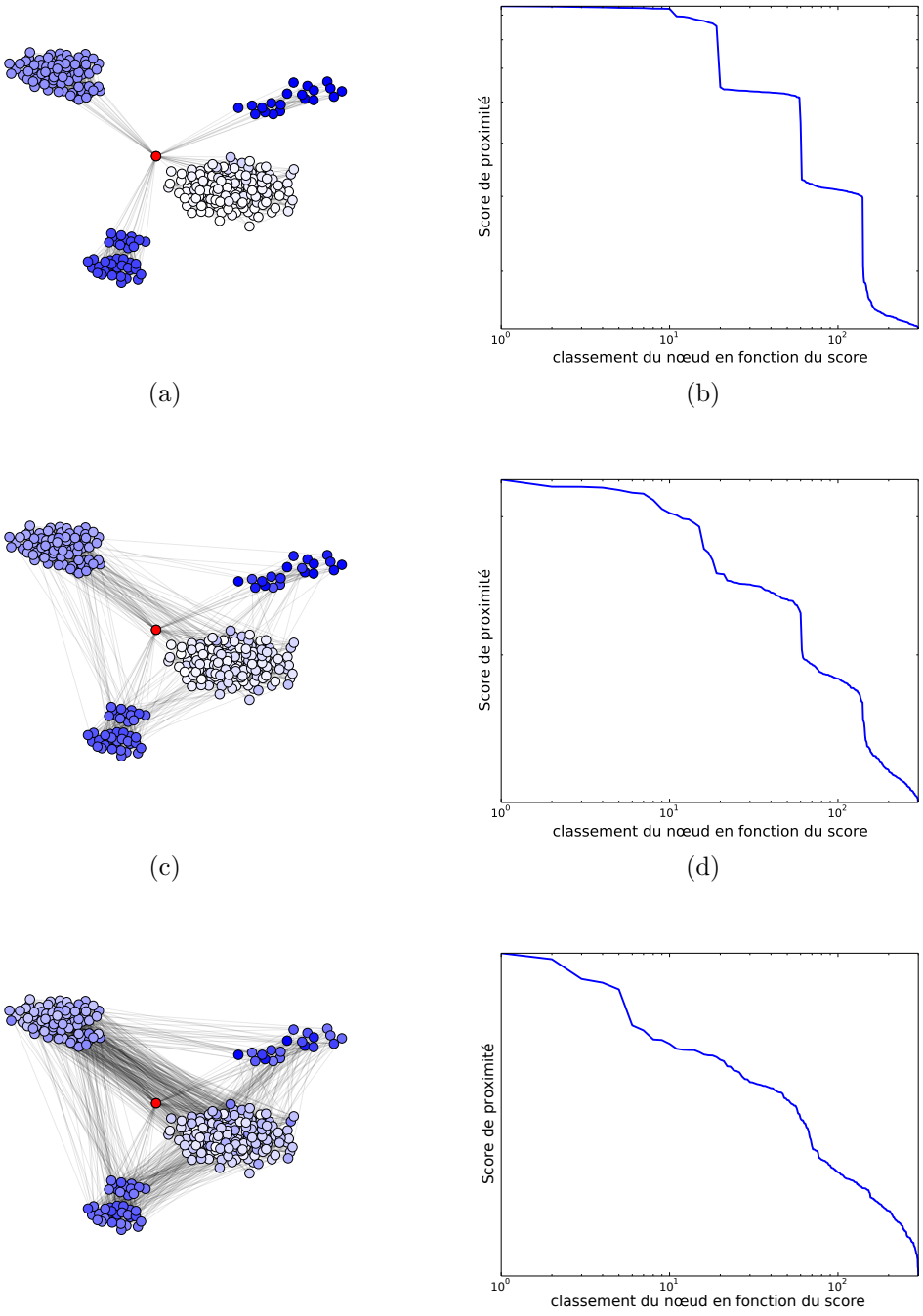
3.2 État de l'art sur les mesures de proximité

Les mesures de proximité que nous considérons ici sont des fonctions prenant en entrée un graphe et deux nœuds et retournant une quantité permettant d'évaluer combien ces deux nœuds sont proches l'un de l'autre. Étonnamment le lien entre mesure de proximité et communauté que nous venons de mettre en évidence n'est pas (ou très peu) exploité dans la littérature. Les mesures de proximité ont pourtant de nombreuses autres applications, bien plus répandues, pour l'étude des réseaux, notamment en ce qui concerne la prédiction de liens : *intuitivement deux nœuds non-connectés, mais qui sont évalués comme très proches l'un de l'autre par une mesure de proximité ad hoc ont plus de chance de se connecter dans le futur que s'ils étaient évalués comme éloignés l'un de l'autre.* La prédiction de liens est si importante qu'elle constitue un domaine à part entière de l'analyse des réseaux. L'une de ses applications les plus populaires est la recommandation, par exemple la suggestion d'amis sur un réseau social ou de produits dans les boutiques en ligne. Il existe ainsi plusieurs articles d'état de l'art sur les mesures de proximité, dont certains sont généraux [44] et d'autres appliqués à la prédiction de liens [161, 94].

Nous reprenons et étendons ici ces états de l'art en détaillant les mesures de proximité qui nous semblent être les plus pertinentes pour la détection de communautés. Les deux mesures de proximités les plus immédiates sont sans doute :

- la distance entre les deux nœuds, c'est-à-dire le nombre minimal de sauts pour aller de l'un à l'autre et
- le nombre de voisins partagés par les deux nœuds, ou bien des variantes prenant en compte seulement le voisinage des deux nœuds (comme la similarité de Jaccard entre deux ensembles de voisins [2] ou la similarité d'Adamic-Adar [8] qui donne souvent

2. Pour deux nœuds a et b elle est donnée par : $J(a,b) = \frac{|N_a \cap N_b|}{|N_a \cup N_b|}$ où N_a (resp. N_b) est l'ensemble des voisins de a (resp. b).



(a) (b) (c) (d) (e) (f)

Figure 3.4 – Proximité décroissante (nous avons utilisé ici l'opinion propagée [52] que nous détaillerons plus tard dans ce chapitre) des nœuds d'un graphe jouet à un nœud d'intérêt. Sur la représentation du graphe (avec un layout graphviz [57]), le nœud d'intérêt est au centre et l'intensité de la couleur des nœuds est proportionnelle à leur proximité au nœud d'intérêt.

les meilleurs résultats dans le cadre de la prédiction de liens).
Ces deux mesures de proximité sont intéressantes, mais néanmoins trop simples. En effet (comme nous l'avons vu précédemment) la première donne des valeurs entières qui sont en général très faibles et identiques pour tous les nœuds. De plus, elle ne prend pas en compte la redondance des liens : deux nœuds à distance 2 partageant un seul voisin auront la même proximité que deux nœuds à distance 2 partageant un grand nombre de voisins. Avec le deuxième type de mesure de proximité, l'effet inverse se produit. En effet, en utilisant le même exemple, deux nœuds à distance 2 partageant un grand nombre de voisins auront une plus grande proximité que deux nœuds à distance 2 n'en partageant qu'un seul, ainsi la redondance est bien prise en compte. Par contre, toute paire de nœuds à distance supérieure à 2 aura une proximité nulle, ce qui n'est pas satisfaisant.

La construction de mesures de proximité plus élaborées prenant en compte la distance et la redondance s'impose donc. Voici les mesures de proximité que nous avons jugées les plus pertinentes :

— l'indice de Katz [79] :

$$\mathrm{Katz}(i,j) = \sum_{l=0}^{\infty} \beta^l N_l^{\mathrm{SP}}(i,j)$$

où N_l^{SP} est le nombre de chemins (simples, c'est-à-dire sans boucle) de longueur l entre les deux nœuds i et j. N_l^{SP} est cependant souvent trop long à calculer sur de grands graphes pour $l \geq 4$. Nous verrons qu'utiliser les NBP (Non-Bactracking Paths), c'est-à-dire le nombre de chemins sans boucle de longueur 2, mais pouvant avoir des boucles plus grandes permet de raisonnablement pallier ce problème. Il existe également une variante, le Local Path Index, ne prenant en compte que les chemins de longueur inférieure à un paramètre λ [101].

— Le pagerank enraciné [119] déjà cité précédemment : soit un marcheur aléatoire qui commence à marcher depuis un nœud d'intérêt en allant à chaque fois aléatoirement vers un voisin et en se téléportant au nœud de départ avec la probabilité α. La probabilité de trouver le marcheur au temps infini situé sur un nœud donné correspond à la proximité de ce nœud au nœud d'intérêt. On remarque que cette mesure de proximité n'est pas symétrique. De plus, cette mesure de proximité entre un nœud d'intérêt et tous les nœuds du graphe peut être obtenue avec l'algorithme de point fixe suivant :

$$X_{t+1} = (1-\alpha)TX_t + \alpha X_0$$

où X_t est *le vecteur de scores* après n itérations ($X_t[i]$ correspond à la probabilité d'avoir un marcheur au nœud i après t pas), X_0 est initialisé avec le vecteur nul sauf pour le nœud d'intérêt qui est initialisé à 1. T est la matrice de transition : $T_{kl} = \frac{l_{kl}}{d_l}$, où l_{kl} est le poids du lien entre les nœuds k et l, et d_l est le degré du nœud l. $\alpha \in]0,1[$ est un paramètre qui contrôle la profondeur de l'exploration du réseau. Le temps de calcul nécessaire est essentiellement linéaire en $O(t(n+m))$ où n et m sont respectivement le nombre de nœuds et de liens dans le graphe et t le

nombre d'itérations nécessaires à la convergence (en général de l'ordre du diamètre du graphe, c'est-à-dire en $O(\log(n))$). Bien que cette méthode soit déjà rapide, il est possible de la rendre encore plus rapide dans certains cas. En particulier, lorsque le coefficient α est grand, on peut en général obtenir, pour un graphe creux, une approximation en un temps constant ne dépendant pas de la taille du graphe [18, 150]. Pour cette raison, nous utiliserons cette mesure de proximité dans le chapitre 6 lorsque le temps de calcul sera le principal facteur limitant.

— Le hitting time (resp. commuting time). Pour un nœud source et un nœud cible donnés, c'est le nombre moyen de pas nécessaires à un marcheur pour aller (resp. pour aller et revenir) de la source à la cible. Pour un nœud cible donné, le hitting time pour chaque autre nœud du graphe vu tour à tour comme source peut être calculé en résolvant le système linéaire suivant, comme détaillé dans [104] :

$$H_i(j) = \begin{cases} 0 & \text{si } j = i \\ 1 + \sum_{k \in V(j)} H_i(k) & \text{sinon.} \end{cases}$$

$H_i(j)$ correspond au hitting time pour le nœud source j et le nœud cible i et $V(j)$ dénote l'ensemble des voisins du nœud j. Le temps de convergence de l'algorithme du point fixe utilisé pour calculer le rooted pagerank étant trop long pour la résolution de cet autre système linéaire, nous avons implémenté un algorithme qui permet de le résoudre en un temps essentiellement linéaire à l'aide de la méthode du gradient conjugué [73] vue comme une méthode itérative et pré-conditionnée astucieusement [3].

— Une autre mesure de proximité paramétrée est le pagerank enraciné biaisé par le degré de [24] ou par des attributs des liens ou des nœuds. Dans ce cas le poids des liens du réseau utilisé pour calculer le pagerank enraciné dépend des attributs et cette dépendance peut être apprise de façon supervisée.

Toutes ces mesures de proximité peuvent permettre de mettre en évidence le lien entre communautés et proximité comme nous l'avons fait précédemment. Cependant le rooted pagerank et l'indice de Katz sont paramétrés et dans certains cas il est souhaitable de pouvoir s'affranchir de ces paramètres ou bien de savoir comment les fixer. De plus, nous verrons plus loin dans ce chapitre, que ces paramètres ne sont pas toujours bien adaptés. Le calcul du hitting time, quant à lui, est plutôt compliqué bien que rapide grâce à notre algorithme à base de descente par gradient conjugué.

Pour pallier ces problèmes (temps de calcul et pertinence des paramètres le cas échéant) nous proposons deux nouvelles mesures de proximité.

— La première sans paramètre est intuitive, simple à calculer et permet dans la plupart des cas d'arriver à des résultats très satisfaisants. Elle est basée sur la dynamique d'opinion, nous l'appelons l'*opinion propagée*.

— La deuxième mesure que nous proposons incorpore principalement deux paramètres qui permettent de faire indépendamment et simultanément un compromis entre

3. le code est disponible à l'url donné en introduction : `http://bit.ly/maxdan94`

"distance et redondance" d'une part et un compromis entre "popularité et intimité" d'autre part. Compromis qui, comme nous allons le voir, nous semblent nécessaires et ne semblent pas pouvoir être gérés par les mesures de proximité de l'état de l'art comportant un seul paramètre. Notre deuxième mesure de proximité peut être vue comme une variante de l'indice de Katz. Notre mesure, que nous appelons *Katz+*, permet, contrairement à la mesure de Katz, de traiter de grands graphes et de gérer les deux compromis.

Ce sont ces deux mesures de proximité que nous allons privilégier dans le reste du manuscrit bien que d'autres soient utilisables et donnent des résultats également pertinents (moyennant souvent un temps de calcul accru).

3.3 L'opinion propagée : une mesure de proximité sans paramètre

Nous proposons ici une nouvelle mesure de proximité basée sur la dynamique d'opinion. Elle permet de calculer rapidement la proximité de tous les nœuds du graphe à un nœud d'intérêt donné. Dans ce qui suit, nous ferons l'amalgame entre proximité au nœud d'intérêt et un réel compris entre 0 et 1 qui illustre l'opinion d'un nœud.

Étant donné un nœud d'intérêt, le principe du calcul de notre mesure de proximité consiste, dans un premier temps, à fixer la valeur de l'opinion du nœud d'intérêt à 1 et l'opinion de tous les autres nœuds à 0. Puis, à chaque pas de temps, l'opinion de chaque nœud est mise à jour avec la moyenne de l'opinion de ses voisins et l'opinion du nœud d'intérêt est remise à 1 (elle ne change donc pas tout au long du processus). Ce processus converge vers une opinion de 1 pour tous les nœuds du graphe (si celui-ci est connexe). Ce n'est pas cet état final connu qui nous intéresse pour calculer la proximité des nœuds du graphe au nœud d'intérêt, mais un état transitoire suffisamment avancé. Il faut, par conséquent, s'arrêter à un nombre d'itérations t suffisamment grand, mais avant d'obtenir 1 pour chaque nœud du graphe. Ainsi l'opinion de 1 du nœud d'intérêt se sera propagée dans le réseau et les opinions des autre nœuds du réseau (qui étaient initialement de 0) auront augmenté vers des valeurs plus grandes illustrant leur proximité au nœud d'intérêt.

Nous allons maintenant montrer qu'un large intervalle est possible pour le choix du paramètre t sans induire de grands changements ; nous proposons également une méthode pour nous affranchir complètement de ce paramètre. Nous proposons pour cela deux conjectures que nous validons ensuite empiriquement :

Conjecture 1 *Après un nombre d'itérations suffisamment grand, le classement des nœuds en fonction de leur opinion ne change plus.*

Conjecture 2 *Après un nombre d'itérations suffisamment grand, la différence entre les opinions d'une paire de nœuds est proportionnelle à la différence entre les opinions de*

n'importe quelle autre paire de nœuds[4].

Les deux conjectures suggèrent simplement que, pour quatre nœuds a, b, c et d avec une opinion à l'itération t notée respectivement O_a^t, O_b^t, O_c^t et O_d^t, nous avons :

$$\lim_{t\to\infty} \frac{O_a^t - O_b^t}{O_c^t - O_d^t} = C_{a,b,c,d}$$

où $C_{a,b,c,d}$ est une constante dépendant seulement des nœuds a, b, c et d.

Nous proposons une troisième conjecture :

Conjecture 3 *Ce nombre d'itérations est petit devant la taille du graphe.*

Ces conjectures ont été testées sur différents graphes générés et réels avec des résultats concluants. Nous montrons ici, sur la figure 3.5, les résultats obtenus sur le réseau *blogs politiques* de [9]. Comme on peut le voir, après quelques itérations, le classement ne change plus et les différences entre les opinions deviennent effectivement proportionnelles.

Pour s'affranchir du paramètre t, il est possible de remettre à l'échelle les opinions en dilatant les valeurs de manière à ce que l'opinion la plus faible soit 0 et la plus forte (celle du nœud d'intérêt) reste à 1. Des scores entre 0 et 1 sont donc obtenus à chaque itération et le processus converge alors vers un point fixe.

Le nœud d'intérêt étant étiqueté u, le calcul consiste donc à répéter de manière itérative les trois étapes suivantes jusqu'à la convergence :

$$\begin{array}{lcll} C^t(u) & = & MC^{t-1}(u) & \text{MOYENNAGE} \\ C^t(u) & = & \frac{C^t(u) - min(C^t(u))}{1 - min(C^t(u))} & \text{MISE A L'ÉCHELLE} \\ C_u^t(u) & = & 1 & \text{RÉINITIALISATION} \end{array}$$

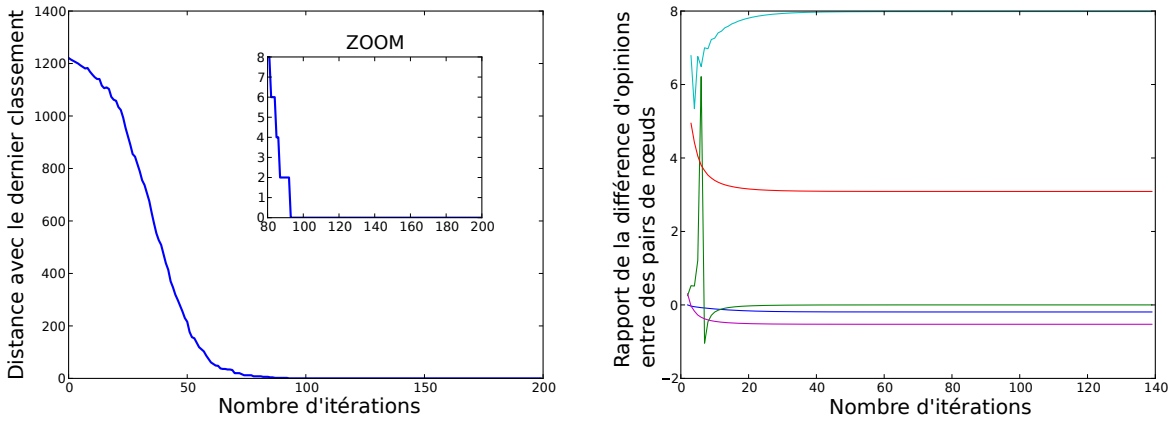
où :

- $C^t(u)$ est le vecteur d'opinions après t itérations pour le nœud d'intérêt u. La composante j du vecteur $C^t(u)$ est notée $C_j^t(u)$;
- $C^0(u)$ est le vecteur d'opinions initial. Il est nul sauf pour le sommet u, pour qui il est fixé à 1 : $C_u^0(u) = 1$;
- M est la matrice de moyennage, qui est la transposée de la matrice stochastique de transition : $M_{kl} = \frac{l_{kl}}{d_k}$, où l_{kl} est le poids du lien (k, l) et d_k le degré du sommet k.

3.3.1 Résultat sur des réseaux réels

Nous avons appliqué l'opinion propagée à plusieurs petits réseaux réels. Nous avons détaillé les résultats dans la figure 3.6 pour le réseau *netscience* (détaillé dans l'annexe A) : la figure 3.6a montre les valeurs obtenues toutes les 10 itérations sans l'étape de mise à l'échelle. Bien que les valeurs convergent vers 1, nous voyons que n'importe quel nombre d'itérations entre 20 et 100, voire plus, donne des valeurs qui permettent de bien rendre compte de la proximité au nœud d'intérêt et de découvrir la structure communautaire. La

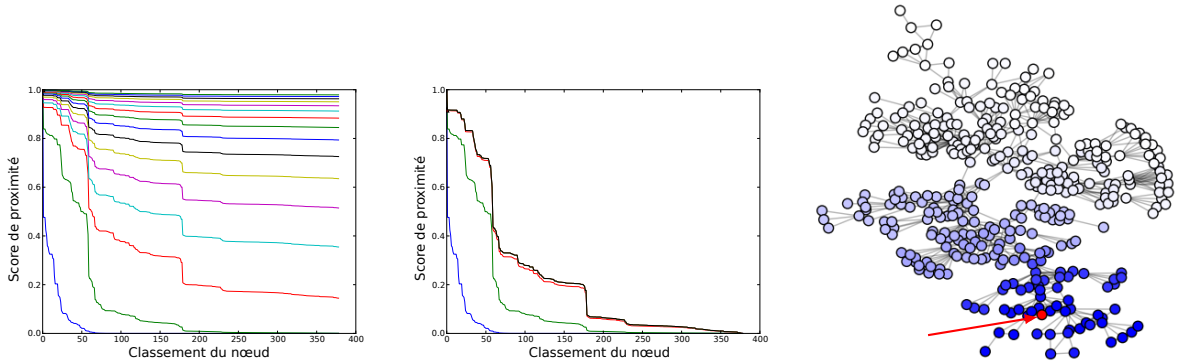
4. Bien que la conjecture 2 implique la conjecture 1, nous les avons séparées pour une meilleure clarté du manuscrit.



(a) (b)

Figure 3.5 – Expériences validant les conjectures 1 et 2. Elles ont été menées sur le réseau polblog asymétrisé [9] (cf. annexe A). La figure 3.5a valide la conjecture 1 en comparant, pour l'opinion propagée, les classements obtenus à chaque itération au dernier classement obtenu (pour 200 itérations). On peut voir que, après seulement 95 itérations, le classement ne change plus. La distance utilisée pour comparer les classements est simplement le nombre de nœuds mal classés. La figure 3.5b valide la conjecture 2 en montrant le ratio de la différence entre les scores de deux paires de nœuds choisies aléatoirement. Nous avons réalisé cinq fois cette même expérience ce qui a donné lieu à cinq courbes. Comme on peut le voir, après 40 itérations les ratios deviennent constants, ainsi les différences sont proportionnelles.

figure 3.6b montre le même résultat obtenu après 10, 20, 100 et 10 000 itérations mais cette fois avec l'étape de mise à l'échelle. Nous voyons qu'après 100 itérations la méthode a déjà pratiquement convergé. La figure 3.6c montre un dessin du réseau avec les nœuds colorés en fonction de leur proximité au nœud d'intérêt obtenue par la méthode avec l'étape de mise à l'échelle.



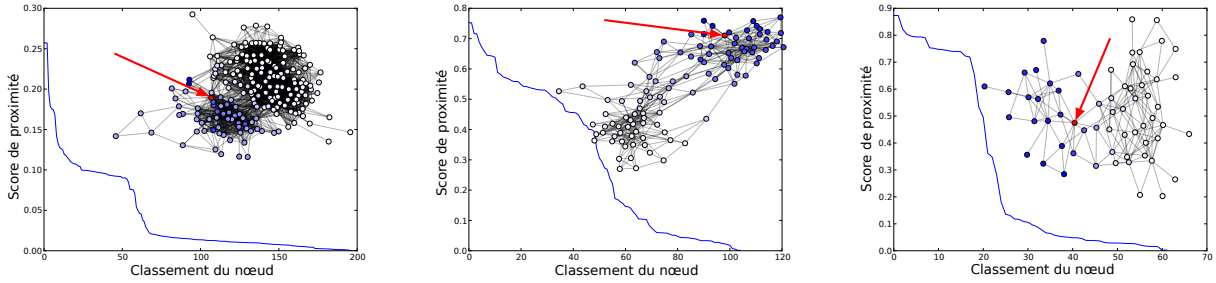
(a) (b) (c)

Figure 3.6 – Résultats de la convergence de l'opinion propagée sur le réseau *netscience*. La figure 3.6a est sans mise à l'échelle, 3.6b est avec mise à l'échelle. Sur le graphe une flèche montre le nœud d'intérêt et plus un nœud est foncé plus son score est élevé (la proximité obtenue par l'opinion propagée mise à l'échelle est utilisée ici pour ce score). Le dessin du réseau, sur la figure 3.6c, est réalisé avec un layout Graphviz [57]. Sur de petits réseaux une simple échelle linéaire peut être utilisée.

La figure 3.7 présente le résultat final obtenu avec l'étape de mise à l'échelle pour trois autres petits réseaux réels : *joueurs de jazz*, *livres politiques* et *dauphins* (détaillés en annexe). Une fois encore, les valeurs de proximité obtenues sont intuitivement bonnes et permettent à chaque fois de trouver la communauté souhaitée du nœud d'intérêt en coupant, par exemple, à la plus grande pente.

Nous avons enfin testé la mesure de proximité avec l'étape de mise à l'échelle sur le réseau des blogs politiques qui est un peu plus grand et pour lequel nous possédons une vérité de terrain, ici une connaissance des communautés existantes fournie par des experts. Le jeu de données est composé de 759 blogs politiques libéraux (ou étiquetés comme tels par des experts) et 443 blogs conservateurs. Sur des réseaux de cette taille il devient nécessaire d'étudier les décroissances en échelle logarithmique car les transitions brutales peuvent passer inaperçues en échelle linéaire. La figure 3.8 rend compte de la convergence. Après 40 itérations, les valeurs de proximité au nœud d'intérêt triées par ordre décroissant dessinent une structure en "plateau / décroissance / plateau". Cette décroissance se produit autour du 600^{e} nœud.

Sur la figure 3.9, nous avons également dessiné le réseau à l'aide de la représentation de [68], en utilisant des cercles (resp. des carrés) pour les blogs libéraux (resp. conservateurs). Nous avons choisi le nœud d'intérêt aléatoirement : il s'agit d'un blog libéral indiqué par une



(a) (b) (c)

Figure 3.7 – Résultats de la convergence de l'opinion propagée pour, figure 3.7a le réseau *joueurs de jazz*, figure 3.7b le réseau *livres politiques* et figure 3.7c le réseau *dauphins*. Sur le dessin une flèche montre le nœud d'intérêt et plus un nœud est foncé plus son score est élevé. Les dessins sont réalisés avec le layout Graphviz.

flèche. Sur ce dessin nous avons également coloré les nœuds en fonction de leur proximité au nœud d'intérêt suivant une échelle logarithmique : plus un nœud est foncé, plus son score est élevé. Comme on peut le voir, la couleur d'un nœud est, en grande majorité, en accord avec son étiquette : la plupart des blogs libéraux sont colorés en bleu et la plupart des blogs conservateurs en blanc. Il y a 561 blogs libéraux parmi les 600 nœuds ayant la proximité la plus grande au nœud libéral d'intérêt, i.e., 93.5% des 600 nœuds les mieux classés sont libéraux, ce qui est cohérent puisque le nœud d'intérêt est un blog libéral ; 617 nœuds libéraux apparaissent parmi les 759 premiers nœuds, i.e., $81,4\%$ des 759 nœuds de plus forte proximité sont libéraux.

3.3.2 Résultat sur des réseaux synthétiques

Les tests sur les réseaux réels ont montré la pertinence de l'idée générale de la méthode, les expériences sur des réseaux synthétiques que nous présentons maintenant permettent de tester plus précisément certains points spécifiques de la méthode.

La figure 3.10a montre les résultats d'une expérience menée sur un réseau synthétique composé de 3 sous-graphes aléatoires d'Erdos-Renyi contenant chacun 100 nœuds avec une probabilité de connexion de 0.3. Les nœuds appartenant à des sous-graphes différents sont connectés avec une probabilité plus faible de 0.05. Ainsi nous obtenons trois communautés assez clairement identifiables.

Nous voyons ici que les résultats obtenus par l'opinion propagée ne sont pas toujours ceux attendus : des sommets hors de la communauté du nœud d'intérêt obtiennent des valeurs de proximité très élevées. Ces sommets sont les voisins du nœud d'intérêt dans le graphe (i.e., des nœuds adjacents au nœud d'intérêt) et donc, indépendamment du fait que les voisins soient ou pas dans la même communauté que le nœud d'intérêt, ils obtiennent un score élevé. En fait, la valeur obtenue par un voisin du nœud d'intérêt de degré d est au minimum $\frac{1}{d}$ du fait de l'étape de moyennage et du fait que le nœud d'intérêt a une opinion

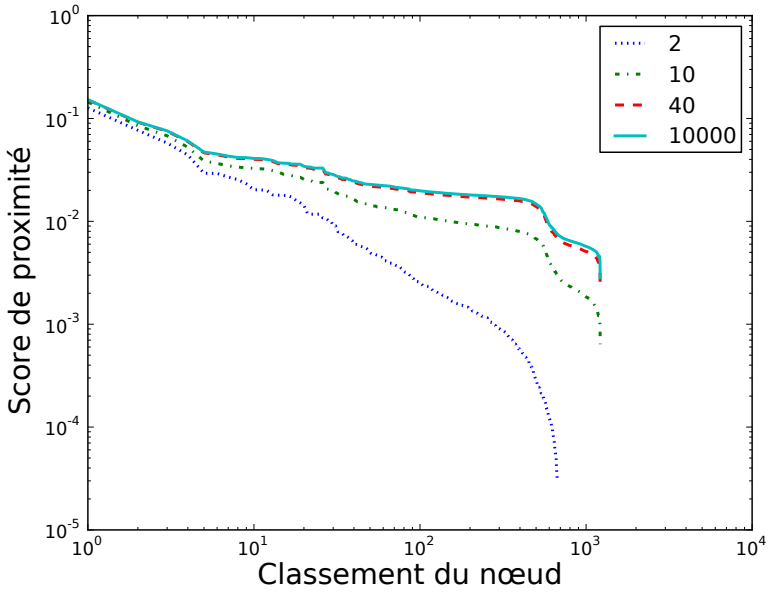


Figure 3.8 – Expériences montrant la convergence de l'opinion propagée. Les expériences sont réalisées sur le réseau *blogs politiques* (détaillé dans l'annexe A), pour lequel nous avons sélectionné un nœud d'intérêt au hasard. La courbe montre le score de chaque nœud en fonction de son classement après 2, 10, 40 et 10 000 itérations. Bien que l'ordre des nœuds change légèrement pendant les cents premières itérations, comme montré sur la figure 3.5a, les changements sont négligeables après 40 itérations.

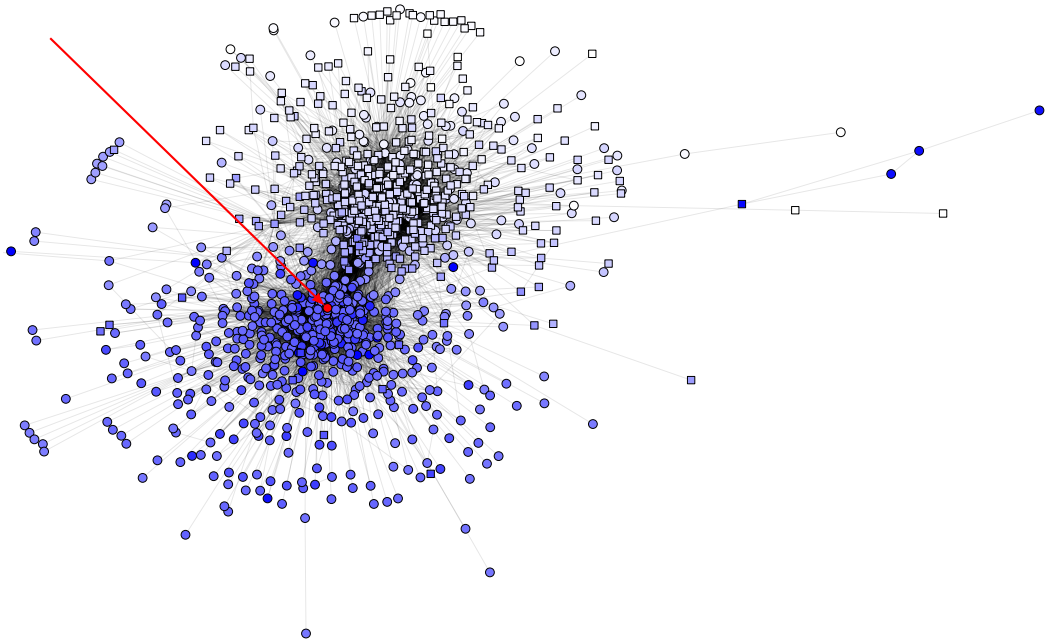


Figure 3.9 – Représentation du réseau de blogs politiques avec l'algorithme de [68]. Les cercles (resp. carrés) représentent des blogs libéraux (resp. conservateurs). Une flèche indique le nœud d'intérêt et, plus un nœud est bleu plus son score est élevé, en suivant une échelle logarithmique.

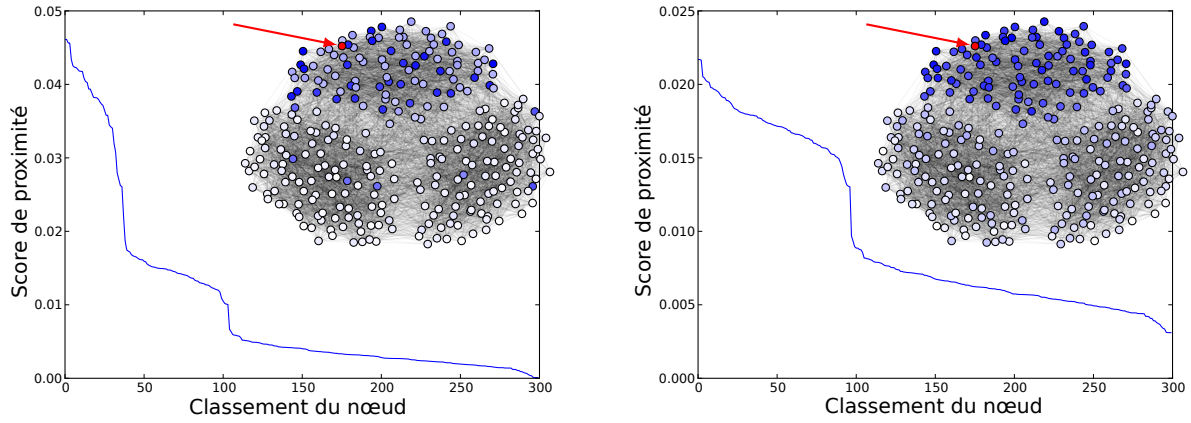
de 1 et que celle des autres voisins du nœud en question est supérieure à 0.

Cependant, on peut argumenter que ce défaut n'en est pas vraiment un ; en effet, on veut que les nœuds qui partagent une communauté avec le nœud d'intérêt aient un score élevé et, en ce sens, un nœud adjacent au nœud d'intérêt constitue avec ce dernier une communauté de deux nœuds et il est normal qu'il ait un score élevé. Cet effet peut cependant être facilement éliminé, comme montré sur la figure 3.10b, en ajoutant un pas supplémentaire après convergence qui consiste à enlever le nœud d'intérêt et à appliquer une étape de moyennage supplémentaire (la valeur d'un nœud devient la valeur moyenne de celle de ses voisins). Cet effet concerne seulement les premiers voisins et on peut donc uniquement changer le score des premiers voisins (i.e., les nœuds adjacents) avec la formule :

$$S = (S - \frac{1}{d})\frac{d}{d-1},$$

où S désigne l'opinion propagée d'un premier voisin et d son degré.

Nous venons de mettre ici en évidence deux effets de la mesure d'opinion propagée et, plus généralement, deux effets à prendre en compte dans une mesure de proximité :



(a) (b)

Figure 3.10 – 3.10a : Résultats obtenus pour un graphe composé de 3 graphes d'Erdos-Renyi (100,0.3), avec une probabilité de connexion entre des graphes appartenant à différent graphes d'Erdos-Renyi de 0.05. 3.10b : même résultat, mais avec une étape de moyennage additionnelle en enlevant le nœud d'intérêt. Plus le score est élevé, plus le nœud est foncé.

1. "un effet de distance" : plus un nœud est proche du nœud d'intérêt, en termes de distance (nombre de sauts), plus son opinion propagée est élevée, et
2. "un effet de redondance" : plus il existe de courts chemins entre un nœud donné et le nœud d'intérêt, plus son opinion propagée est élevée.

Nous venons également de voir l'effet important de la distance, pour certains jeux de données, qui avantage fortement les voisins. Ce biais peut facilement être éliminé en ajoutant un pas correcteur. La question est de savoir si le problème persiste pour les voisins à distance supérieure ou égale à 2. Pour vérifier cela, nous avons représenté les valeurs moyennes de l'opinion propagée en fonction de la distance des nœuds pour le réseau *Wikipédia 2008* (le nœud d'intérêt choisi étant la page "Boxing") et un graphe d'Erdos-Renyi ayant le même degré moyen. Comme montré sur la figure 3.11, il y a une décroissance exponentielle des valeurs de l'opinion propagée en fonction de la distance pour le graphe d'Erdos-Renyi alors que, sur le réseau Wikipédia, il y a une corrélation entre distance et opinion propagée uniquement pour les voisins du nœud d'intérêt, i.e., les nœuds à distance 1. Cela montre que l'effet dû à la distance prévaut pour le graphe d'Erdos-Renyi (où il n'y a, a priori, pas d'effet de redondance dû à la présence de communautés). Dans le graphe Wikipédia, cet effet n'existe que pour les nœuds à distance 1, c'est-à-dire les nœuds adjacents.

Enfin, il faut noter que la structure en "plateau / décroissance / plateau", telle qu'observée dans les figures 3.7 et 3.10, n'apparaît pas tout le temps. Cela dépend en majorité de deux facteurs :

1. la position du nœud d'intérêt, selon qu'il est central au sein d'une communauté ou

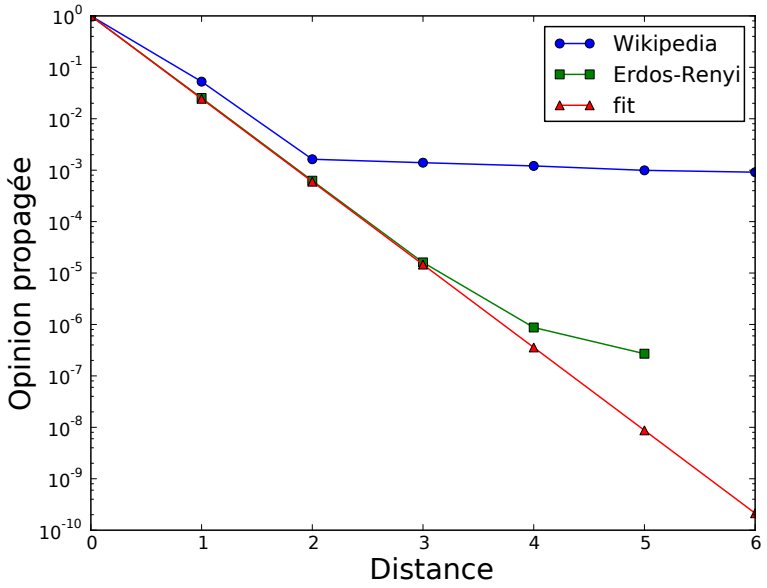
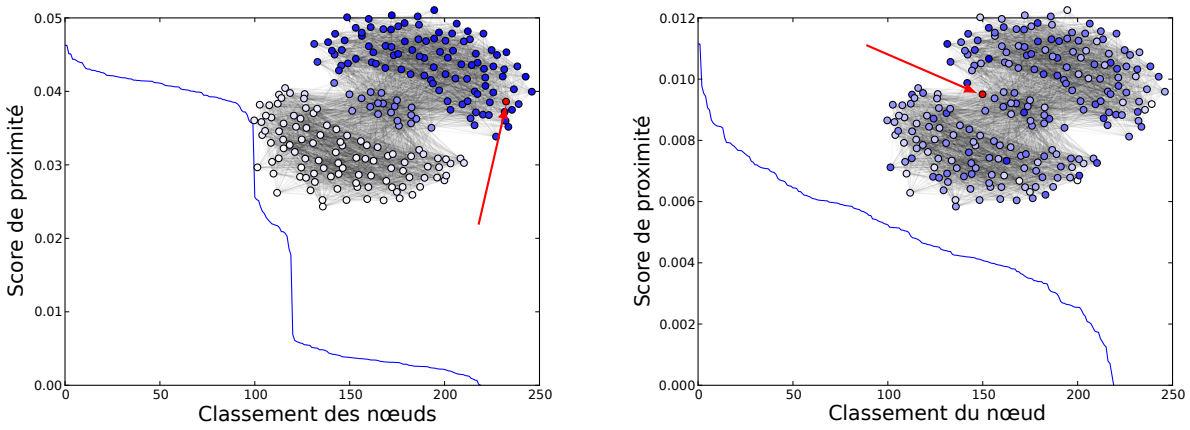


Figure 3.11 – Opinion propagée moyenne en fonction de sa distance au nœud d'intérêt, pour le réseau *Wikipédia 2008* et un graphe d'Erdos-Renyi contenant le même nombre de liens et de nœuds. La courbe $\frac{1}{k^d}$ où k est fixé au degré moyen des graphes précédents est également représentée, soit $k = \frac{2e}{n} = 40$ et la variable d représente la distance au nœud d'intérêt.



(a) (b)

Figure 3.12 – Opinion propagée avec le pas de correction supplémentaire pour deux graphes d'Erdos-Renyi de 110 nœuds avec une probabilité de connexion de 0.3 se recouvrant sur 20 nœuds. Plus le score est élevé, plus le nœud est foncé. 3.12a : le nœud d'intérêt est central à sa communauté et donne une structure nette en "plateau / décroissance / plateau". 3.12b : le nœud d'intérêt est périphérique et la structure en "plateau / décroissance / plateau" n'est plus clairement visible.

plutôt périphérique et appartenant à plusieurs communautés. La figure 3.12 illustre ce phénomène : quand le nœud est central à une communauté, une structure en "plateau / décroissance / plateau" émerge, alors que la décroissance est beaucoup plus régulière quand le nœud est périphérique ;

2. la structure communautaire du nœud d'intérêt, selon que sa (ou ses) communauté(s) est (sont) bien définie(s) ou pas. La figure 3.13 illustre ce phénomène : plus la communauté est clairement définie, plus la structure en "plateau / décroissance / plateau" est nette.

3.3.3 Comparaison plus avancée : confrontation à la vérité de terrain

Afin de valider plus avant cette approche "mesure de proximité", nous confrontons les scores de proximité à un nœud d'intérêt donné et la vérité du terrain. Nous étudions la proportion des nœuds qui se trouvent dans la communauté naturelle de ce nœud d'intérêt parmi les nœuds les mieux classés.

Pour cela, nous avons travaillé avec le réseau *Wikipédia 2012*, sélectionné des catégories agrégées qui correspondent à nos communautés vérité de terrain (comme détaillé dans l'annexe A) et essayé de trouver pour chaque communauté des nœuds importants.

La figure 3.14 montre certains de ces résultats. La figure 3.14a (resp. 3.14b, 3.14c)

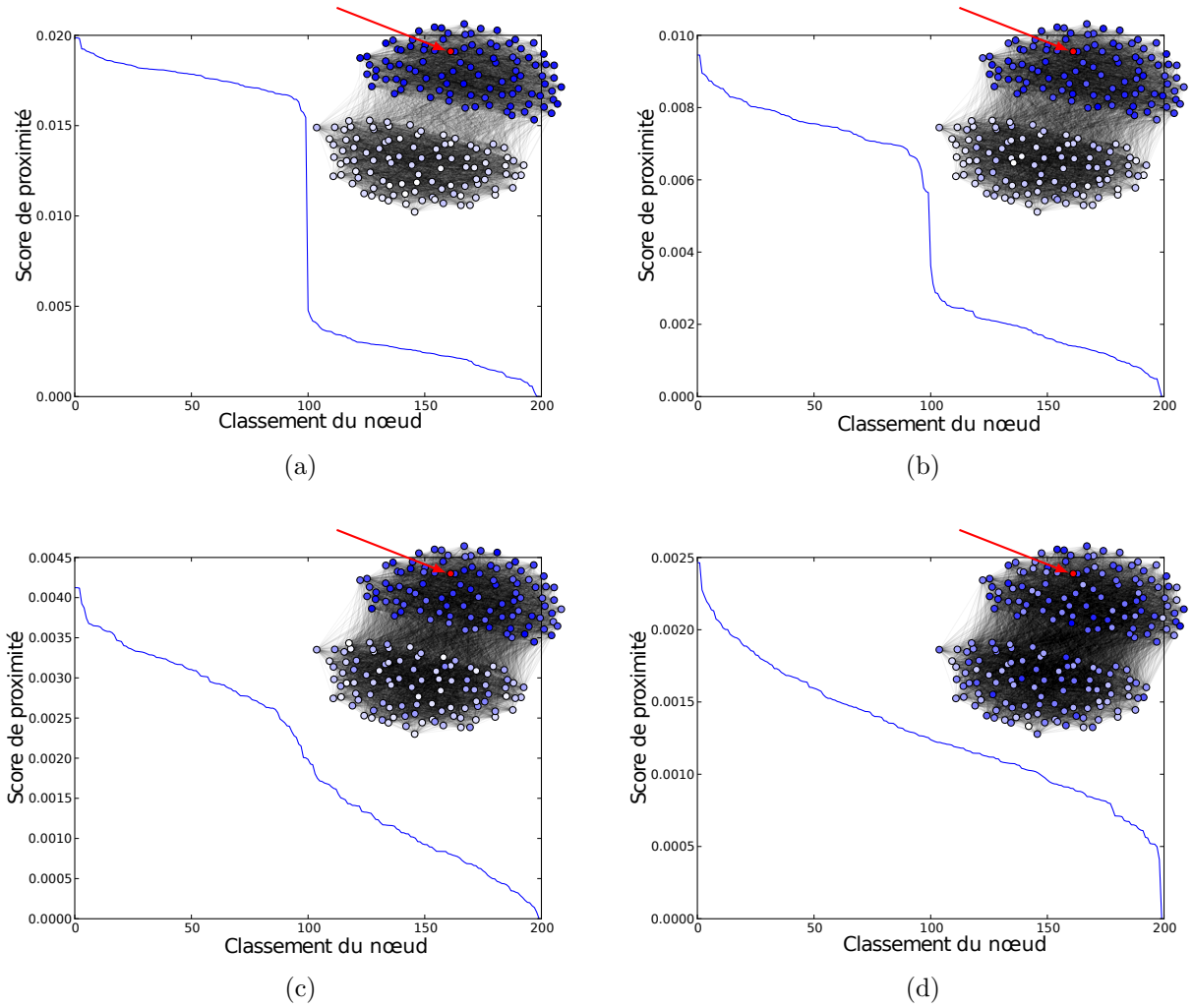


Figure 3.13 – Opinion propagée avec le pas supplémentaire de correction pour un graphe composé de deux sous-graphes d'Erdos-Renyi $(100, 0.5)$. 3.13a (resp. 3.13b, 3.13c, 3.13d) : deux nœuds dans des sous-graphes différents ont une probabilité d'être liés de 0.1 (resp. 0.2, 0.3, 0.4).

montre le résultat obtenu pour le nœud "Tayo Koki" (resp. "Cruiserweight", "Magnus Carlsen") et sa communauté naturelle "Sumo" (resp. "Boxing", resp. "Chess"). Comme nous pouvons le voir pour chacun de ces exemples, la plus grande pente semble coïncider avec la fin de la communauté (en suivant le classement en fonction de la proximité). La figure 3.14d montre le résultat obtenu avec le nœud "Chess Boxing" et les communautés "Chess" et "Boxing". Dans ce cas, la méthode ne marche pas car le nœud "Chess Boxing" n'est pas central à une communauté et appartient à plusieurs communautés.

Le tableau 3.2 montre les résultats associés aux figures 3.14 ainsi que d'autres résultats pour ce même réseau *Wikipédia 2012* : nous avons comparé l'ensemble des nœuds situés

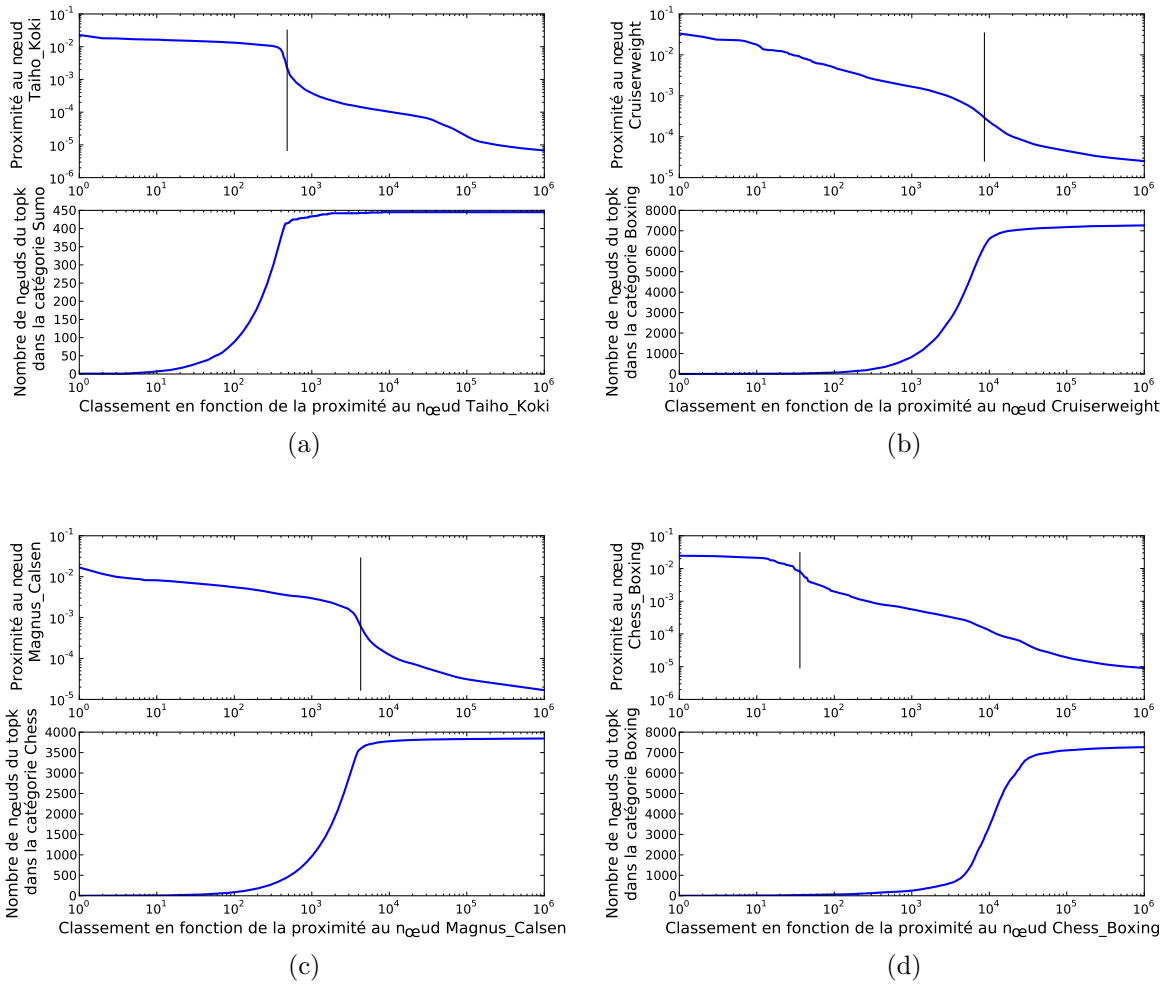


Figure 3.14 – Résultats obtenus dans le réseau *Wikipédia 2012*. La barre verticale identifie la position de la plus grande pente.

Catégorie	nœud d'intérêt	nombre de voisin	score F_1 voisins	pente maximale	nombre de noeuds	score F_1
Chess	Chess	4929	0.63	3.26	4027	0.92
3850 nœuds	Magnus Calsen	205	0.06	3.24	4102	0.90
	Queens Gambit	273	0.13	3.79	4560	0.88
	World Chess Championship	1040	0.27	3.35	4197	0.90
	Chess Boxing	113	0.01	1.37	35	0.00
Sumo	Sumo	1361	0.47	3.32	415	0.91
445 nœuds	Taiho Koki	97	0.25	5.10	461	0.92
	Yokozuna	9	0.01	2.61	478	0.89
	Lyoto Machida	238	0.01	1.58	82	0.01
Boxing	Boxing	9597	0.60	1.62	6809	0.83
7289 nœuds	Cruiserweight	304	0.06	1.58	8302	0.79
	Vitali Klitschko	308	0.02	0.94	9444	0.61
	Chess Boxing	113	0.01	1.37	35	0.01

Tableau 3.2 – Résultats obtenus sur le réseau *Wikipédia 2012*.

avant la plus grande pente avec l'ensemble des nœuds dans la catégorie du nœud d'intérêt en termes de score F_1 [5] et de taille de communauté. La méthode est efficace lorsqu'un représentant (c'est-à-dire, un nœud uniquement dans la communauté en question et important en son sein) est choisi, mais ne marche pas lorsque le nœud candidat n'est pas un représentant (dernier exemple à chaque fois). Nous avons également renseigné la taille et le score F_1 obtenus par le nœud d'intérêt et l'ensemble de ses voisins. Remarquons également que la valeur de la plus grande pente semble indiquer dans quelle mesure la communauté est bien dessinée et à quel point le nœud d'intérêt est central à la communauté.

3.4 Une nouvelle mesure de proximité paramétrée : Katz+

La mesure de proximité que nous venons de présenter n'a pas de paramètre et permet cependant d'obtenir des résultats intéressants dans de nombreux cas. Toutefois il est nécessaire, parfois, d'avoir des paramètres que l'on peut calibrer afin d'obtenir un résultat précis, par exemple dans le cas d'une complétion d'un groupe de nœuds en une communauté comme nous allons le voir dans le chapitre 5. Nous présentons ici une telle mesure de proximité paramétrée.

Avant d'aller plus loin, nous posons deux questions au lecteur pour bien lui faire comprendre l'intérêt d'incorporer des paramètres à une mesure de proximité :

1. Considérons une paire de nœuds 1 et 2 connectés mais isolés du reste du réseau et une

5. Pour deux communautés A et B, on a $F_1(A,B) = 2\frac{|A\cap B|}{|A|+|B|}$. Ce score est classiquement utilisé pour comparer des ensembles, la similarité de Jaccard l'est également. Mais lorsqu'il s'agit dévaluation on préfère généralement F_1 qui peut être vu comme la moyenne harmonique de la précision et du rappel : $F_1(A,B) = \frac{2precision(A,B)rappel(A,B)}{precision(A,B)+rappel(A,B)}$ avec $precision(A,B) = \frac{|A\cap B|}{|B|}$ et $rappel(A,B) = \frac{|A\cap B|}{|A|}$.

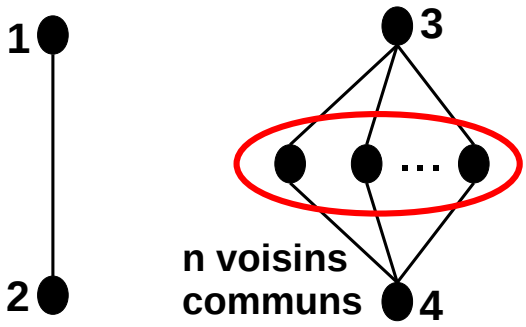


Figure 3.15 – Quelle paire de nœuds est la plus proche $\{1, 2\}$ ou $\{3, 4\}$?

paire de nœuds 3 et 4 non connectés mais partageant n voisins. Le module 3, 4 et les n voisins sont isolés du reste du réseau, comme illustré sur la figure 3.15. Qui sont les plus proches : $(1, 2)$ ou $(3, 4)$? Cette question dépend de n et d'une compréhension plus fine du réseau étudié. La fonction de proximité doit donc être contrôlée par au moins un paramètre afin de fixer ce compromis entre distance et redondance. Cette problématique est généralisable aux chemins de longueur supérieure à 2 : deux nœuds non connectés mais avec un voisin en commun sont-ils plus ou moins proches que deux nœuds non connectés, sans voisin commun, mais ayant n chemins de longueur 3 entre eux ?

2. Un autre problème vient de la présence de nœuds de fort degré (ou hubs). Soit un nœud 1 choisi au hasard dans le graphe : est-ce qu'un nœud 2 fortement lié à 1 et à son voisinage et faiblement lié au reste du graphe est plus proche de 1 qu'un nœud 3 fortement lié partout (y compris au nœud 1 et à son voisinage) ? Ici aussi le choix doit être laissé libre et nécessite donc un autre paramètre dans la mesure de proximité pour fixer ce compromis entre ce que l'on peut appeler popularité et intimité.

Nous venons ainsi de mettre en évidence l'intérêt d'au moins deux paramètres dans une mesure de proximité afin qu'elle soit précise et conforme aux préférences de l'utilisateur :

— un paramètre α pour contrôler le compromis entre distance et redondance (première question),

— un paramètre β pour contrôler le compromis entre popularité et intimité (deuxième question).

Afin de prendre en compte ces deux compromis et les problèmes liés aux différentes mesures de proximité détaillées dans l'état de l'art, nous proposons ici une nouvelle mesure de proximité basée sur la combinaison des chemins de différentes longueurs. Cette mesure repose sur l'indice de Katz, mais elle est plus générale que ce dernier et peut être calculée sur de grands graphes. Notre mesure de proximité peut également être calculée rapidement plusieurs fois avec des jeux de paramètres différents ce qui sera très utile par la suite, nous

y reviendrons dans le chapitre 5. La proximité du nœud j au nœud i est donnée par la formule suivante :

$$\mathrm{Prox}(i,j) = \sum_{l=0}^{\infty} \gamma_{l,d_j} N_l^{\mathrm{SP}}(i,j) \ , \tag{3.1}$$

où les γ_{l,d_j} sont des paramètres qui contrôlent la contribution des chemins de longueur l pour un nœud cible j de degré d_j. γ_{l,d_j} doit intuitivement décroître en fonction de l et de d_j pour pénaliser les nœuds plus éloignés ou de degré plus élevé.

Notons que cette mesure de proximité n'est pas symétrique, on a en général : $\mathrm{Prox}(i,j) \neq \mathrm{Prox}(j,i)$. Cependant cela n'est pas un problème pour nous puisque nous cherchons, à chaque fois, à calculer la proximité de tous les nœuds du graphe à un unique nœud donné et à comparer les valeurs obtenues entre elles. Cette mesure de proximité prend en compte, par l'intermédiaire des coefficients γ_{l,d_j}, les deux ingrédients qui nous ont semblé particulièrement pertinents, soit : (i) le compromis entre distance et redondance et (ii) le besoin de prendre en compte le degré des nœuds.

Nous allons maintenant simplifier cette mesure de proximité. En particulier, nous allons proposer un ansatz pour le coefficient γ_{l,d_j} (afin de revenir à seulement deux paramètres α et β) et utiliser des chemins sans cycle de longueur 2 (*non-backtracking paths* ou NBP) au lieu des chemins simples qui sont trop longs à calculer.

3.4.1 Complexité du calcul des chemins simples

Le nombre de chemins entre les nœuds i et j étant difficile à calculer (bien que des techniques d'approximation existent), nous proposons d'utiliser le nombre de chemins sans cycle de longueur 2 (ou NBP) ; les cycles de taille 3 ou plus sont autorisés. Le nombre de NBP entre un nœud i et chaque nœud du graphe est facile à calculer à l'aide du système d'équations suivant, en posant X_l le vecteur contenant le nombre de NBP de longueur l (i.e., la $j^{\text{ème}}$ coordonnée correspond au nombre de NBP entre i et j) :

$$\begin{aligned} X_0 &= \delta_i \\ X_1 &= AX_0 \\ X_2 &= AX_1 - DX_0 \\ \forall l \geq 2, X_{l+1} &= AX_l - (D-I)X_{l-1} \ , \end{aligned} \tag{3.2}$$

où δ_i est le vecteur nul sauf pour la coordonnée i qui vaut 1, A est la matrice d'adjacence, D est la matrice diagonale des degrés et I est la matrice identité. Le terme $(D-I)X_{l-1}$ élimine les chemins qui reviennent en arrière lors du $(l+1)^{\text{ème}}$ pas. La complexité pour calculer le nombre de NBP de longueur l entre un nœud d'intérêt et tous les autres nœuds du graphe est donc linéaire en fonction de la longueur des chemins, en $O(l(n+m))$, où n est le nombre de nœuds et m le nombre de liens.

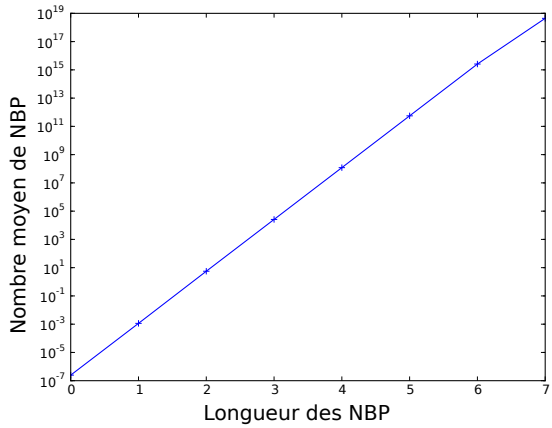


Figure 3.16 – Nombre de NBP entre un nœud choisi aléatoirement dans le graphe et chaque autre nœud en fonction de la longueur du NBP. Résultats obtenus sur *Wikipédia 2008*.

3.4.2 Longueur et nombre de NBP

La longueur et le nombre des différents NBP entre les nœuds i et j doivent être pris en compte et nous avons besoin d'un coefficient pour contrôler cela. Nous avons choisi de multiplier le nombre de NBP de longueur l par un coefficient α^l, où $\alpha \in]0,1[$. Le choix de ce handicap exponentiel est lié à la croissance exponentielle du nombre de NBP en fonction de leur longueur comme montré sur la figure 3.16.

Afin de diminuer davantage le temps de calcul, nous avons également ajouté une borne supérieure λ pour la longueur des NBP pris en compte. Nous verrons plus tard, dans le chapitre 5, que les longs NBP ne sont en général pas pertinents pour la détection de communautés. En effet, les NBP courts restent majoritairement dans la même communauté, alors que les longs en sortent souvent. Nous pouvons alors simplifier la mesure de proximité comme suit :

$$\mathrm{Prox}(i,j) = \sum_{l=0}^{\lambda} \gamma_{d_j} \alpha^l N_l^{\mathrm{NBP}}(i,j) \ . \tag{3.3}$$

3.4.3 Impact du degré des nœuds

Comme le montre la figure 3.17, le nombre de NBP augmente de manière linéaire en fonction du degré du nœud d'arrivée et cela est indépendant de la longueur des NBP considérés. Il y a alors deux choix naturels extrêmes pour γ_{d_j} :

1. fixer $\gamma_{d_j} = 1$, cette option consiste à ne pas du tout handicaper les nœuds de fort degré. Utiliser cette option pourrait cependant faire passer des hubs peu pertinents devant des nœuds de faible degré plus pertinents ;
2. fixer $\gamma_{d_j} = 1/d_j$, à l'inverse cette seconde option pénaliserait chaque nœud avec une

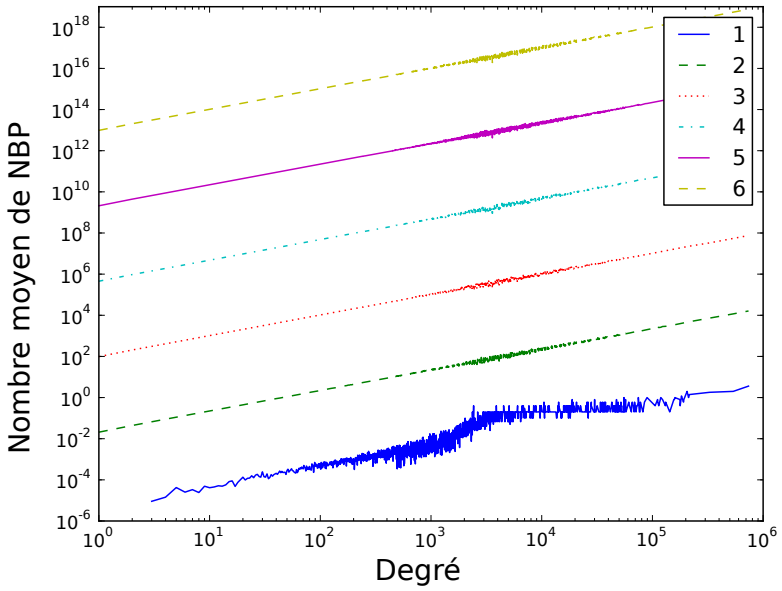


Figure 3.17 – Nombre moyen de NBP depuis un nœud choisi aléatoirement vers tous les autres nœuds du graphe en fonction du degré du nœud d'arrivée et de la longueur des NBP. Résultats obtenus sur *Wikipédia 2008*.

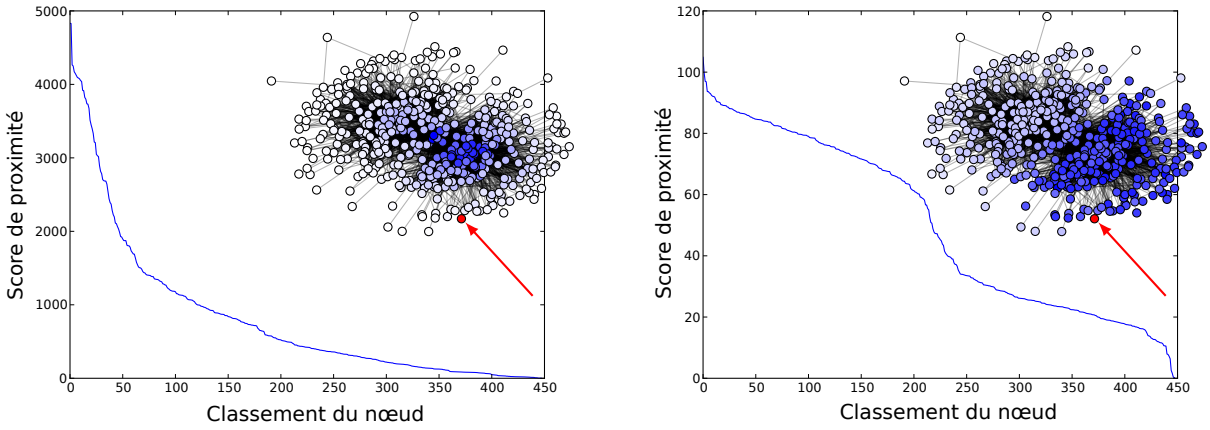
importance exactement égale à son degré. Les hubs perdraient alors leur centralité naturelle.

Nous proposons une solution intermédiaire qui consiste à handicaper les nœuds de fort degré avec une pénalité polynomiale en fonction de leur degré. Cela est contrôlé par le paramètre $\beta \in [0, 1]$ et les valeurs extrêmes 0 et 1 nous ramènent aux deux cas présentés précédemment alors que toute valeur intermédiaire handicape les nœuds de fort degré mais de manière sous-linéaire.

$$\mathrm{Prox}(i, j) = \frac{1}{d_j^{\beta}} \sum_{l=0}^{\lambda} \alpha^l N_l^{\mathrm{NBP}}(i, j) \tag{3.4}$$

La correction par le degré est importante. En effet, la figure 3.18 montre le résultat obtenu pour deux graphes générés avec le modèle configurationnel [116] sans et avec la correction sur le degré (i.e., le paramètre β). Nous voyons que, sans la correction sur le degré, la structure communautaire artificielle n'est pas capturée : seuls les nœuds de fort degré ont un score élevé relativement aux autres. La structure communautaire artificielle est en revanche retrouvée à la quasi-perfection lorsque l'on utilise la correction.

Enfin, nous avons remarqué empiriquement que ce handicap à décroissance polynomiale avantage trop les nœuds de faible degré dans les grands graphes de terrain tel que Wikipédia. En effet, si un hub connecte un nœud de très faible degré au nœud d'intérêt, ce chemin de longueur 2 lui donne trop d'importance. Nous avons donc décidé de contre-



(a) (b)

Figure 3.18 – 3.18a (resp. 3.18b) Score de proximité sans (resp. avec) la correction sur le degré. Plus un nœud est foncé, plus son score de proximité est élevé. Le graphe est composé de deux graphes de 250 nœuds se recouvrant sur 50 nœuds, chacun généré avec le modèle configurationnel en utilisant une distribution de degrés en loi de puissance avec un exposant -2. $\alpha = 0.5$ et $\lambda = 5$ pout les deux tests, $\beta = 0$ (resp. 0.9) pour la figure 3.18a (resp. 3.18b).

carrer ce problème en utilisant un dernier paramètre : une borne inférieure δ. Si un nœud est de degré inférieur à δ, nous le pénalisons avec un facteur $\frac{1}{\delta^\beta}$, au lieu d'un facteur $\frac{1}{d_j^\beta}$, c'est-à-dire que nous le pénalisons comme s'il était de degré δ.

3.4.4 Mesure de proximité finale

En conclusion, nous proposons la mesure de proximité suivante, que nous appelons Katz+ :

$$\mathrm{Prox}_{\alpha,\beta,\lambda,\delta}(i,j) = \sum_{l=0}^{\lambda} \gamma_{l,d_j} N_l^{\mathrm{NBP}}(i,j) \tag{3.5}$$

avec

$$\gamma_{l,d_j} = \begin{cases} \frac{\alpha^l}{d_j^\beta} & \text{if } d_j \geq \delta \\ \frac{\alpha^l}{\delta^\beta} & \text{if } d_j < \delta \end{cases} \tag{3.6}$$

En utilisant un exemple similaire à celui utilisé pour l'opinion propagée, sur la figure 3.12a, nous voyons en comparant les figures 3.19a et 3.19c qu'il est possible, avec le bon jeu de paramètres, d'obtenir un résultat très similaire à celui obtenu avec l'opinion propagée. Nous voyons également que contrairement à l'opinion propagée, Katz+ permet, avec le bon jeu de paramètres, de sélectionner (figure 3.19d) ou de ne pas sélectionner (figure 3.19b) les nœuds dans la zone recouvrante. Les paramètres permettent donc de mieux

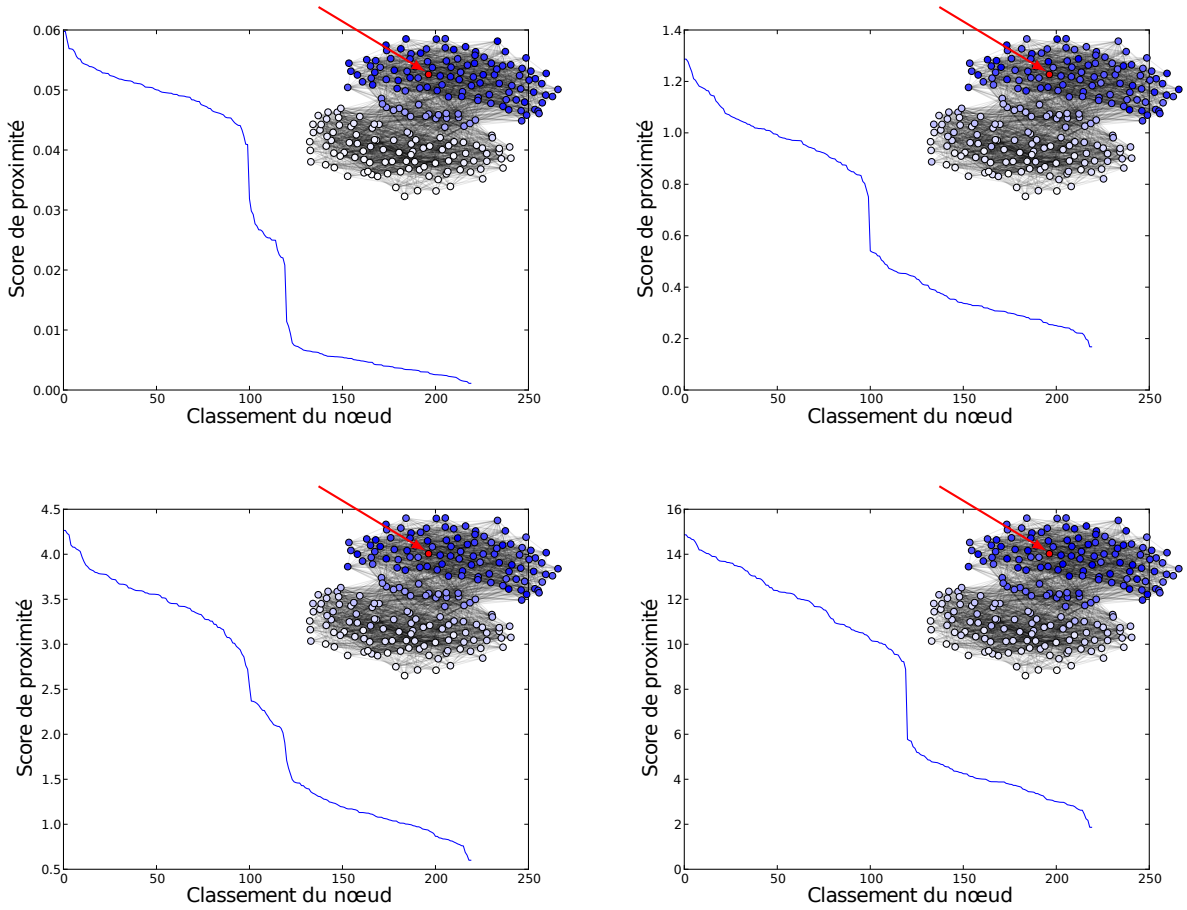


Figure 3.19 – Résultats obtenus sur un graphe composé de deux graphes d'Erdos-Renyi de 110 nœuds avec une probabilité de connexion de 0.3 se recouvrant sur 20 nœuds. La figure 3.19a est obtenue avec l'opinion propagée. Les figures 3.19b,3.19c et 3.19d sont obtenues avec les coefficients $\alpha = 0.9$, $\lambda = 5$, $\delta = 0$ et respectivement $\beta = 1.4$, $\beta = 1.$ et $\beta = 0.6$. On remarque une grande similarité entre les courbes 3.19a et 3.19c, ainsi que la possibilité avec Katz+ de sélectionner les nœuds périphériques à la communauté du nœud d'intérêt ou non.

contrôler la définition de communautés et donc le résultat qui sera obtenu, alors qu'en utilisant l'opinion propagée, qui est sans paramètre, on obtient un résultat intermédiaire non-ajustable où les nœuds de la zone recouvrante obtiennent un score moyen.

3.4.5 Discussion

Le scénario suivant illustre une limite de notre mesure Katz+. Elle n'est, en effet, pas entièrement satisfaisante. Considérons un nœud 1 relié à un nœud 2 par un nœud 3 de fort degré. Considérons aussi une configuration similaire, où un nœud 1_{bis} est relié à un nœud

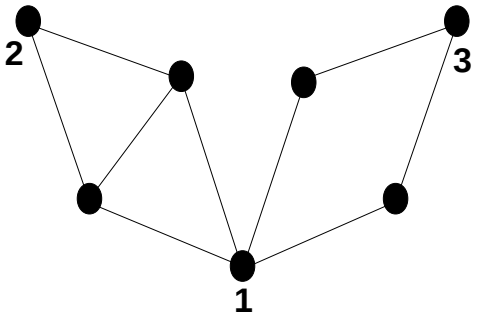


Figure 3.20 – Quelle paire de nœuds est la plus proche $\{1, 2\}$ ou $\{1, 3\}$?

2_{bis} par un nœud 3_{bis} de faible degré. Il s'avère que Katz+ donne la même proximité pour la paire $\{1, 2\}$ et la paire $\{1_{bis}, 2_{bis}\}$, alors que l'on aimerait que $\{1, 2\}$ soit moins proche que $\{1_{bis}, 2_{bis}\}$. Cela peut donner des résultats incohérents tels que : deux nœuds de faible degré liés entre eux par l'intermédiaire d'un hub (tel que la page USA dans Wikipédia) se retrouvent plus proches que deux nœuds de degré moyen liés par l'intermédiaire d'un nœud de faible degré. On peut évidement pallier ce problème en considérant le degré des nœuds présents sur les chemins, mais cela augmente la complexité du calcul de la mesure de proximité. Nous avons proposé de simplement handicaper davantage les nœuds de très faible degré (par l'intermédiaire du paramètre δ), mais cette solution ajoute un paramètre dont on aimerait bien se passer.

Une autre possibilité est d'utiliser le pagerank enraciné normalisé par le degré[6]. Le pagerank enraciné, comme l'indice de Katz, avantage de manière trop importante et incontrôlée les hubs. Bien qu'il permette de gérer le compromis "distance/redondance" (pour des nœuds de même degré, la proximité au nœud d'intérêt sera cohérente), il ne permet pas de gérer le compromis "popularité/intimité" : on observe qu'avec un α trop petit (par exemple de l'ordre de 10^{-9}), on obtient quelque chose de très similaire à la distance (on peut trouver une bijection entre les valeurs obtenues et la distance entre nœuds) et qu'avec un α un peu plus grand les hubs prennent trop d'importance. Pour ajouter un paramètre permettant de gérer ce deuxième compromis nous proposons de diviser les valeurs obtenues par le degré à la puissance β, comme pour Katz+ (les justifications de cet ansatz sont les mêmes que celles mises en avant pour Katz+).

Là encore des exemples spécifiques peuvent être construits et donner des résultats contre-intuitifs, comme sur la figure 3.20 tirée de [24] où le nœud 2 est évalué par le rooted page rank corrigé par le degré comme plus proche du nœud 1 quelles que soient les valeur de α et β.

Il semble donc nécessaire de prendre en compte le degré des nœuds sur les chemins pour réaliser une mesure de proximité paramétrée parfaite. Nous nous contenterons cependant

6. Nous avons vu lors de l'état de l'art la définition du pagerank enraciné : soit un marcheur aléatoire qui commence à marcher depuis un nœud d'intérêt en allant à chaque fois aléatoirement vers un voisin et en se téléportant au nœud de départ avec la probabilité α. La probabilité de trouver, au temps infini, le marcheur au niveau d'un nœud donné correspond à la proximité de ce nœud au nœud d'intérêt.

d'utiliser Katz+ lorsque nous aurons besoin d'une mesure de proximité paramétrée. Elle fournit, en effet, des résultats meilleurs que les mesures de l'état de l'art, notamment sur les cas spécifiques que nous allons étudier.

Avant de conclure ce chapitre, nous allons détailler *la méthode push* qui permet de calculer un pagerank enraciné approché de manière locale et donc beaucoup plus rapidement que par d'autres méthodes ; nous en aurons besoin au chapitre 6.

3.4.6 Calcul du pagerank enraciné approximé

Afin de pouvoir calculer une mesure de proximité très rapidement, les auteurs de [16] ne calculent pas le pagerank enraciné exact, mais un pagerank enraciné approché. Ceci permet de rendre le temps d'exécution de l'algorithme linéaire en fonction de la taille de l'ensemble des nœuds en sortie et indépendant de la taille du graphe.

Le calcul se fait à l'aide de *la méthode push* qui consiste à ne pas calculer le pagerank personnalisé avec la probabilité de redémarrer α et le vecteur e_i (vecteur nul sauf pour le nœud i où il vaut 1). Au lieu de calculer le vecteur $pr_\alpha(e_i)$ tel que :

$$pr_\alpha(e_i) = \alpha e_i + (1-\alpha)Tpr_\alpha(e_i),$$

il s'agit de calculer le pagerank d'un autre vecteur, celui du vecteur $e_i - r$ où initialement $r = e_i$, mais tel qu'au fur et à mesure des itérations, la norme infinie du vecteur $D^{-1}r$ décroît et devient inférieure à ϵ qui est le paramètre d'approximation donné en entrée (la multiplication par D^{-1} divise chaque entrée du vecteur par le degré du nœud associé).

Ainsi l'algorithme résout :

$$pr_\alpha(e_i - r) = \alpha(e_i - r) + (1-\alpha)Tpr_\alpha((e_i - r)).$$

et cette équation est un invariant de l'algorithme.

Initialement $r = e_i$ et $pr_\alpha(e_i - r)$ noté p pour simplifier est le vecteur nul. Puis à chaque itération, un nœud j avec son entrée $r[j] > \epsilon\ d[j]$ ($d[j]$ dénote le degré du nœud j) est aléatoirement sélectionné et :

- $\alpha\ r[j]$ est ajouté à $p[j]$,
- $\frac{r[j](1-alpha)}{d[j]}$ est ajouté à l'entrée du vecteur r de chacun des voisins de j, et
- $r[j]$ est mis à 0.

jusqu'à ce qu'il n'y ait plus de nœud j avec $r[j] > \epsilon\ d[j]$.

Ces opérations garantissent bien l'invariant de boucle et permettent donc bien de calculer le pagerank voulu de manière approchée. L'algorithme retourne non pas le pagerank enraciné sur le nœud d'intérêt i, mais un pagerank personnalisé par un vecteur $e_i - r$ tel que la norme infinie du vecteur $D^{-1}r$ est inférieure à ϵ.

3.5 Conclusion

Nous avons expliqué, dans ce chapitre, le lien qui existe entre mesure de proximité et structure communautaire. Nous avons ensuite présenté l'état de l'art des mesures de proximité les plus adaptées pour la détection de communautés. Nous avons également proposé

deux nouvelles mesures de proximité, l'une sans paramètre (l'opinion propagée) et l'autre paramétrée (Katz+). Nous allons dans les trois chapitres qui suivent exploiter ce lien entre mesure de proximité et communautés afin de construire des algorithmes qui permettent de trouver toutes les communautés auxquelles un nœud donné appartient (chapitre 4), de compléter un ensemble de nœuds en une communauté (chapitre 5) et de trouver des communautés recouvrantes dans un réseau et potentiellement toutes dans certains cas (chapitre 6).

Deuxième partie

Applications

Chapitre 4

Identification de toutes les communautés d'un nœud

Nous avons montré dans le chapitre 3 que lorsque l'on mesure la proximité d'un nœud d'un graphe à tous les autres nœuds, et que l'on classe ces scores par proximité décroissante, on obtient souvent une loi de puissance pour laquelle aucune échelle caractéristique ne peut être extraite. L'une des origines possibles de cette loi de puissance est que le nœud considéré appartient à des communautés de tailles différentes et qui se recouvrent. En effet, chacune des communautés forme une structure en "plateau / décroissance / plateau"et l'addition de ces structures donne une loi de puissance.

Afin de pouvoir utiliser la méthodologie basée sur les mesures de proximité pour détecter des communautés, il faudrait revenir à une forme de "plateau / décroissance / plateau". L'idée que nous proposons et étudions dans ce chapitre consiste à chercher des communautés bi-ego-centrées, c'est-à-dire des communautés partagées par deux nœuds. L'objectif est donc de trouver, à partir de deux nœuds, cette structure en "plateau / décroissance / plateau".

4.1 Communautés bi-ego-centrées : définition et détection

4.1.1 Principe

Afin de pouvoir calculer des communautés bi-ego-centrées, il est nécessaire de pouvoir évaluer la proximité d'un nœud à un ensemble de nœuds. Pour un ensemble de deux nœuds $\{u, v\}$, un nœud w est proche de la paire s'il est à la fois proche de u ET de v. Ainsi, connaissant la proximité $P(u, w)$ de u à w et $P(v, w)$ celle de v à w, une possibilité est de définir la proximité $P(\{u, v\}, w)$ de w à $\{u, v\}$ comme :

$$P_{min}(\{u, v\}, w) = \min(P(u, w), P(v, w)).$$

D'autre solutions sont également possibles, comme calculer le produit (ou la moyenne géométrique, ce qui est équivalent) :

$$P_{prod}(\{u, v\}, w) = P(u, w) \times P(v, w),$$

ou bien la moyenne harmonique :

$$P_{harm}(\{u, v\}, w) = \frac{2 \times P(u, w) \times P(v, w)}{P(u, w) + P(v, w)}.$$

Si l'on fait la somme (ou la moyenne arithmétique, ce qui est équivalent) :

$$P_{som}(\{u, v\}, w) = P(u, w) + P(v, w))$$

ou bien que l'on prend le maximum :

$$P_{max}(\{u, v\}, w) = \max(P(u, w), P(v, w)),$$

on évalue dans quelle mesure le nœud w est proche du nœud u OU du nœud v, ce qui n'est pas le but recherché.

La figure 4.1 illustre ces cinq approches sur un graphe jouet. On observe, sur les figures 4.1a et 4.1b, que les nœuds choisis appartiennent à deux communautés chacun (dont une en commun) et que l'évaluation de la proximité des nœuds du graphe jouet à chacun des nœuds sélectionnés fait ressortir les nœuds appartenant à ces deux communautés. Calculer le minimum 4.1c, le produit 4.1d ou la moyenne harmonique 4.1e de ces proximités fait ici bien ressortir la communauté que les deux nœuds ont en commun, contrairement au maximum 4.1f qui fait ressortir les nœuds appartenant à l'union des communautés et la somme 4.1g qui donne quelque chose d'intermédiaire.

On remarque par ailleurs qu'il y a peu de différence, sur cet exemple, entre faire le minimum, le produit ou la moyenne harmonique pour deux nœuds, bien que le minimum semble donner un résultat plus tranché. Pour plus de deux nœuds le produit ou la moyenne harmonique sont à privilégier puisque le minimum est moins stable : un seul nœud peut en effet complètement changer le résultat s'il a une faible proximité aux autres nœuds.

Nous proposons donc les idées suivantes :

— $P_{min}(\{u, v\}, w) = \min(P(u, w), P(v, w))$ mesure la proximité du nœud w à la paire $\{u, v\}$;

— la mesure de proximité étant censée donner un meilleur score pour deux nœuds dans la même communauté que pour deux nœuds dans des communautés différentes, la proximité d'un nœud à une paire $\{u, v\}$ doit donner une proximité plus grande si le nœud partage une communauté avec u et une communauté avec v (même si ce n'est pas la même) ;

— cela est généralisable à un ensemble de plus de deux nœuds. Dans ce cas P_{min} ou P_{harm} semblent plus stables pour mesurer la proximité à l'ensemble.

Ces idées sont en accord avec l'expérience faite sur le graphe jouet 4.1. Nous allons maintenant les valider plus avant sur le benchmark de Lancicineti et Fortunato [87] et sur le réseau *Wikipédia 2008*.

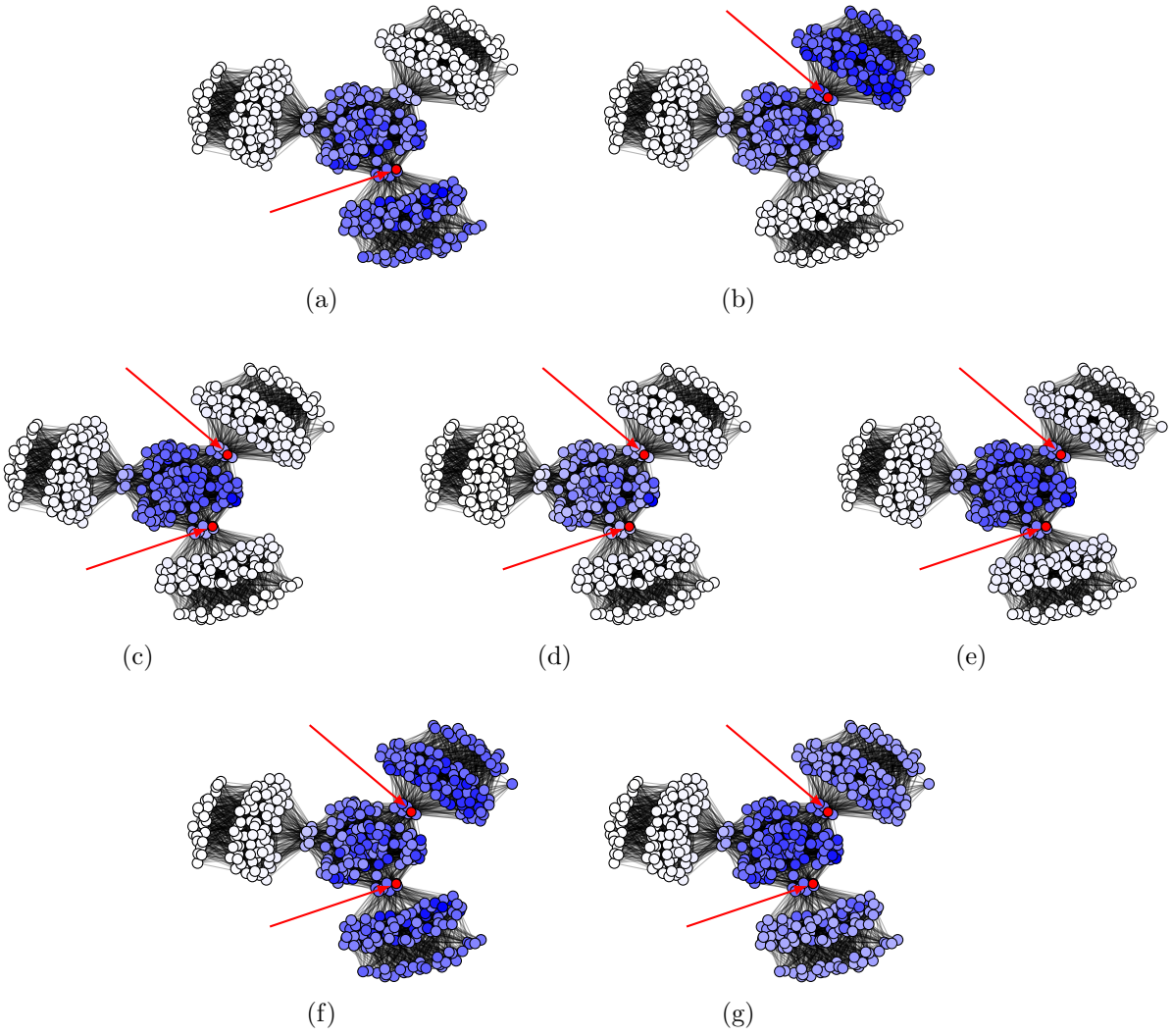


Figure 4.1 – Résultats pour un graphe jouet, illustrant une structure en communautés recouvrantes, composé de quatre graphes d'Erdos-Renyi de 100 nœuds avec une probabilité d'avoir un lien entre 2 nœuds de $p = 0.2$ et se recouvrant sur 10 nœuds. Des flèches pointent vers le(s) nœud(s) d'intérêt(s). Plus le nœud est foncé, plus son score est élevé : le score représente la proximité par rapport au nœud sélectionné pour les figures 4.1a et 4.1b. Pour la figure 4.1c (resp. 4.1d, 4.1e, 4.1f et 4.1g), le score représente le minimum (resp. le produit, la moyenne harmonique, le maximum et la somme) des scores obtenus pour chaque nœud sur les deux autres figures. La proximité utilisée est l'opinion propagée.

4.1.2 Validation sur le benchmark de Lancicineti et Fortunato

Nous avons généré à l'aide du benchmark de Lancicineti et Fortunato un réseau de 100 000 nœuds ayant 10 000 nœuds appartenant à 3 communautés, le degré moyen est fixé à 15, la taille maximum des communautés à 1000 et le paramètre de mixage à 0.2. Les autres paramètres sont laissés à leur valeur par défaut. Nous avons choisi deux nœuds partageant une même communauté et appartenant chacun à trois communautés. Nous avons choisi ces deux nœuds de manière à avoir un résultat représentatif.

Le résultat est présenté sur la figure 4.2 : comme on peut le voir pour le nœud 1 (resp. pour le nœud 2), les nœuds appartenant à l'union de ses trois communautés ont, en grande majorité, une proximité au nœud 1 (resp. au nœud 2) plus élevée que les autres : ils apparaissent en premier lorsqu'ils sont classés par ordre décroissant en fonction de leur proximité. De plus la décroissance des proximités ici forme une structure en "plateau / décroissance / plateau" qui coïncide avec la transition "dans les communautés / hors des communautés". En effet, en sélectionnant les nœuds classés avant la plus forte pente, nous obtenons un ensemble de 1928 nœuds (resp. 1833 nœuds), dont la similarité de Jaccard avec l'ensemble des nœuds partageant au moins une communauté avec le nœud 1 (resp. nœud 2), ensemble de 1993 nœuds (resp. 1854 nœuds), est de 0.92 (resp. 0.93).

Pour le minimum, le résultat est encore plus frappant : comme précédemment, les nœuds appartenant à la communauté commune aux deux nœuds d'intérêts sont classés en premier. Mais, ici, la structure en "plateau / décroissance / plateau" est encore plus frappante et la plus grande pente coïncide avec la transition "dans la communauté / hors de la communauté". Encore une fois, en coupant à la plus forte pente nous obtenons un ensemble de 836 nœuds, dont la similarité de Jaccard avec l'ensemble des nœuds dans la communauté commune, ensemble de 963 nœuds, est de 0.89.

4.1.3 Test sur Wikipedia

Nous avons appliqué la méthode proposée sur le réseau *Wikipédia 2008*. La figure 4.3a montre le résultat pour deux nœuds : "Folk wrestling" et "Torii school". La première page est dédiée aux différentes luttes traditionnelles pratiquées autour du monde et à différentes époques. La deuxième page est dédiée à une école d'art japonaise. Les deux courbes de "proximité VS classement" sont deux lois de puissance légèrement déformées et ne permettent pas d'extraire une communauté. Ceci peut s'expliquer par le fait que ces deux pages appartiennent à beaucoup de communautés de différente taille.
La figure 4.3b montre la courbe "proximité VS classement" obtenue pour le nœud "Sumo", ainsi que pour le minimum des proximités de "Folk wrestling" et de "Torii school". Le minimum remis à l'échelle a exactement la même structure que la courbe de "Sumo" : un plateau suivi par une forte décroissance autour du 350^{e} nœud puis un plateau.
De plus, en comparant les 350 premiers nœuds de chaque expérience on constate que :

— 337 nœuds sont à la fois dans les 350 premiers de "Sumo" et du minimum de "Folk wrestling" et "Torii school" ;
— 14 nœuds sont à la fois dans les 350 premiers de "Sumo" et "Torii school" ;

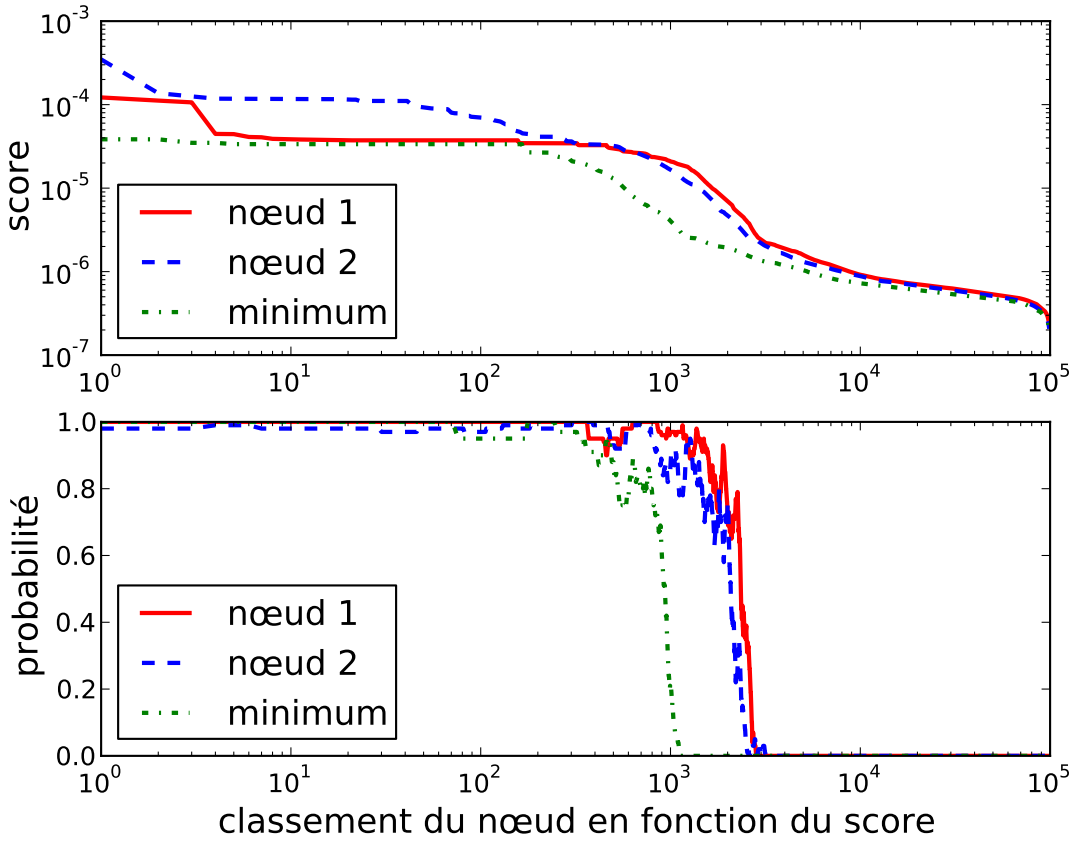
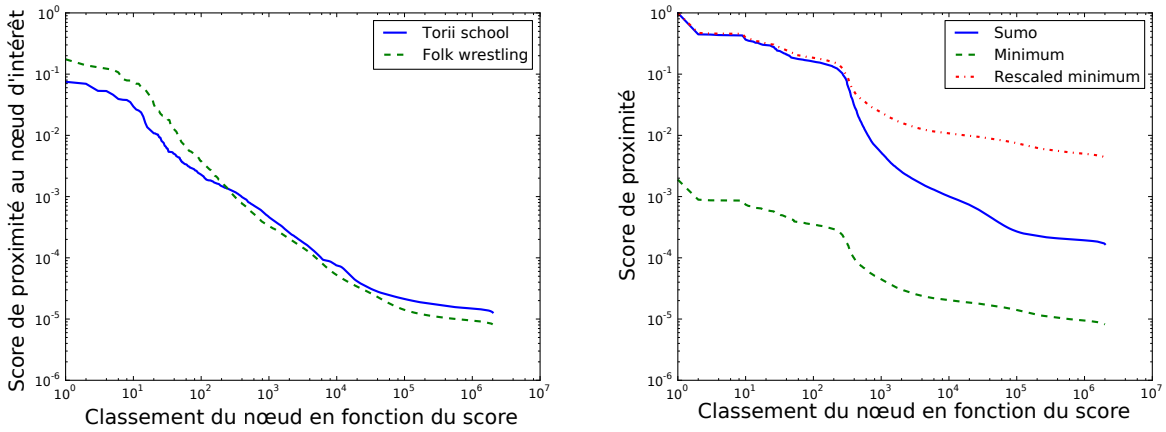


Figure 4.2 – Expérience sur le benchmark LF. Haut : proximité des nœuds en fonction de leur classement pour les deux nœuds ayant trois communautés dont une en commun, ainsi que le minimum de ces proximités en fonction du classement. Bas : la valeur à la position x correspond à la proportion de nœuds classés de x à $x + k$ qui sont dans l'une des trois communautés du nœud en question (dans la communauté en commun pour le minimum) en fonction du classement respectif. Nous avons pris $k = 100$ pour éviter de trop grandes oscillations.



(a) (b)

Figure 4.3 – La figure 4.3a montre les courbes “proximités VS classement” pour deux nœuds dans le réseau Wikipedia de 2009, “Folk wrestling” et “Torii school”. La figure 4.3b montre le résultat pour le nœud “Sumo” avec le minimum des proximités obtenues pour les deux nœuds “Folk wrestling” et “Torii school” en fonction du classement et la même courbe mise à l'échelle pour commencer à 1.

— 12 nœuds sont à la fois dans les 350 premiers de “Sumo” et “Folk wrestling”.

De plus, en évaluant manuellement les pages, on voit que les 350 premières pages de “Sumo” et du minimum sont très liées à la thématique sumo alors que les pages situées après la forte décroissance le sont beaucoup moins.

On peut donc ici conclure que :

— “Folk wrestling” et “Torii school” sont reliés à la communauté sumo de manière transversale : les nœuds qui sont à la fois proches de “Folk wrestling” et de “Torii school” sont des nœuds qui sont proches de “Sumo” ;

— le nœud “Sumo” et la paire {“Folk wrestling”,“Torii school”} définissent tous deux la même communauté, celle des pages de la thématique sumo.

Sur cet exemple nous avons donc identifié un ensemble de pages qui définissent une communauté également identifiable par un seul sommet : la communauté ego-centrée de “Sumo”. Cependant, nous pensons qu'il est également possible d'identifier des communautés multi-ego-centrées qui ne soient pas aussi des communautés ego-centrées. Nous allons maintenant proposer une méthodologie générale pour extraire toutes les communautés bi-ego-centrées.

4.2 Méthodologie

4.2.1 Idée générale et algorithme

Prenons l'exemple d'une personne dans un réseau social, nous l'appellerons Françoise. Françoise peut appartenir à plusieurs communautés : sa famille, plusieurs communautés d'amis ou de collègues. Ces communautés peuvent se recouvrir, c'est-à-dire que certains de ses amis peuvent également être ses collègues. Ainsi Françoise seule ne permet pas de définir une communauté unique. Par contre, dans l'entreprise où Françoise travaille, il y a sûrement un collègue, Michel, que Françoise ne connaît pas en dehors de son entreprise : l'ensemble des deux personnes {Françoise, Michel} peut alors définir l'ensemble des personnes travaillant dans l'entreprise où Françoise travaille. Cependant, si Michel pratique le tennis avec Françoise en dehors de l'entreprise, l'ensemble {Françoise, Michel} ne permet plus de définir une communauté unique de Françoise. Ainsi les nœuds doivent être proches, de manière à partager au moins une communauté, mais pas trop proche de manière à n'en partager qu'une seule. Pour parvenir à définir toutes les communautés de Françoise, il suffit donc de trouver ces personnes "ni trop proches, ni trop lointaines" de Françoise partageant une unique communauté avec elle, puis d'extraire l'intersection des communautés de Françoise avec chacune de ces personnes.

L'idée directrice est donc que : **"en général, un nœud appartient à beaucoup de communautés et ne peut donc pas caractériser une communauté à lui seul mais, deux nœuds bien choisis peuvent caractériser une communauté"**. Notre but est, suivant ce principe, de proposer un algorithme qui, étant donné un nœud u dans un réseau tente d'identifier et d'étiqueter toutes les communautés (possiblement recouvrantes) auxquelles ce nœud appartient, en utilisant d'autres nœuds comme artifices. De manière plus détaillée, l'algorithme consiste à :

1. **Choisir un ensemble de nœuds candidats** moyennement similaires à u. En effet si un nœud est trop peu similaire (resp. trop similaire) à u, il ne partagera aucune communauté (resp. beaucoup de communautés) avec u. Le but est que les deux nœuds n'en partagent qu'une.
2. **Chercher une communauté bi-ego-centrée** sur u et chaque nœud candidat v :
   — Pour chaque nœud du graphe, calculer le minimum de sa similarité à u et à v et trier les valeurs obtenues par ordre décroissant.
   — Si la pente maximale de la courbe de similarité en fonction du rang est supérieure à un certain seuil, alors une communauté est détectée, composée des nœuds situés avant cette pente.
   — Si u appartient à la communauté détectée, alors la communauté est conservée.
3. **Nettoyer et étiqueter les communautés trouvées** :
   — Plusieurs nœuds candidats peuvent donner naissance à des communautés très similaires. Afin d'éliminer ce bruit, on peut identifier ces communautés très similaires afin de n'en garder qu'une.
   — À l'inverse, si une communauté n'est similaire à aucune autre, elle est éliminée.

On a en effet observé que cela se produit quand la pente maximale détectée est faible et on peut donc assimiler cela à une erreur.

— La communauté reçoit l'étiquette du nœud le mieux classé.

Ces étapes sont détaillées ci-dessous. Pour le choix des candidats, dans l'idéal, pour un nœud u donné, il faudrait tester tous les autres nœuds du graphe afin de tester l'existence ou la non-existence d'une communauté bi-ego-centrée avec la notion de "plateau / décroissance / plateau". Cependant, si l'on choisit un nœud v très éloigné de u, il est peu probable que u et v partagent une communauté. Dans ce cas, chaque nœud du graphe sera loin de u ou bien loin de v et donc sa proximité à l'ensemble $\{u, v\}$ sera faible. À l'inverse, si l'on choisit un nœud v très proche de u, le calcul de la proximité des nœuds au nœud u ou au nœud v donnera un résultat similaire et donc la proximité à l'ensemble $\{u, v\}$ n'apportera pas d'information supplémentaire (on obtiendra la même courbe pour u, v et $\{u, v\}$). En général, le nombre de nœuds pertinents est petit par rapport au nombre total de nœuds dans le réseau : il est donc important de bien cibler ces nœuds pertinents. Le calcul de la proximité de u à tous les nœuds du graphe nous permet justement d'obtenir un score pour chaque nœud du graphe qui évalue la proximité entre ce nœud et u. Comme nous cherchons des nœuds qui ne soient "ni trop proches ni trop éloignés" du nœud d'intérêt u de manière à ce que chacun d'eux ne partage qu'une unique communauté avec u, il suffit donc de choisir deux seuils, un seuil inférieur et un seuil supérieur, et d'examiner tous les nœuds entre ces deux seuils. Si un nœud v est à la bonne distance de u on peut obtenir une structure en "plateau / décroissance / plateau" et donc identifier une communauté. On peut choisir ces deux seuils manuellement en choisissant quelques nœuds associés à différentes valeurs de proximité et en estimant approximativement entre quels seuils se situe la bonne distance. Après avoir fixé ces seuils, on peut chercher les communautés bi-ego-centrées sur le nœud d'intérêt et chacun des nœuds entre ces seuils par la détection de "plateau / décroissance / plateau" (première solution). Si le temps de calcul est un problème, nous ne sommes pas obligés de calculer les communautés pour chaque nœud entre les deux seuils. En effet, plusieurs nœuds entre les deux seuils vont certainement donner le même résultat. Il est donc possible d'accélérer le traitement, par exemple en choisissant un échantillon aléatoire de ces nœuds (deuxième solution). En allant plus loin, on peut aussi considérer que deux nœuds candidats vont donner les mêmes solutions s'ils sont trop proches l'un de l'autre. Ainsi, lorsque nous avons calculé le résultat pour un nœud v, nous pouvons éliminer les k nœuds les plus proches de v de la liste des candidats (troisième solution).

Pour chaque nœud candidat testé qui a donné lieu à une structure "plateau / décroissance / plateau", nous obtenons un groupe de nœuds situés avant la pente maximale. Cependant, plusieurs de ces groupes sont très similaires et il faut donc éliminer les duplications avant d'obtenir nos communautés finales. Nous proposons de nettoyer les groupes de la façon suivante : si la similarité de Jaccard entre deux groupes est trop élevée, alors nous supposons que les communautés sont les mêmes (à un peu de bruit près) et nous ne gardons que leur intersection. Nous proposons une étape additionnelle de nettoyage qui, bien qu'optionnelle, peut dans certains cas donner de meilleurs résultats. Il arrive qu'un groupe obtenu ne soit similaire à aucun autre et, dans ce cas, nous supprimons simple-

ment le groupe. En effet une communauté qui fait sens devrait être trouvée avec plusieurs nœuds candidats. Si un groupe n'est trouvé qu'une fois, il est fort probable que ce groupe ait été détecté alors qu'il n'aurait pas dû l'être. Cela arrive notamment si le seuil choisi pour décider si la pente maximale est suffisante est un peu trop bas. Enfin, un score est associé aux nœuds dans les groupes que nous avons extraits. Nous proposons d'utiliser l'étiquette du nœud le mieux classé comme étiquette de la communauté. Dans le cas où la communauté provient de l'intersection de plusieurs groupes, nous utilisons la somme des scores obtenus dans chacun de ces groupes.

La complexité et le temps de calcul des différentes étapes sont les suivants :

— $O(t)$ ($t = O(n \log(n))$ pour l'opinion propagée) pour calculer la proximité de tous les nœuds du graphe au nœud d'intérêt.
— $O(n_{candidat} t)$ où $n_{candidat}$ est le nombre de candidats choisis pour calculer la proximité de tous les nœuds du graphe à chaque candidat.
— $O(n_{candidat} n)$ pour faire le minimum pour chaque nœud entre la proximité au nœud d'intérêt et à chaque candidat.
— $O(n_{candidat} n \log(n))$ pour trier le minimum des proximités pour chaque candidat par ordre décroissant et détecter la plus grande pente.
— $O(n_{com}^2)$ pour calculer la matrice de Jaccard entre les communautés et regrouper les communautés trop similaires (i.e., les doublons).

Le temps total de l'algorithme est donc en $O(n \log(n))$ tant que le nombre de candidats est petit devant la taille du graphe. Pour wikipédia et l'exemple sur le nœud "Chess Boxing" détaillé dans la section suivante, il nous a fallu un peu moins d'un jour pour obtenir les résultats.

Nous allons maintenant détailler plus avant l'algorithme et les différents points clés, à savoir : le choix des nœuds candidats, le choix du seuil pour la pente à partir duquel nous considérons qu'une communauté existe et, enfin, le nettoyage et l'étiquetage à travers un exemple concret extrait de données réelles.

4.3 Résultats et validation

Les différentes étapes décrites précédemment permettent d'obtenir un ensemble de communautés étiquetées distinctes et auxquelles le nœud d'intérêt appartient. Nous allons maintenant présenter nos résultats sur différents réseaux, puis les comparer aux résultats de baselines et à d'autres méthodes.

4.3.1 Résultats sur *Wikipédia 2008*

Nous présentons ici les résultats obtenus en appliquant la méthode au réseau *Wikipédia 2008*, et plus précisément au nœud "Chess Boxing". Le chess boxing est un sport mêlant échecs et boxe dans des rounds alternatifs. Les résultats pour cette page sont facilement interprétables et une validation manuelle est possible.

Pour cette expérience, nous avons aléatoirement sélectionné 3000 nœuds candidats parmi les nœuds classés entre le 100^{e} et le 10 000^{e} en fonction de la proximité au nœud "Chess Boxing". Ces valeurs ont été choisies après une série de tests afin d'obtenir des résultats pertinents tout en conservant une vitesse d'exécution raisonnable. Les 3000 candidats ont mené à l'identification de 770 groupes de nœuds, c'est-à-dire que 770 candidats ont donné lieu à une décroissance en forme de "plateau / décroissance / plateau" avec une pente maximale suffisante, les autres n'ayant pas donné une telle structure. La figure 4.4 montre 3 exemples de candidats qui ont été fructueux et ont mené à l'identification d'un groupe, ainsi que l'exemple d'un candidat non fructueux.

L'étape suivante consiste à nettoyer les groupes obtenus. Les figures 4.5a et 4.5b montrent la matrice des similarités de Jaccard entre ces 770 groupes de nœuds. Les lignes (et les colonnes) de la matrice ont été réordonnées de manière à ce que les groupes similaires soient proches les uns des autres. On voit que 716 groupes (correspondant au gros carré blanc) sont très similaires entre eux et peu similaires aux autres groupes. On obtient de plus un groupe trouvé 18 fois, deux groupes trouvés 12 fois, un groupe trouvé 5 fois, et six groupes trouvés une seule fois.

L'intersection des 716 groupes très similaires donnera la communauté étiquetée "Queen's Gambit" (les étiquettes seront expliquées plus tard, cf. tableau 4.1) composée de 1619 nœuds. La majorité des nœuds candidats ont donc abouti à cette communauté, ce qui vient du fait qu'elle est la plus grosse et qu'on a donc plus de chance de l'extraire. De manière plus générale, si le nœud d'intérêt appartient à plusieurs communautés dont une très grande par rapport aux autres, la communauté très grande sera trouvée bien plus fréquemment que les autres. Cela peut poser des problèmes car il sera naturellement plus difficile d'extraire des petites communautés.

Les quatre groupes plus petits sont bien visibles sur la figure 4.5b où l'on a supprimé les lignes et les colonnes correspondant à la communauté "Queen's Gambit". Ils correspondent aux communautés étiquetées "Enki Bilal", "Uuno Turhapuro", "Da Mystery of Chessboxin' " et "Gloria" (cf tableau 4.1). On y voit aussi les 6 groupes qui ne sont similaires à aucun autre. Ces 6 groupes sont en fait des erreurs : une structure en "plateau / décroissance / plateau" a été détectée parce que le seuil de la pente a été fixé un peu trop bas. Ces groupes sont automatiquement effacés pendant la phase de nettoyage et ne posent donc pas de problème.

Cette décomposition en cinq groupes est facilement obtenue en faisant l'intersection des groupes trop similaires. Ici nous avons fait l'intersection des groupes ayant une similarité de Jaccard supérieure à 0.7 [1], les groupes qui ne sont similaires à aucun autre sont simplement éliminés. L'étiquette et la taille des 5 groupes sont présentés tableau 4.1 avec quelques exemples de nœuds présents dans ces groupes. On peut voir que l'algorithme trouve des communautés de taille très différente (de 26 nœuds à 1619 nœuds sur cet exemple).

Même si certaines communautés trouvées peuvent sembler étonnantes, on peut toutes les justifier directement à l'aide du contenu des pages Wikipédia correspondantes :

1. si A et B sont de même taille alors, $jac(A, B) > 0.7$ ssi A et B se recouvrent sur plus de 82.3%. Si $A \subset B$ alors, $jac(A, B) > 0.7$ ssi $|A| > 0.7|B|$

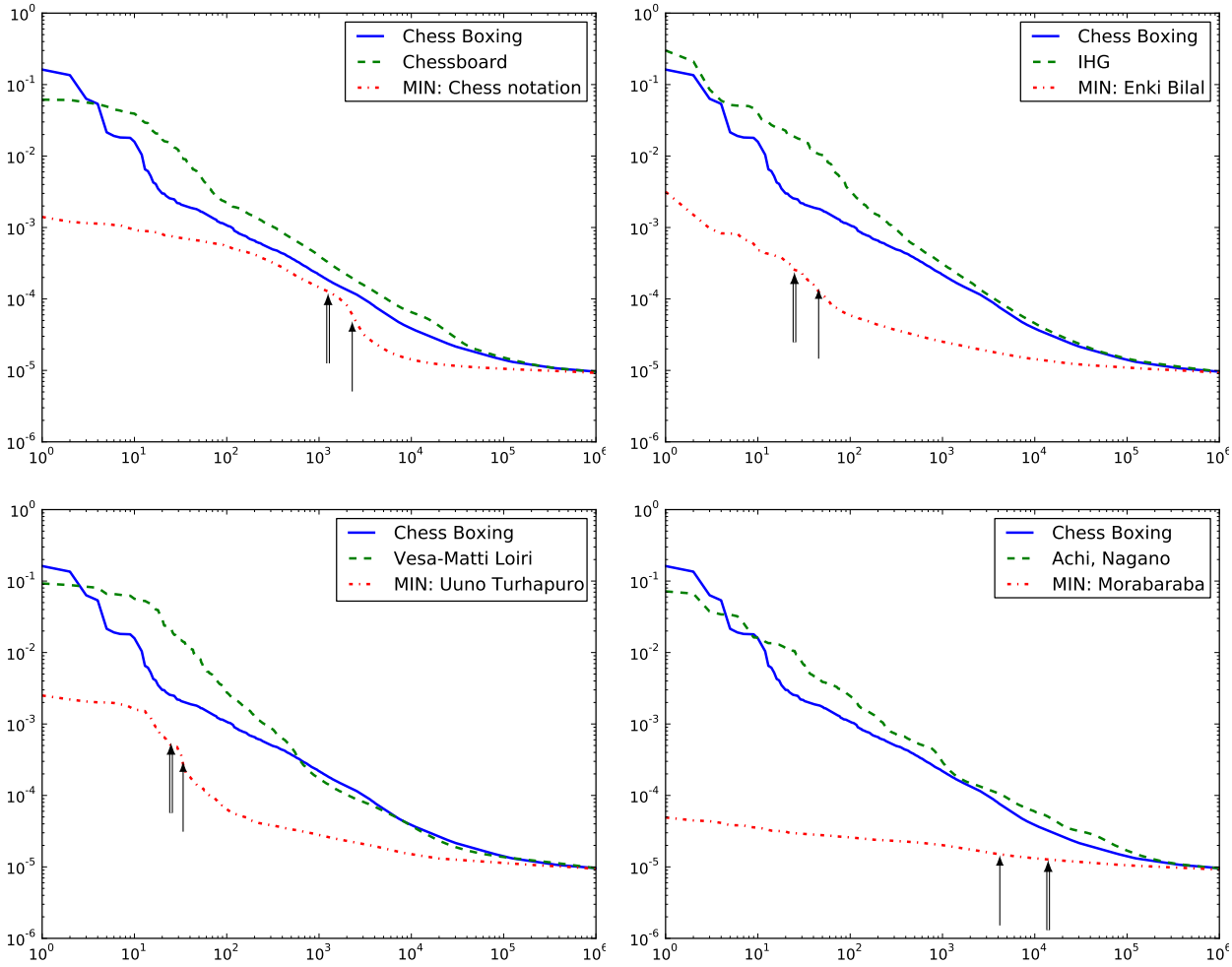
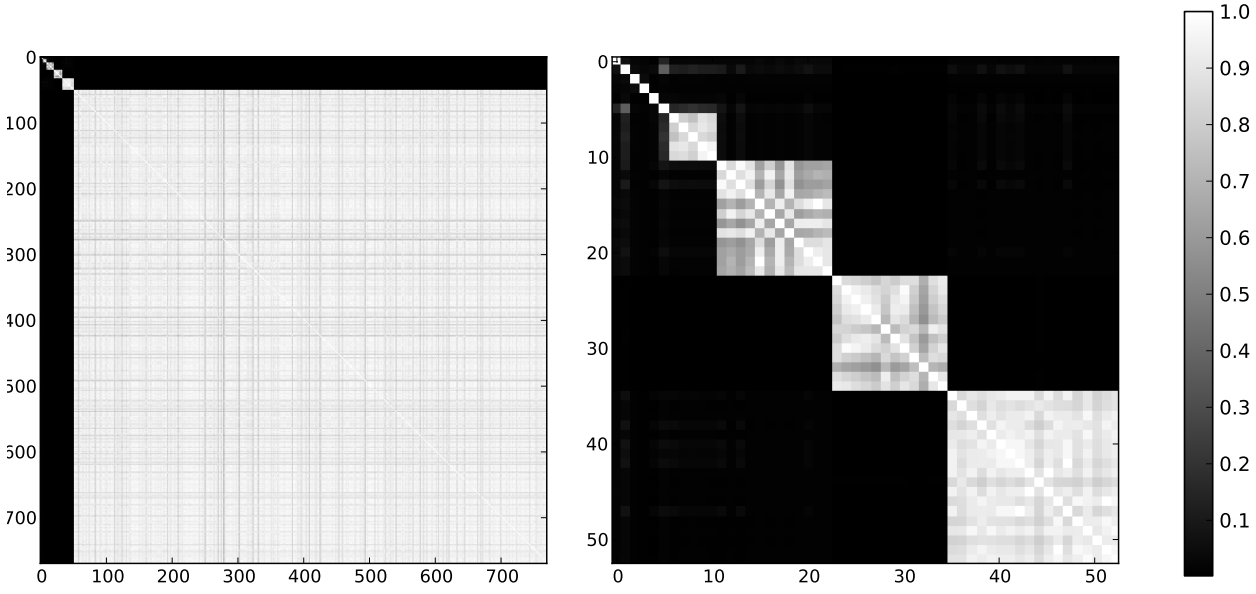


Figure 4.4 – Chaque figure montre les courbes correspondant à un essai : à chaque fois, la première (resp. la deuxième) courbe correspond aux valeurs de proximité au nœud Chess Boxing (resp. à un candidat pour le 2^e nœud, la légende montre l'identifiant de ce candidat). La troisième courbe correspond au minimum des proximités, la légende affiche le nom du nœud le mieux classé pour ce minimum. La double flèche montre la position du nœud "Chess Boxing", tandis que la flèche simple montre la position de la plus grande pente détectée.

— Enki Bilal est un dessinateur de bandes dessinées français, sa page Wikipédia explique qu'il a écrit la bande dessinée "Froid Équateur" qui a inspiré le créateur du chess boxing, Iepe Rubingh, pour la création de ce sport. Les nœuds de cette communauté sont principalement des pages dédiées aux autres bandes dessinées d'Enki Bilal.

— Uuno Turhapuro est un personnage de film finlandais. Sur la page Wikipedia dédiée au chess boxing on apprend que Uuno Turhapuro est, de même qu'Enki Bilal, re-



(a) (b)

Figure 4.5 – La figure 4.5a montre la matrice des similarités de Jaccard pour les 770 communautés (les colonnes et les lignes de la matrice sont réarrangées de manière à ce que des communautés détectées similaires soient à côté). Figure 4.5b montre un zoom sur la partie en haut à gauche de la matrice, i.e., après avoir enlevé les communautés correspondant au gros groupe de communautés similaires.

connu pour avoir inspiré la création du sport. Une scène dans le film montre le héros faisant une partie d'échec à l'aveugle avec un casque téléphonique tout en boxant avec une autre personne.

— Da Mystery of Chessboxin' est le titre d'une chanson écrite par le groupe de rap américain, The Wu-Tang Clan.

— "Gloria" est une page de désambiguïsation pour les pages contenant le mot gloria dans leur titre. Après inspection manuelle, la page actuelle de Wikipédia dédiée au chess boxing ne semble avoir aucun lien avec gloria. Cependant, elle contient la phrase "On April 21, 2006, 400 spectators paid to watch two chess boxing matches in the Gloria Theatre, Cologne". En vérifiant l'histoire de Wikipédia on voit qu'un lien de la page "Chess boxing" vers la page "Gloria" a été ajouté le 3 mai 2006 puis retiré le 31 janvier 2008. Comme le jeu de données que nous avons utilisé a été mesuré pendant cette période, "Chess Boxing" et "Gloria' étaient réellement connectés et il existait donc une communauté contenant ces deux pages.

— Queen's Gambit est une ouverture classique aux échecs et la communauté étiquetée Queen's Gambit est composée de pages fortement liées à la thématique des échecs. Bien que nous aurions aimé que cette communauté soit étiquetée Chess, Queens's Gambit est suffisamment spécifique aux échecs pour caractériser cette communauté.

Étiquette	Taille	Exemples de pages dans la communauté
Enki Bilal	35	Rendez-vous à Paris, Exterminateur 17, Iepe Rubingh, Le Sommeil du monstre, White Birds, The Black Order Brigade, Froid-Équateur, La Foire aux immortels, Fabrice Giger, Goran Vejvoda
Uuno Turhapuro	26	Uuno Turhapuro – kaksoisagentti, Uuno Turhapuro (film), Professori Uuno D.G. Turhapuro, Simo Salminen, Jori Olkkonen, Funny-Films Oy, Uuno Epsanjassa, Marjatta Raita, Chess boxing
Da Mystery of Chessboxin´	254	Wu-Tang Clan, Protect Ya Neck, Legend of the Wu-Tang Clan, Grandmasters (album), Gravel Pit, Wu-Tang Clan discography, Shame on a Nigga, Mr. Xcitement, U-God, Masta Killa, N-Tyce
Gloria	55	Gloria (Poulenc), Mass in E Flat Major, Gloria D. Miklowitz, Desiree Casado, Gloria (1999 film), Gloria (Disillusion album), Gloria, Oriental Mindoro, Gloria (singer), Chess boxing, Mass in F Minor
Queen's Gambit	1619	Checkmate, Fast chess, Baltic Defense, Symmetrical Defense, Closed Game, Marshall Defense, Tarrasch Defense, Torre Attack, Chigorin Defense, Chess handicap, Blindfold chess, Chess notation

Tableau 4.1 – Étiquette, taille et exemples de pages (les mieux classées à l'exception des pages avec un titre trop long) dans les communautés identifiées sur la page "Chess Boxing" de *Wikipédia 2008*. Le sens et la pertinence des exemples peuvent être vérifiés directement sur Wikipedia.

La page "Chess" y est classée en 3^{e} position.

L'algorithme n'a pas trouvé de communautés liées à la boxe. Cela pourrait être une erreur, cependant la page Wikipédia dédiée au chess boxing explique que la plupart des pratiquants de ce sport viennent du milieu des échecs et apprennent la boxe après. Il est donc possible que ce sport soit plus proche de la communauté des échecs que de celle de la boxe. Cela pourrait donc expliquer que "Chess boxing" entre bien dans la communauté des échecs, mais pas dans celle de la boxe.

4.4 Comparaison à d'autres méthodes

4.4.1 Comparaison avec des méthodes naïves

Nous avons comparé nos résultats aux deux méthodes de base suivantes afin de nous assurer que des méthodes naïves ne peuvent pas faire aussi bien.

Première méthode :

1. extraire le sous-graphe induit par les nœuds à une distance inférieure à 2 au nœud d'intérêt ;
2. chercher les communautés dans le sous-graphe induit à l'aide d'un algorithme de partitionnement de graphes classique, la méthode de Louvain [32].

Pour le nœud "Chess Boxing" dans le réseau *Wikipédia 2008*, le sous-graphe induit par les nœuds à distance 2 est composé de 0.5 million de nœuds et plus de 10 millions de liens. Louvain donne 15 communautés qui sont toutes énormes et ne semblent pas pertinentes après une analyse manuelle. Utiliser un seuil sur la distance au nœud d'intérêt ne semble pas être suffisamment limitant pour n'obtenir que des nœuds pertinents.

Deuxième méthode :

1. calculer la proximité de tous les nœuds du graphe au nœud d'intérêt ;
2. extraire le sous-graphe induit par les k nœuds les mieux classés en fonction de la proximité au nœud d'intérêt ;
3. chercher les communautés dans le sous-graphe induit, à nouveau à l'aide de la méthode de Louvain.

Pour le nœud "Chess Boxing" dans le réseau *Wikipédia 2008*, nous avons choisi les 5000 premiers nœuds. Ce nombre a été choisi pour avoir un nombre de nœuds suffisamment restreint, mais de manière à garder une population assez variée. Nous avons obtenu avec cette méthode 15 communautés. Après une analyse manuelle nous avons trouvé que certaines des communautés découvertes sont pertinentes et similaires à celles découvertes par notre méthode : un groupe de pages liées aux échecs, un groupe lié aux bandes dessinées et un autre à la musique rap. Cependant, nous avons également trouvé des communautés ne semblant pas du tout pertinentes.

4.4.2 Comparaison à une méthode de l'état de l'art

Nous avons ici comparé notre résultat à celui d'une méthode dédiée aux communautés locales et basée sur l'optimisation d'une fonction de qualité. Nous avons choisi la méthode de [117] que nous pensons être l'une des plus avancées puisqu'elle corrige des défauts de méthodes précédentes.

L'algorithme a donné deux communautés :

— La première contient 7 nœuds : "Comic book", "Enki Bilal", "Cartoonist", "La Foire aux immortels", "La Femme Piège", "Froid-équateur" et "Chess boxing". Elle semble pertinente et est très similaire à la communauté découverte par notre méthode et que nous avons étiquetée "Enki Bilal".

— La seconde contient 5 nœuds : “Germany”, “Netherlands”, “1991”, “International Arctic Science Committee” et “Chess boxing”. Cette communauté n'est similaire à aucune de celles que nous avons découvertes et ne semble pas pertinente.

4.5 Conclusion et perspectives

Nous avons proposé, dans ce chapitre, un algorithme qui permet d'identifier et d'étiqueter les communautés ego-centrées sur un nœud d'un graphe en calculant des communautés bi-ego-centrées sur ce nœud et d'autres bien choisis. Notre approche est basée sur la recherche d'irrégularités dans la décroissance des valeurs d'une mesure de similarité. L'algorithme est efficace en temps et permet de trouver les communautés ego-centrées dans des graphes contenant des millions de nœuds. En utilisant la notion de communauté multi-ego-centrée, l'algorithme identifie dans un premier temps des nœuds candidats pouvant permettre l'identification de communautés, puis cherche des communautés centrées sur notre nœud d'intérêt et sur ces candidats, et procède enfin à une phase de nettoyage et d'étiquetage des communautés. Nous avons validé les résultats sur un jeu de données réel et l'avons comparé à des méthodes de base et à une méthode performante de l'état de l'art.

Bien que cet algorithme soit performant en l'état, de nombreuses pistes d'amélioration sont possibles. Tout d'abord, la détection de communautés se base sur la recherche d'une structure de type “plateau / décroissance / plateau”. La méthode actuelle peut être améliorée, notamment par la recherche de plusieurs décroissances, ce qui permettrait de trouver des communautés à des échelles différentes pour un même candidat.

De plus, nous avons observé que l'algorithme peut avoir des difficultés à identifier de petites communautés si elles sont proches de grosses communautés. Pour cette raison, l'application de l'algorithme à des pages très populaires telles que "Biology" ou "Europe" ne conduit qu'à l'obtention d'une grosse communauté, alors que l'on s'attendrait à trouver divers sous-domaines de la biologie ou différents pays européens. Une piste d'amélioration pourrait consister à relancer récursivement l'algorithme sur des communautés identifiées pour trouver des sous-communautés.

Enfin, l'algorithme utilise pour l'instant la notion de communauté bi-ego-centrée, or il est possible que certaines communautés n'apparaissent qu'à partir de 3 sommets ou plus. Cette généralisation doit être validée sur des exemples de petite taille car le temps de calcul sera fortement augmenté, à moins que l'on améliore très significativement la méthode de sélection des candidats. Une approche pourrait être de considérer que si un candidat v a fourni de bons résultats, alors des nœuds qui lui sont très similaires n'apporteront pas de nouvelle information. Cette notion de rapidité de l'algorithme est centrale, notamment si l'on souhaite suivre l'évolution des communautés sur plusieurs instants de temps. Nous y reviendrons dans la conclusion générale.

Chapitre 5

Complétion de communauté

Ce chapitre est dédié à la complétion de communauté : étant donné un ensemble de nœuds dans un graphe, comment identifier tous les autres nœuds du graphe qui devraient figurer dans cette communauté ?

Contrairement aux approches présentées dans les chapitres 3 et 4 qui étaient centrées sur un nœud ou deux, le fait d'avoir à disposition un ensemble de nœuds permet d'étudier les propriétés de ceux-ci pour prédire les nœuds manquants. Cette connaissance supplémentaire permet d'utiliser des mesures de proximité paramétrées en tirant avantage de ces paramètres. En effet, s'il est évidemment possible de fixer arbitrairement ou de manière plus ou moins intuitive les paramètres de telles fonctions, lorsque l'on cherche à compléter une communauté, il est possible d'apprendre ces paramètres. Nous décrivons, dans la section suivante, notre méthode pour effectuer cet apprentissage.

5.1 Apprentissage des paramètres pour évaluer la proximité

Étant donnée une mesure de proximité entre nœuds P_Θ paramétrée par le jeu de paramètres Θ, on peut effectuer l'apprentissage de Θ de la façon suivante : pour un nœud u dans l'ensemble de nœuds donnés en entrée, on calcule la proximité de tous les autres nœuds à u, puis on sélectionne le jeu de paramètres tels que les autres nœuds de l'ensemble d'entrée soient le mieux classés possible. On recommence ensuite pour chaque nœud de l'ensemble en entrée.

Il y a plusieurs manières d'évaluer la pertinence d'un tel classement, c'est-à-dire d'évaluer le fait que les nœuds de l'ensemble d'entrée sont mieux classés que les autres. Les deux méthodes les plus utilisées sont :

— l'AUC (Area Under the roc Curve) qui, étant donné un classement, est une fonction strictement décroissante du nombre d'inversions nécessaires pour classer tous les exemples positifs en premier ;

— la précision à top-k qui, étant donné un classement, est égale à la proportion d'exemples positifs classés parmi les k premiers exemples.

Nous utiliserons principalement l'AUC puisque elle est sans paramètre, elle est très largement utilisée en apprentissage, et elle nous donne des résultats pertinents. L'inconvénient de cette mesure est que l'optimisation des paramètres doit être effectuée en testant de manière exhaustive l'ensemble des combinaisons de paramètres (par exemple avec une approche de type "grid search"). En effet, l'AUC d'un classement n'est pas une fonction dérivable et les méthodes d'optimisation efficaces, de type quasi-newton comme LBFGS [95] (qui nécessitent que la fonction à optimiser soit dérivable) ne sont pas utilisables.

La recherche exhaustive implique que nous devons utiliser une mesure de proximité qui, étant donné un jeu de paramètres, s'évalue le plus rapidement possible. Ainsi, bien que toute mesure de proximité paramétrée soit utilisable, nous utiliserons dans la suite la mesure de proximité Katz+ introduite dans le chapitre 3 :

$$\mathrm{Prox}_{\alpha,\beta,\lambda,\delta}(i,j) = \sum_{l=0}^{\lambda} \gamma_{l,d_j} N_l^{\mathrm{NBP}}(i,j) \tag{5.1}$$

avec

$$\gamma_{l,d_j} = \begin{cases} \frac{\alpha^l}{d_j^\beta} & \text{if } d_j \geq \delta \\ \frac{\alpha^l}{\delta^\beta} & \text{if } d_j < \delta \end{cases} \tag{5.2}$$

Avec cette mesure de proximité, la complexité de la phase d'apprentissage peut être évaluée exactement. Calculer le nombre de NBP pour toutes les longueurs inférieures ou égales à l prend un temps $\mathcal{O}(l(n+m))$, avec n le nombre de nœuds et m le nombre de liens dans le graphe, en utilisant le système d'équations 3.2. Ceci est à répéter une fois pour chaque exemple positif, soit n_{pos} (nombre d'exemples positifs) fois. Ensuite, étant donné un jeu de paramètres, il faut un temps $\mathcal{O}(ln)$ pour calculer la proximité de tous les nœuds au nœud d'intérêt. Ceci doit être effectué une fois pour chaque jeu de paramètres, soit n_{param} (nombre de jeux de paramètres) fois. Ainsi la complexité totale de la phase d'apprentissage est de :

$$\mathcal{O}(n_{pos} l(n+m) + n_{pos} n_{param} ln),$$

ce qui est essentiellement linéaire en fonction de la taille du graphe en entrée.

5.2 Résultats de l'apprentissage

Nous présentons ici des résultats obtenus sur le réseau *Wikipédia 2012*. Nous avons sélectionné les 542 pages qui sont dans la catégorie "Graph theory" ou dans l'une de ses sous-catégories directes. Nous avons divisé cet ensemble de 542 pages en un ensemble d'apprentissage que nous utiliserons pour apprendre les paramètres de la fonction de qualité et un ensemble de test pour la validation des résultats. Chaque ensemble contient 271 nœuds.

Nous rappelons ici le sens des différents paramètres utilisés dans la mesure de proximité :

— α : α^l contrôle la contribution des chemins de longueur l ;

— λ : est la longueur maximale des chemins utilisés ;

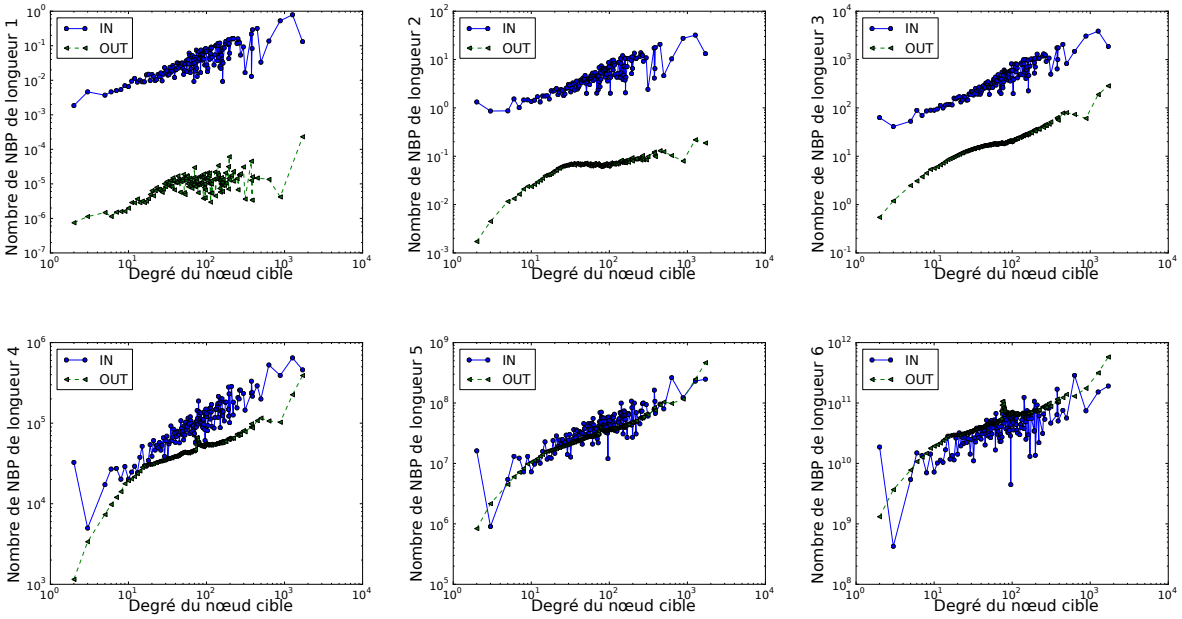


Figure 5.1 – Nombre moyen de NBP de longueur 1 à 6 (de gauche à droite et de haut en bas) en fonction du degré du nœud d'arrivée pour un nœud source et un nœud d'arrivée dans la catégorie "Graph theory" (IN) d'une part et pour un nœud source dans la catégorie "Graph theory" et un nœud d'arrivée hors de la catégorie "Graph theory" d'autre part (OUT).

— β : d^β pénalise le score d'un nœud de degré d ;
— δ : évite de trop avantager les nœuds de degré inférieur à δ.

Dans le but d'accroître la vitesse de l'optimisation de l'AUC par recherche exhaustive, nous avons conduit des tests préliminaires pour un éventail de jeux de paramètres avec $\alpha \in [0, 1]$, $\beta \in [0, 1.5]$, $\delta \in [\![0, 1000]\!]$ et $\lambda \in [\![2, 5]\!]$. Dans tous les cas la valeur optimale pour λ était 3 et pour δ elle était autour de 5. Par contre, les valeurs optimales pour α ont varié entre 0.002 et 1 et celles de β entre 0.6 et 0.9. Nous avons alors fixé $\lambda = 3$ et $\delta = 5$ et effectué une optimisation de α sur l'ensemble $\{0.001^{i/100} \mid i \in [\![0, 100]\!]\}$ et β sur l'ensemble $\{0.5 + 0.005i \mid i \in [\![0, 100]\!]\}$.

Le fait que les chemins plus longs que 3 ne soient pas particulièrement pertinents est appuyé par la figure 5.1. Il y a en effet beaucoup plus de chances qu'un chemin court qui part d'un nœud appartenant à une catégorie donnée aboutisse à un nœud de la même catégorie. Les chemins de longueur supérieure ou égale à 3 ont, au contraire, autant de chances de finir dans la catégorie qu'en dehors et ne portent ainsi pas beaucoup d'information.

Nous avons donc procédé à l'optimisation pour chaque nœud de l'ensemble d'apprentissage. La figure 5.2 montre les valeurs obtenues pour tous les couples (α, β) pour le nœud "Incidence list". On peut voir qu'ici le maximum est obtenu pour $\alpha = 0.0105$ et $\beta = 0.705$.

Nous avons d'abord vérifié grâce à l'ensemble de test que le modèle ne souffre pas

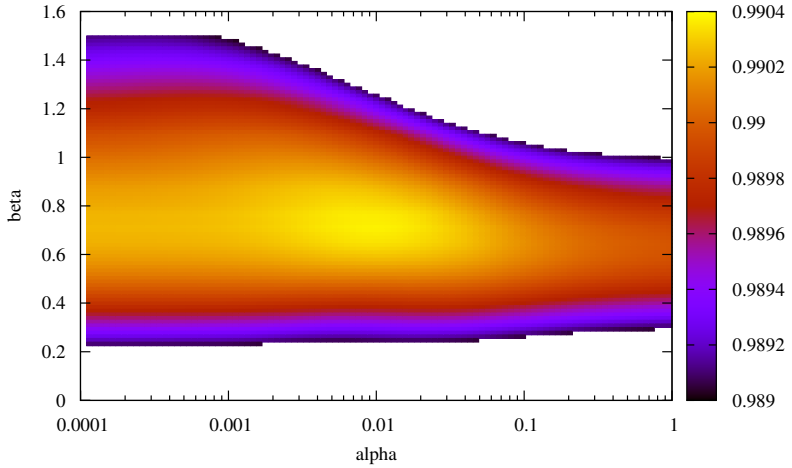


Figure 5.2 – AUC en fonction de α et β pour $\lambda = 3$ et $\delta = 5$ pour la page "Incidence list" de la catégorie "Graph theory". Un unique maximum est obtenu pour $\alpha = 0.0105$ et $\beta = 0.705$. L'AUC vaut alors $0,9904$.

de surapprentissage, c'est-à-dire que les performances sur l'ensemble d'apprentissage sont meilleures que celles sur l'ensemble de test. La figure 5.3 montre un nuage de points correspondant à l'AUC calculée sur l'ensemble d'apprentissage en fonction de l'AUC calculée sur l'ensemble de test pour chaque nœud de référence de l'ensemble d'apprentissage. On voit que les performances sont similaires et qu'il n'y a ici pas de surapprentissage.

La figure 5.4 montre le classement obtenu à partir de trois nœuds pour les couples optimaux (α, β) pour :

— un nœud donnant une mauvaise AUC, "Global shipping network" ;
— un nœud donnant une AUC moyenne, "Resistance distance" ;
— un nœud donnant une très bonne AUC, "Multiple edges".

L'AUC et la précision à top-1000 de quelques nœuds sont présentés dans le tableau 5.1. L'observation principale est qu'un nœud avec une bonne AUC permet de bien distinguer les nœuds de la communauté (complétion de l'ensemble d'entrée en une communauté) de ceux qui sont en dehors. Par exemple, la figure 5.4c montre une structure en "plateau / décroissance / plateau" avec une décroissance autour du 1000$^{\text{e}}$ nœud, ce qui indique qu'il y a approximativement 1000 nœuds proches de "Multiple edges" et que les autres sont plus éloignés. C'est donc un moyen d'identifier la communauté de la page "Multiple edges" qui contient la majorité des pages de la catégorie "Graph theory". Au contraire, la figure 5.4a pour le nœud "Global shipping network" ne montre pas de structure en "plateau / décroissance / plateau" et les nœuds de l'ensemble d'apprentissage et de l'ensemble de test sont mal classés. Ce nœud est, en effet, très périphérique à la communauté "Graph theory", voire en dehors. Le nœud "Resistance distance" donne un résultat intermédiaire :

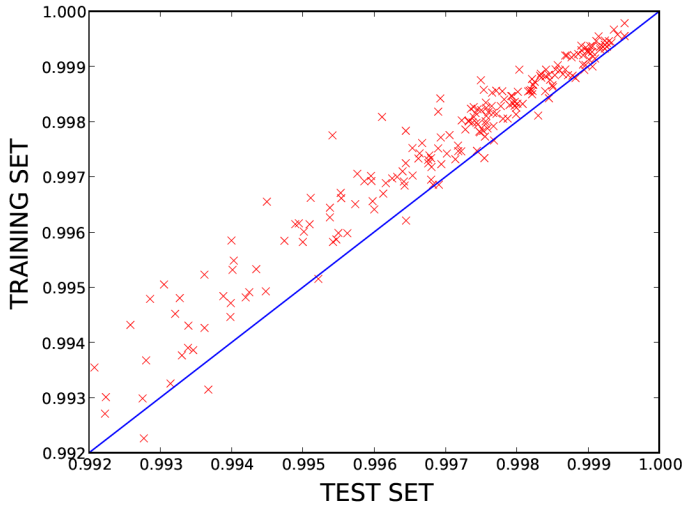


Figure 5.3 – Nuage de points de l'AUC de l'ensemble de test en fonction de l'AUC de l'ensemble d'apprentissage. Chaque point correspond à une page de l'ensemble d'apprentissage pour la catégorie "Graph theory".

les nœuds de l'ensemble d'apprentissage et de test sont moyennement bien classés et il y a une très légère structure en "plateau / décroissance / plateau". Ce nœud appartient aussi à la communauté des pages parlant d'électricité, il est, entre autres, lié à "Ohm" et "Resistance", en plus d'appartenir à la communauté des pages parlant de théorie des graphes. Ce nœud caractérise donc mal la communauté "Graph theory" à lui seul.

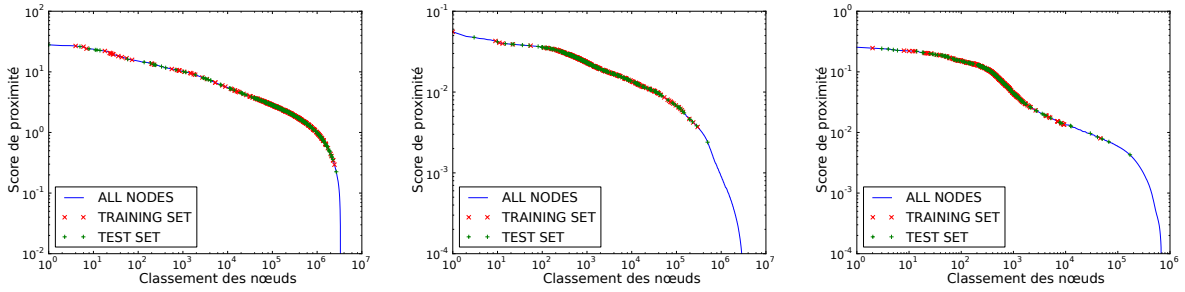


Figure 5.4 – De gauche à droite, scores de proximité en fonction du classement pour les couples (α, β) donnant la meilleure AUC en partant des nœuds "Global shipping network", "Resistance distance" et "Multiple edges".

L'AUC du classement obtenu par un nœud permet donc d'évaluer dans quelle mesure ce nœud est central pour l'ensemble donné en entrée ; ceci est appuyé davantage par le tableau 5.1 montrant des pages donnant :

— de très bonnes AUC : ces pages sont intuitivement très centrales à la communauté des pages parlant de théorie des graphes ;

— des AUC moyennes : ces pages parlent clairement de théorie des graphes, mais ont aussi leur place au sein d'autres communautés ;
— des AUC mauvaises : ces pages semblent placées dans la catégorie “Graph Theory” par erreur.

Cependant, si l'on considère deux nœuds donnant un classement moyen,“Fan Chung” et “Resistance distance” par exemple, et que, pour chaque nœud, on effectue le produit des deux scores de proximité on obtient un nouveau classement dont l'AUC est comparable à celle de “Multiple edges”. Cela signifie donc que l'ensemble {“Fan Chung”,“Resistance distance”} est suffisant pour caractériser la communauté “Graph theory”. Cela montre également que combiner les classements individuels permet d'obtenir de meilleurs classements.

5.3 Combinaison de scores

Comme expliqué dans la section précédente, classer les nœuds en fonction de leur proximité à un seul nœud est parfois insuffisant pour caractériser une communauté et, en général, il est important d'impliquer plusieurs nœuds. Même si dans notre exemple certains nœuds comme “Multiple edges” ou “Graph (mathematics)” sont suffisant pour caractériser à eux seuls leur communauté, un meilleur classement peut être obtenu par combinaisons.

Il y a beaucoup de manières de combiner les scores, comme expliqué au chapitre 4. Nous suggérons d'utiliser le produit de ces scores qui est plus stable que le minimum. La figure 5.5 montre l'AUC obtenue pour les 271 nœuds de l'ensemble d'apprentissage classés par ordre décroissant, ainsi que les AUC obtenues en effectuant le produit des scores obtenus par les k meilleurs classements, pour toutes les valeurs de k. Nous voyons ici que prendre plus d'un nœud de référence donne lieu à de meilleurs classements ; cependant, prendre un trop grand nombre de nœuds de référence diminue la qualité des classements.

L'objectif est ici de trouver l'ensemble de nœuds qui caractérise le mieux la communauté de l'ensemble fourni en entrée, ce qui équivaut à trouver la combinaison de nœuds qui donnera la meilleure AUC. Pour combiner les scores, nous proposons d'effectuer les deux opérations suivantes :

1. pour chaque nœud de l'ensemble en entrée, apprendre les paramètres optimaux de manière indépendante ;
2. combiner les scores obtenus pour chaque nœud du graphe de manière à obtenir le meilleur classement possible.

La proximité d'un nœud à l'ensemble de nœuds donné en entrée est ensuite définie à partir du score combiné.

Cependant, le nombre potentiel de combinaisons croît exponentiellement avec la taille de l'ensemble en entrée. Mais, comme nous avons vu que prendre trop de nœuds en compte ne donne pas nécessairement de bons classements, nous pensons qu'un petit ensemble de nœuds est suffisant pour caractériser une communauté. Nous proposons donc d'examiner seulement les paires, voire les triplets, de nœuds de référence. Pour notre exemple, le

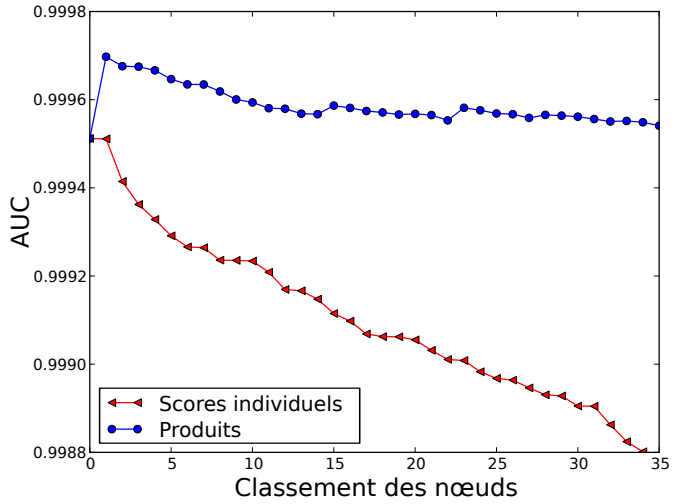


Figure 5.5 – AUC de toutes les pages considérées individuellement et AUC de la combinaison des k meilleures pages, pour tout k.

Classement	Page	AUC	P1000
1	Multiple edges	0.9997	0.804
2	Graph (mathematics)	0.9996	0.646
3	Random regular graph	0.9995	0.520
4	Random geometric graph	0.9995	0.572
5	Graph energy	0.9995	0.719
130	Graph rewriting	0.9974	0.1808
144	Resistance distance	0.9971	0.4833
167	Four color theorem	0.9962	0.4944
179	Control flow graph	0.9958	0.1513
198	Fan Chung	0.9939	0.3469
267	Pixel connectivity	0.9480	0.007
268	Condegram spiral plot	0.9465	0.0147
269	Agent Network Topology	0.9449	0.007
270	Ruth Aaronson Bari	0.9250	0.0258
271	Global shipping network	0.9073	0.0775
	moyenne	0.9933	0.4319

Tableau 5.1 – Nœuds donnant les meilleurs classements, des classements moyens et les moins bons classements avec l'AUC et la précision à top 1000 des classements associés.

meilleur classement est obtenu par la combinaison du premier et du troisième classement individuel, c'est-à-dire des scores obtenus par les nœuds 'Multiple edges" et "Random regular graph".

Classement	Page	Catégorie
3	Graphlets	Networks
6	Walls and Lines	G. theory
8	Complete graph	G. theory → Graphs → G. families → Regular G.
9	Chang G.	G. theory → Graphs → G. families → Regular G.
13	Local McLaughlin graph	G. theory → Graphs → G. families → Regular G.
14	Complete bipartite graph	G. theory → Graphs → G. families → Parametric families of G.
15	Quartic graph	G. theory → Graphs → G. families → Regular G.
23	Watkins snark	G. theory → Graphs → G. families → Regular G.
30	Brouwer-Haemers graph	G. theory → Graphs → G. families → Regular G.
32	Null graph	G. theory → Graphs → G. families → Regular G.
33	Bipartite graph	G. theory → Graphs → G. families
34	Planar graph	G. theory → Graphs → G. families
36	Biased graph	G. theory → Graphs → G. families
40	Szekeres snark	G. theory → Graphs → G. families → Regular G.
41	Process graph	G. theory → Graphs → Application-specific G.
42	Regular graph	G. theory → Graphs → G. families → Regular G.
43	Cubic graph	G. theory → Graphs → G. families → Regular G.
44	Petersen graph	G. theory → Graphs → G. families → Regular G.
45	Shannon multigraph	G. theory → Graphs → G. families → Parametric families of G.
49	Panconnectivity	G. theory → Graphs → G. families
54	Double-star snark	G. theory → Graphs → G. families → Regular G.
57	Meredith graph	G. theory → Graphs → G. families → Regular G.
58	Butterfly graph	G. theory → Graphs → Individual G.
64	Perfect graph	G. theory → Graphs → G. families → Perfect G.
65	Flower snark	G. theory → Graphs → G. families → Regular G.

Tableau 5.2 – Classement, titre et catégorie des 50 premières pages qui n'appartiennent pas à la catégorie "Graph theory" ni à l'une de ses sous-catégories directes. Si la page appartient à plusieurs catégories, nous avons choisi la plus pertinente.

La table 5.2 montre les cinquante premiers nœuds de ce meilleur classement qui ne sont ni dans l'ensemble d'apprentissage, ni dans l'ensemble de test, c'est-à-dire qui ne sont pas dans la catégorie "Graph theory". Nous pouvons voir que, bien que ces pages ne soient ni dans la catégorie "Graph theory" ni dans l'une de ses sous-catégories directes, elles sont très liées à la thématique de la théorie des graphes, et sont dans des sous-catégories de "Graph theory" plus lointaines, ou bien ont été ajoutées depuis l'extraction du dataset.

Une fois que les scores de proximité donnant la meilleure AUC sont obtenus, la question cruciale est de savoir où couper pour décider quels nœuds sont dans la communauté et quels nœuds n'y sont pas. Nous observons, comme classiquement dans cette thèse, que les nœuds situés bien avant la plus grande pente sont clairement dans la communauté et que les nœuds situés bien après ne le sont clairement pas ; les nœuds situés autour de la plus grande pente sont en général plus discutables : on voit qu'ils sont très proches du sujet, mais ne sont pas toujours pertinents. Nous proposons ainsi trois solutions :

1. couper à la plus faible valeur de la dérivée seconde, ce qui correspond à couper juste avant la plus grande pente et donc à exclure ces pages discutables et à ne garder

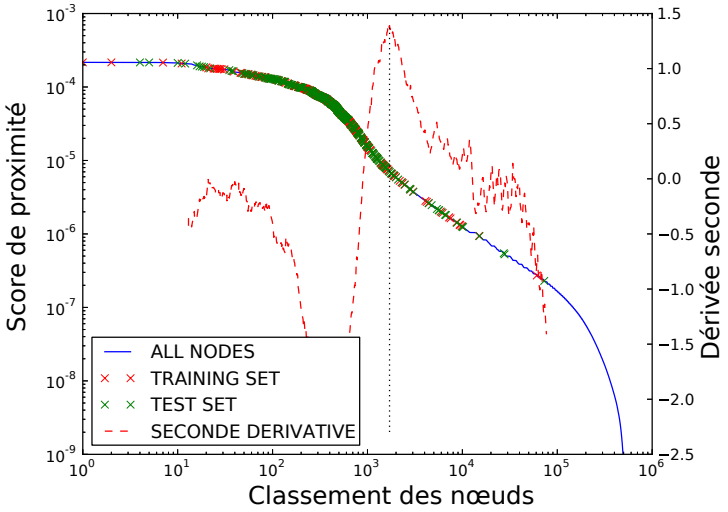


Figure 5.6 – Scores de proximité en fonction du meilleur classement combiné ainsi que la dérivée seconde de cette courbe.

que les pages centrales à la communauté ;

2. couper à la plus grande valeur de la dérivée seconde, ce qui correspond à couper juste après la plus grande pente et donc à garder ces pages situées à la limite de la communauté ;
3. couper à la plus grande pente, ce qui correspond à un résultat intermédiaire.

La figure 5.6 montre les scores obtenus pour le meilleur ensemble de nœuds de référence (i.e., celui donnant la meilleure AUC). Pour cet exemple nous avons coupé à la dérivée seconde la plus grande, car cette solution a fourni les meilleurs résultats, en observant la place des nœuds des ensembles d'apprentissage et de test. Cela a donné une communauté de 1708 nœuds qui contient 91% (resp. 92%) des nœuds de l'ensemble de test (resp. ensemble d'apprentissage).

5.4 Validation et comparaison

Les approches similaires à la nôtre ne sont pas spécifiquement dédiées à la complétion d'un ensemble de nœuds en une communauté. Par exemple, [84] et [149] utilisent une mesure de proximité pour trouver les k nœuds connectés les plus pertinents vis-à-vis d'un ensemble de nœuds, k étant un paramètre fixé par l'utilisateur. Ainsi nous avons utilisé quatre autres mesures de proximité qui n'utilisent pas d'apprentissage pour comparer et valider nos résultats :

1. la distance entre nœuds, notée "DISTANCE" dans la suite du document ;
2. la proximité de [149], notée "T and F" (suivant les initiales des auteurs) dans la suite, avec les paramètres recommandés par les auteurs ($c = 0.5$ pour la probabilité

de téléportation au nœud de départ et $\alpha = 0.5$ pour la normalisation en fonction du degré) ;

3. l'opinion propagée (cf. chapitre 3), "CAROP" (pour carryover opinion) ;
4. le Local Path Index de [101] en apprenant le paramètre β qui handicape les longs chemins "KATZ".

Nous avons :

1. calculé la proximité de tous les nœuds du graphe à chaque nœud de l'ensemble d'apprentissage,
2. combiné les scores de manière à obtenir la proximité à l'ensemble d'entrée,
3. coupé à la plus grande valeur de la dérivée seconde.

Nous avons ensuite comparé les performances de chaque méthode sur l'ensemble de test. Notre méthode a donné de meilleurs résultats à chacune des trois étapes : la figure 5.7 compare le nombre de nœuds de l'ensemble de test qui sont parmi les k premiers classés en fonction de k pour le meilleur classement individuel obtenu pour chaque méthode ; la courbe de notre méthode est bien au-dessus des autres pour tout k. Nous obtenons 80% des nœuds de l'ensemble de test dans le top 1000 avec notre méthode, alors que les autres en ont moins de 60% ; nous obtenons 95% des nœuds tests dans le top 2000 alors que les autres méthodes ont besoin du top 3000, voire plus, pour arriver au même pourcentage.

Notons que l'apprentissage de β pour le Local Path Index a donné de très petites valeurs, autour de 10^{-5}. Ceci est causé par le fait que des valeurs plus grandes pour β donnent une trop grande importance aux hubs, parfois peu pertinents, qui se classent en conséquence devant des nœuds de plus faible degré mais très pertinents. Cela explique pourquoi il est aussi important d'handicaper les hubs, comme nous l'avons expliqué auparavant. Il faut également noter que le fait de changer les paramètres pour "T and F" est très coûteux et que l'apprentissage des paramètres c et α avec cette mesure de proximité prendrait trop de temps.

5.5 Deux idées originales d'application de la méthode

5.5.1 Une vision communautaire des capitalistes sociaux sur Twitter

Les capitalistes sociaux sont des utilisateurs essayant de gagner en popularité, afin que leurs tweets aient plus de visibilité, en appliquant des méthodes de type FMIFY (Follow Me and I Follow You : l'utilisateur assure à ses followers qu'il les suivra en retour) et IFYFM (I Follow You, Follow Me : ces utilisateurs suivent d'autres utilisateurs en espérant que ceux-ci les suivent en retour). D'importants profils Twitter sont connus pour avoir utilisé cette méthode à leurs débuts, tels que ceux de Barack Obama, Britney Spears ou easyJet [56].

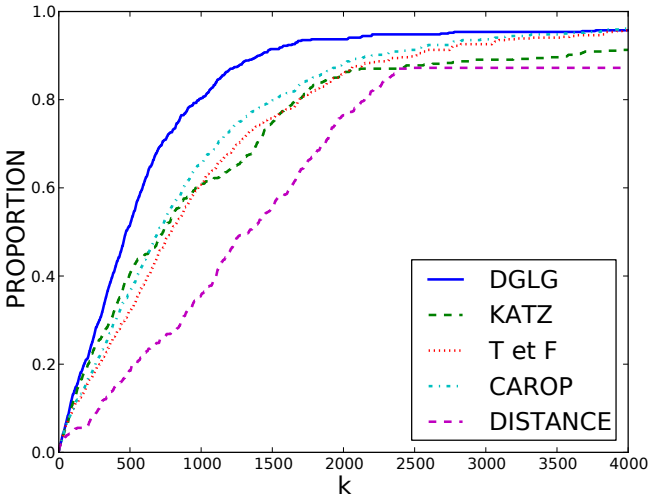


Figure 5.7 – Proportion des nœuds tests parmi les k nœuds les mieux classés en fonction de k pour le meilleur classement individuel obtenu pour notre méthode "DGLG" et les méthodes de l'état de l'art : "DISTANCE", "T and F", "CAROP" et "KATZ".

Ces utilisateurs peuvent aussi créer de faux comptes et les connecter entre eux et vers l'extérieur. Ils peuvent donc être apparentés à des Sybils [1]. Détecter ces utilisateurs malhonnêtes est crucial car ils peuvent polluer le réseau par des spams ; s'ils sont nombreux, ils peuvent également créer de l'astroturfing i.e., manipuler l'opinion des populations à des fins commerciales ou politiques en émettant la même information à travers différentes sources pour faire croire aux utilisateurs normaux que l'avis général de la population est le leur ou créer des hoax.

Étant donnés ces mécanismes de connexion (FMIFY, IFYFM et Sybil), un capitaliste social a tendance à être lié à beaucoup d'autres capitalistes sociaux et à peu d'utilisateurs normaux (comme montré dans [69], 68% des capitalistes sociaux ont plus de 50% de capitalistes sociaux parmi leurs followers). L'ensemble des capitalistes sociaux pourrait donc former à grande échelle une communauté bien définie. L'idée de communauté multi-égo-centrée semble donc très appropriée pour ce problème : étant donnés quelques capitalistes sociaux, on pourrait chercher leur communauté multi-égo-centrée et ainsi obtenir, dans l'idéal, l'ensemble des capitalistes sociaux présents sur Twitter.

Nous avons essayé d'appliquer la méthode de complétion de communauté décrite dans ce chapitre à ce problème, mais nous n'avons cependant pas rencontré de grands succès. Les seuls réseaux Twitter disponibles étant antérieurs à 2009, nous pensons que le capitalisme social n'était pas assez développé à cette époque et que les capitalistes sociaux constituaient plusieurs petites communautés déconnectées et donc difficiles à détecter avec notre méthode. Nous avons cependant mis au point une méthode de détection des capitalistes

1. Dans les réseaux sociaux un Sybil est un utilisateur créant plusieurs comptes dans le but de spammer ou de faire de l'astroturfing.

sociaux différente que nous détaillons dans le chapitre 7.

Cette idée que ces utilisateurs malhonnêtes qui essayent de gagner en visibilité se trouvent "au premier ordre" entre eux et isolés du reste du réseau a été, par ailleurs, utilisée pour établir des mesures de confiance dans les réseaux sociaux en ligne [86]. L'idée générale de ce travail est qu'un utilisateur malhonnête de ce type a des connexions avec d'autres utilisateurs malhonnêtes et ne se connecte que rarement à des utilisateurs honnêtes à travers des connexions dites "connexions d'attaque". Ainsi, une marche aléatoire partant d'un tel utilisateur serait plus longue à converger vers sa distribution de probabilité finale que celle partant d'un utilisateur "normal". Une mesure de confiance d'un utilisateur pourrait donc être déduite de ce temps de convergence.

5.5.2 Communauté des requêtes pédophiles effectuées sur un réseau pair-à-pair

Les réseaux pair-à-pair (P2P) constituent un moyen populaire d'échanger de grands volumes de données sur Internet. La pédophilie est un enjeu de société important qui a un impact sur la régularisation de l'Internet. Des travaux de recherche récents [91] ont développé un filtre sémantique dans le but de détecter les requêtes pédophiles effectuées sur un tel réseau reposant sur la combinaison de mots-clefs. La proportion des échanges de fichiers pédopornographiques a ainsi pu être estimée.

Nous pensons qu'il est possible d'améliorer ce filtre sémantique en complétant la communauté des requêtes déjà étiquetées comme pédophiles par le filtre. On considère pour cela le graphe utilisateurs-requêtes associé. Ce graphe étant ici biparti, une adaptation de la méthode est nécessaire ; nous exposons des travaux préliminaires allant dans ce sens dans le chapitre 9. Notre approche nous permet d'identifier des requêtes non détectées comme pédophiles, et donc de nouvelles combinaisons de mots-clefs. Nous présentons également quelques mesures pour vérifier la pertinence des catégories initialement définies dans le filtre.

5.6 Conclusion et perspectives

Nous avons présenté dans ce chapitre une méthodologie qui, étant donné un ensemble de nœuds, calcule un score de proximité entre tous les nœuds du graphe et cet ensemble. Notre méthodologie utilise une proximité paramétrée, apprend ces paramètres à partir des nœuds donnés en entrée et combine les classements individuels obtenus pour chacun des nœuds de cet ensemble. De plus, l'étude des classements individuels pour un nœud donné permet de savoir s'il est plutôt central ou périphérique par rapport à l'ensemble.

Si la majorité des nœuds de l'ensemble de départ appartiennent à une communauté, une structure en "plateau / décroissance / plateau" de la courbe des scores de proximité (en fonction du classement) est obtenue et une coupe à la dérivée seconde la plus grande permet une détection précise de la communauté, dite multi-ego-centrée. Nous avons validé

la méthodologie avec des tests sur des graphes jouets et sur le benchmark de [87], ainsi qu'en complétant des catégories (annotées par les utilisateurs) dans Wikipédia.

Une possible extension de ces travaux est l'étude de communautés multi-ego-centrées pondérées, possiblement avec des poids négatifs.

Chapitre 6

Détection de communautés recouvrantes

6.1 Introduction

Parmi les problèmes liés à la détection de communautés, le problème le plus général et le plus difficile reste la détection de tous les groupes de nœuds pertinents dans un grand réseau réel, c'est-à-dire la détection de toutes les communautés recouvrantes. En effet, bien que des solutions à ce problème aient été proposées (par exemple la percolation de k-cliques [120, 85, 127], BigClam [164], la partition de liens [10], etc.), aucune solution ne fait l'unanimité et n'est entièrement satisfaisante. La conception d'un critère pour décider si un groupe de nœuds est pertinent et la détection rapide de ces -potentiellement très nombreux- groupes sont les principaux goulots d'étranglement scientifiques. Nous proposons ici une méthode de détection de communautés recouvrantes faisant face à ces deux problèmes.

Nous montrons dans la section 6.2 qu'en général et contrairement à ce que l'on pourrait penser, le nombre de groupes pertinents est du même ordre de grandeur, voire plus petit que le nombre de nœuds dans le réseau. Ainsi, si l'on peut détecter chacun de ces groupes en un temps constant (ne dépendant pas de la taille du réseau), alors la conception d'un algorithme rapide (linéaire) pouvant traiter les réseaux réels les plus grands est possible.

Dans la section 6.3 nous montrons qu'en général, dans les réseaux réels, une communauté a un représentant, c'est-à-dire un nœud qui appartient uniquement à cette communauté et qui est important en son sein. Nous montrons également, dans cette même section, comment on peut détecter rapidement la communauté d'un représentant avec une "approche mesure de proximité". Cette "approche mesure de proximité" requiert effectivement un temps ne dépendant pas de la taille du réseau.

Un algorithme basé sur la détection de la communauté ego-centrée des représentants et linéaire en fonction de la taille du réseau est alors concevable. Ces communautés détectées sont alors étiquetables avec l'étiquette de leur représentant. Nous présentons cet algorithme dans la section 6.4. Nous y présentons également une procédure de nettoyage permettant d'éliminer les communautés trouvées plusieurs fois, dont la complexité est également

linéaire dans les réseaux réels. La méthode de nettoyage consiste, lorsque deux communautés sont jugées trop similaires, à enlever la communauté évaluée comme la moins pertinente. La condition pour avoir un temps linéaire pour cette procédure de nettoyage est que le nombre de communautés partageant un nœud donné ne dépende pas de la taille du réseau (nous faisons le calcul seulement pour les paires de communautés partageant au moins un nœud). Nous montrons que cette condition est effectivement satisfaite dans les réseaux réels.

Dans la section 6.5, nous comparons la performance de notre méthode à celles de l'état de l'art sur de grands réseaux réels. Nous discutons les différents critères de comparaison entre des communautés recouvrantes détectées et des communautés recouvrantes "vérité de terrain" et choisissons ceux qui nous paraissent les plus pertinents. Selon ces critères, notre méthode obtient des résultats comparables voire meilleurs que les méthodes de l'état de l'art sur ces grands réseaux réels. Nous montrons également que notre méthode est capable de traiter les plus grands réseaux contrairement aux autres méthodes.

Finalement, nous concluons et présentons les perspectives d'amélioration de notre méthode dans la section 6.6. Une voie d'amélioration certaine consiste en l'examen de communautés bi-ego-centrées et pas seulement ego-centrées afin de détecter des communautés sans représentant unique. Nous avons présenté, dans le chapitre 4, des travaux concluants allant dans ce sens dans le cadre de la détection de toutes les communautés d'un nœud.

6.2 Discussion sur le nombre de communautés dans un réseau

Le nombre maximum de groupes dans un réseau de n nœuds est de $2^n - 1$. Il est donc *a priori* possible que le nombre de communautés recouvrantes dans un réseau soit également de cet ordre de grandeur. Si tel était le cas, tout algorithme cherchant à détecter toutes les communautés recouvrantes d'un graphe serait voué à l'échec : (i) le temps de calcul serait trop important, (ii) le stockage des résultats serait impossible et (iii) la structure découverte serait complètement inutile. Remarquons que, même si ce nombre de communautés était en $\Omega(n^2)$, les trois propriétés précédentes seraient également vraies pour de très grands graphes. Avant d'aller plus loin, nous devons donc essayer de nous convaincre que cela n'est pas le cas !

Une communauté étant identifiée par des régions denses (en termes de liens) avec des régions moins denses autour, il semble difficile d'atteindre ce nombre de $2^n - 1$. En effet, parvenir à dessiner des régions denses et des régions moins denses entre tous ces groupes semble difficile.

Une meilleure borne supérieure serait peut-être de considérer le nombre maximum de cliques maximales dans un réseau de n nœuds : les cliques représentant les régions denses à l'extrême et l'absence de liens, les régions moins denses. Ce nombre est de l'ordre de $3^{\frac{n}{3}}$ [110] qui est plus petit que $2^n - 1$, mais reste encore bien trop élevé.

Ajoutons que les réseaux réels sont creux et ont une dégénérescence [1] notée d peu élevée. Une autre borne supérieure du nombre maximum de cliques maximales en fonction de la dégénérescence est $(n-d)3^{\frac{d}{3}}$ [58] : nous approchons ici du nombre de cliques maximales, linéaire en fonction du nombre de nœuds. Remarquons cependant que ce modèle n'autorise pas des communautés hiérarchiques et cette valeur peut donc difficilement être qualifiée de borne supérieure.

Pour aller plus loin, nous proposons au lecteur de se situer comme un nœud dans son propre réseau social de connaissances et de réfléchir autour de quelques formules. Notons C le nombre de communautés dans un réseau, on a clairement $C < n.c$ où c est le nombre moyen de communautés auxquelles un nœud appartient car chaque communauté contient plusieurs nœuds. Nous demandons à présent au lecteur d'évaluer à combien de communautés il appartient (amis, famille, collègues, etc.). Cette question peut sembler floue et difficile sans définition formelle de ce qu'est une communauté, mais le but est seulement d'avoir un ordre de grandeur de c et de se convaincre que ce nombre est petit devant le nombre de nœuds dans le réseau (c'est-à-dire, ici le nombre de personnes sur la Terre, de l'ordre de 7 milliards en 2015). 10, 30 ou même 100 nous semblent toutes être des réponses légitimes. Maintenant, en se considérant comme un nœud moyen du réseau, on obtient donc que C est de l'ordre de n.

On peut aller encore plus loin à l'aide de la formule exacte suivante [2] :

$$C = n\frac{c}{t}.$$

où t dénote la taille moyenne d'une communauté. En effet $\frac{c}{t}$ est le nombre moyen de communautés associées à un nœud et cette formule est évidente puisque, par définition, on a $c = \frac{Ct}{n}$. Nous invitons maintenant le lecteur à réfléchir encore une fois pour évaluer la taille moyenne de ses communautés. C'est encore une question difficile, car une communauté n'est pas une clique et donc les communautés d'une personne ne contiennent pas que des contacts directs (elles contiennent a priori aussi des gens que la personne ne connaît pas), mais encore une fois seul un ordre de grandeur importe. Faire le ratio entre son nombre moyen de communautés et la taille moyenne de ses communautés donnant une approximation de $\frac{c}{t}$. Nous avons l'intuition que c et t sont tous deux petits devant la taille du réseau et semblent également être du même ordre de grandeur, avec possiblement t plus grand que c. Ainsi le nombre de communautés dans un réseau semble être de l'ordre du nombre de nœuds dans le réseau : $C = O(n)$.

Afin de justifier plus avant que ce nombre de communautés est de l'ordre du nombre de nœuds dans le réseau, voire plus petit, nous avons effectué des statistiques sur différents réseaux. Ces résultats sont exposés dans le tableau 6.1 où nous avons comparé le nombre de

1. Aussi appelée nombre de k-core. La dégénérescence d'un graphe G est le nombre maximum d tel qu'il existe un sous-graphe de G ne contenant que des nœuds de degré au moins d.

2. Une autre formule exacte sur laquelle on peut raisonner est la suivante, en posant com_i l'ensemble des communautés du nœud i et t_j la taille de la communauté j : $C = \sum_{i=1}^{n}(\sum_{j \in com_i} \frac{1}{t_j}).\sum_{j \in com_i} \frac{1}{t_j}$ peut être définie comme la quantité de communautés appartenant au nœud i.

Network	n	e	C	t	c	$C.t = c.n$
LiveJournal	3 997 962	34 681 189	287 512	22.63	1.60	6 414 555
Friendster	65 608 366	1 806 067 135	957 154	23.14	0.34	22 151 543
Orkut	3 072 441	117 185 083	6 288 363	14.16	28.98	89 053 454
Youtube	1 134 890	2 987 624	8 385	13.50	0.10	113 200
DBLP	317 080	1 049 866	13 477	53.41	2.27	719 820
Amazon	334 863	925 872	151 037	19.38	8.74	2 927 342

Tableau 6.1 – Nombre de nœuds, nombre de liens, le nombre de communautés, taille moyenne des communautés et nombre moyen de nœud par communauté "vérité de terrain" pour différents réseaux réels. Les différents réseaux sont présentés dans l'annexe A.

nœuds, le nombre de liens et le nombre de communautés "vérité de terrain" pour différents réseaux réels. Nous voyons ici que le nombre de communautés est toujours petit devant le nombre de liens ou de nœuds, sauf pour le réseau Orkut où le nombre de communautés est de l'ordre de deux fois le nombre de nœuds.

Il semble donc, par construction du réseau ou par propriété intrinsèque des communautés, que le nombre de celles-ci soit de l'ordre du nombre de nœuds dans le graphe, tout en étant inférieur à ce dernier. Il serait donc possible d'arriver à un algorithme qui, dans un réseau issu du terrain, retournerait en temps linéaire toutes les communautés du réseau (à condition de pouvoir les trouver en un temps constant). Tout du moins, le nombre de communautés ne constitue plus une excuse pour ne pas chercher cet algorithme. La construction d'un tel algorithme est notre quête et nous la détaillons dans ce chapitre.

6.3 Détection de la communauté d'un représentant

Comme nous l'avons montré dans le chapitre 3, si un nœud d'intérêt n'appartient qu'à une seule communauté et est important en son sein, c'est-à-dire si le nœud est un représentant de communauté, alors sa communauté peut être identifiée en utilisant une "approche mesure de proximité". Cette approche consiste en trois étapes :

1. calculer la proximité de chaque nœud du réseau au nœud d'intérêt en utilisant une mesure de proximité ad hoc,
2. trier ces valeurs par ordre décroissant,
3. chercher une structure en "plateau / décroissance / plateau" et identifier les nœuds avant la forte décroissance comme correspondant à la communauté du nœud d'intérêt.

En pratique, il suffit de chercher la plus grande pente dans la courbe proximité VS classement et de sélectionner les nœuds avant cette plus grande pente comme la communauté du nœud d'intérêt. La valeur de la plus grande pente peut permettre d'estimer dans quelle mesure la communauté est claire et à quel point le nœud d'intérêt en est un bon représentant.

Catégorie	candidat représentant	pente maximale	nombre de noeuds	F-score
Chess (3850 nœuds)	Chess	3.46	4179	0.89
	Magnus Calsen	3.69	4023	0.90
	Queens Gambit	4.37	4051	0.91
	World Chess Championship	4.04	4107	0.90
	Chess Boxing	1.35	48	0.00
Sumo (445 nœuds)	Sumo	3.03	508	0.88
	Taiho Koki	5.97	440	0.91
	Yokozuna	3.07	437	0.88
	Lyoto Machida	1.94	6458	0.08
Boxing (7289 nœuds)	Boxing	2.42	7572	0.79
	Cruiserweight	2.21	7572	0.78
	Vitali Klitschko	1.66	176	0.02
	Chess Boxing	1.34	48	0.01

Tableau 6.2 – Résultats obtenus sur le réseau *Wikipédia 2012*.

Nous avons à disposition un grand choix de mesures de proximité. Cependant, comme nous allons le voir, nous aurons besoin de répéter l'expérience pour chaque nœud du réseau. Nous avons donc besoin ici d'une mesure de proximité très rapide à calculer et dont le support [3] des valeurs est faible. L'opinion propagée présentée dans le chapitre 3 est trop coûteuse pour cette tâche car son temps de calcul est linéaire en la taille du réseau.

Ainsi, nous choisissons l'approximation du pagerank enraciné (sur le nœud d'intérêt) et normalisé par les degrés proposé dans [18, 17]. Nous utilisons une probabilité de revenir au nœud de départ $\alpha = 0.1$ qui est la valeur classiquement utilisée.

Dans les réseaux réels, les communautés semblent en général avoir un représentant, cependant prouver cette assertion ou tout du moins évaluer son niveau de véracité est difficile. Nous proposons ici de répéter l'expérience réalisée dans le chapitre 3 mais en utilisant comme mesure de proximité le pagerank enraciné normalisé par le degré. Le tableau 6.2 montre les résultats obtenus qui sont quasiment aussi bons que ceux obtenus par l'opinion propagée dans le chapitre 3 (en pratique ils sont un peu moins bons, cf. tableau 3.2, mais cette méthode permet de gagner en vitesse de calcul). Nous voyons que pour les trois communautés sélectionnées nous avons pu trouver un représentant et même plusieurs en réalité. Cela nous permet donc d'envisager un algorithme pour calculer toutes les communautés recouvrantes d'un graphe que nous présentons maintenant.

3. le nombre de valeurs non nulles.

6.4 Algorithme

6.4.1 Calcul des groupes

Notre algorithme prend en entrée un graphe et une mesure de proximité et retourne en sortie un ensemble de groupes de nœuds avec, pour chacun, un représentant (son étiquette) et un score évaluant sa pertinence en tant que communauté.

Pour chaque nœud du graphe, notre algorithme :

1. calcule la proximité à tous les autres nœuds du graphe,
2. trie ces valeurs de proximité par ordre décroissant,
3. calcule la plus grande pente en échelle logarithmique,
4. met de côté l'ensemble de nœuds situés avant la plus grande pente (la communauté) en lui associant le sommet avec lequel il a été trouvé (son représentant) et la valeur de la plus grande pente (évaluant sa pertinence en tant que communauté).

Comme nous l'avons vu précédemment, en utilisant le pagerank enraciné approximé [18], le calcul de la proximité d'un nœud à tous les nœuds se fait en temps constant (ne dépendant pas de la taille du graphe) et donc le support de ces valeurs ne dépend également pas de la taille du graphe. Ainsi le temps de cet algorithme est bien linéaire en fonction de la taille du graphe.

6.4.2 Nettoyage des groupes

Il est très probable que plusieurs nœuds aient donné lieu à des groupes de nœuds très similaires. Cela se produit, par exemple, si deux nœuds sont des représentants de la même communauté. Il est donc important de nettoyer ces groupes avant de retourner les communautés finales. Notre solution à ce problème est la suivante : si deux groupes sont trop similaires, nous proposons d'enlever le moins pertinent (celui dont la plus grande pente a été la plus faible). Pour cela, nous évaluons la similarité entre deux groupes avec le score de F_1 [4] qui est très utilisé à ces fins. Étant donnés deux ensembles A et B, ce score de similarité vaut :

$$F_1(A, B) = 2\frac{|A \cap B|}{|A| + |B|}.$$

Si ce score est supérieur à un certain seuil δ, alors nous enlevons la communauté avec la pente la plus faible. $\delta = 0.8$ ou 0.9 sont des valeurs judicieuses, l'idée étant d'enlever simplement des groupes trop similaires.

Pour garder un algorithme efficace, on ne peut pas se permettre de calculer la similarité entre tous les groupes deux à deux. En effet, étant donné qu'il y a un nombre linéaire de

4. Comme détaillé dans le chapitre 3, ce score est classiquement utilisé pour comparer des ensembles, on aurait très bien pu utiliser la similarité de Jaccard. Le F_1 peut être vu comme la moyenne harmonique de la précision et du rappel : $F_1(A, B) = \frac{2precision(A,B)rappel(A,B)}{precision(A,B)+rappel(A,B)}$ avec $precision(A, B) = \frac{|A \cap B|}{|B|}$ et $rappel(A, B) = \frac{|A \cap B|}{|A|}$.

groupes, cela donnerait un algorithme en temps quadratique qui ne pourrait pas traiter les plus grands graphes. Afin de pallier ce problème, nous proposons de calculer la proximité seulement entre deux groupes partageant au moins un sommet. Le calcul pour trouver toutes les paires de groupes partageant au moins un sommet se fait en temps linéaire. En effet, il suffit d'initialiser un tableau avec une entrée pour chaque nœud du graphe contenant une liste chaînée vide, puis de balayer les sommets de chaque groupe et d'ajouter à la liste de l'entrée correspondante au nœud courant un pointeur vers la communauté courante. On obtient ainsi pour une entrée du tableau un pointeur pour chaque communauté contenant le nœud en question. Remarquons qu'une paire de communautés partageant au moins un nœud peut apparaître plusieurs fois (en fait elle apparaît un nombre de fois égal au nombre de nœuds que les deux groupes partagent), mais ceci ne pose pas de problème : le temps pour former cette structure reste essentiellement linéaire en $O(cn)$ avec n le nombre de nœuds et c le nombre moyen de communautés qu'a un nœud, qui est petit devant la taille du réseau comme nous l'avons vu.

Calculer le score F_1 entre deux communautés se fait en temps proportionnel à la somme du nombre de nœuds dans ces communautés, sachant que l'on n'est pas obligé de calculer ce score entre deux communautés si on l'a déjà fait auparavant (on peut s'en rappeler en utilisant un dictionnaire [5]).

Le temps total de cet algorithme de nettoyage est donc en $O(tc^2n)$ où t est la taille moyenne d'une communauté et c^2 désigne ici la moyenne du carré du nombre de communautés qu'a un nœud. En pratique t et c^2 sont très petits devant la taille du réseau et leur produit reste petit devant lui.

6.4.3 Améliorations algorithmiques

Remarquons que pour le calcul des proximités, la boucle sur les nœuds est entièrement parallélisable sans aucune perte de performance, i.e., utiliser n processeurs/cœurs divise le temps par n.

Remarquons également qu'il est inutile de calculer cette mesure de proximité pour un nœud très proche d'un nœud pour lequel les proximités ont déjà été calculées. En effet, si deux nœuds sont très proches, leurs proximités à tous les nœuds du réseau auront des valeurs similaires et la détection de la plus grande pente résultera en la détection de deux groupes de nœuds similaires. Ainsi une fois que nous avons calculé la proximité d'un nœud donné à tous les nœuds du réseau, nous pouvons passer le calcul pour les k nœuds les plus proches, k étant un paramètre. Pour une faible valeur de k, cette technique permet grossièrement de diviser le temps par k, le gain diminue ensuite. Si k est trop grand, on peut passer à côté de certaines communautés.

La construction de la structure pour le nettoyage (liste des communautés incorporant chaque nœud du graphe) est parallélisable (une légère perte en temps peut cependant être observée puisque l'on peut potentiellement écrire dans les mêmes entrées du tableau).

5. une table de hachage

Le calcul des communautés trop similaires à l'aide de cette structure est là encore parallélisable.

Le fait que des communautés apparaissent plusieurs fois (moyennant un peu de bruit) est aussi un indicateur de la pertinence d'une communauté, cette valeur correspond en fait au nombre de représentant qu'a la communauté. On remarque en pratique qu'enlever les communautés n'apparaissant qu'une seule fois améliore les performances de la méthode : ces communautés sont souvent des faux positifs.

6.5 Validation

6.5.1 Métriques

Afin de comparer la performance de différentes méthodes les unes par rapport aux autres, il existe plusieurs métriques pour évaluer la similarité de l'ensemble des communautés trouvées à celle d'un ensemble de communautés "vérité de terrain".

Par exemple, la distance d'édition correspond au nombre minimum d'opérations élémentaires nécessaire pour transformer la structure communautaire trouvée en la structure communautaire "vérité de terrain". Les opérations élémentaires étant :

— l'ajout d'un nœud à une communauté,
— la suppression d'un nœud d'une communauté,
— la création d'une communauté avec un nœud,
— la suppression d'une communauté avec un nœud.

Cependant cette mesure donne des résultats peu intuitifs : avoir trouvé (resp. ne pas avoir trouvé) une grosse communauté qui n'existe pas (resp. qui existe) est bien plus pénalisant qu'en avoir trouvé (resp. ne pas en avoir trouvé) une petite qui n'existe pas (resp. qui existe).

Nous nous orientons donc plutôt vers d'autres métriques moins sensibles à ce genre de problème. Nous allons en présenter deux autres puis les combiner pour conduire à une métrique qui nous semble très pertinente. Notons $T = \{T_1, T_2, ..., T_t\}$ l'ensemble des communautés trouvées et $V = \{V_1, V_2, ..., V_v\}$ l'ensemble des communautés vérité de terrain.

La métrique la plus immédiate est certainement la moyenne du score F_1 des communautés trouvées en comparant chacune d'elles avec la communauté la plus similaire parmi les communautés vérité de terrain :

$$\mathcal{F}_1(T, V) = \frac{1}{t} \sum_{i=1}^{t} \max_{j \in [\![1,v]\!]} F_1(T_i, V_j).$$

Cette valeur illustre combien chaque communauté trouvée est similaire à au moins une communauté vérité de terrain. Remarquons que si une seule communauté a été trouvée et qu'elle est exactement égale à une communauté vérité de terrain, alors la valeur de ce score est de 1 (valeur maximale) et ceci même si il y a beaucoup de communautés vérité de terrain (des communautés qui ne sont pourtant pas trouvées dans notre exemple).

Ainsi, il est important d'associer cette métrique à une autre de manière à avoir un critère d'évaluation plus fiable.

Une métrique, similaire à la première (que l'on peut voir comme son inverse), est la moyenne du score F_1 des communautés vérité de terrain en comparant chacune d'entre elles avec la communauté la plus similaire parmi les communautés trouvées :

$$\mathcal{F}_1(V,T) = \frac{1}{v}\sum_{i=1}^{v} \max_{j\in[\![1,t]\!]} F_1(V_i,T_j).$$

Cette valeur illustre combien chaque communauté vérité de terrain est similaire à au moins une communauté trouvée.

Ici, inversement à la première métrique, si l'on a un très grand nombre de communautés trouvées, alors on aura plus de chances d'avoir un score élevé. En particulier un ensemble composé de tous les groupes possibles (les $2^n - 1$ groupes) aura un score de 1 (score maximal).

Remarquons également que pour la métrique $\mathcal{F}_1(T,V)$ (resp. $\mathcal{F}_1(V,T)$), des communautés vérité de terrain (resp. trouvées) peuvent être associées à plusieurs communautés trouvées (resp. vérité de terrain). Il est possible de pallier ce problème en utilisant un algorithme de mariage pour maximiser la valeur finale et non pas chaque valeur indépendamment, cependant le temps de cet algorithme risque d'être supra-linéaire (en fonction du nombre de communautés totales) et de ne pas passer à l'échelle. Nous verrons que ce problème (utiliser plusieurs fois la même communauté) n'en n'est pas vraiment un si nous combinons les deux scores.

Afin d'avoir une métrique plus juste, il est possible de combiner ces deux métriques ; ceci a été proposé dans [164] où les auteurs proposent d'en faire la moyenne. Ceci ne nous paraît pas pertinent, car les deux ensembles "extrêmes" cités précédemment (un ensemble d'une seule communauté exactement égale à une communauté vérité de terrain et un ensemble composé de tous les groupes possibles) auront tous deux un score d'un peu plus de 0.5 alors qu'ils devraient tous deux avoir un score proche de zéro. Nous proposons donc de faire plutôt la moyenne harmonique (plutôt que la moyenne géométrique) afin de satisfaire ce critère. Ceci est d'ailleurs la technique généralement adoptée pour d'autres scores, comme le score F_1 lui-même, si l'on fait l'amalgame entre $\mathcal{F}_1(T,V)$ et la sensibilité et $\mathcal{F}_1(V,T)$ et la précision.

Nous proposons donc d'utiliser la métrique suivante :

$$\mathcal{S}(T,V) = \frac{2\ \mathcal{F}_1(T,V)\ \mathcal{F}_1(V,T)}{\mathcal{F}_1(T,V) + \mathcal{F}_1(V,T)}.$$

6.5.2 Comparaison à d'autres méthodes de détection de communautés recouvrantes

Nous comparons ici notre méthode à trois méthodes de l'état de l'art pour la détection de communautés recouvrantes à savoir :

Réseau et nombre de communauté	k-clique percolation			BigClam			partition de liens			approche proximité		
scholar	0.14	0.16	0.15	0.07	0.00	0.01	0.16	0.12	0.14	0.13	0.13	0.13
16773	26318 (k=3)			100			11133			18845		
DBLP	0.00	0.03	0.00	0.05	0.10	0.07	0.00	0.03	0.00	0.32	0.19	0.24
24	3113 (k=5)			50			12871			15		
wiki08										0.67	0.03	0.04
79365										355		
wiki12										0.72	0.03	0.05
730752										2046		

Tableau 6.3 – Tableau montrant les scores $\mathcal{F}_1(T, V)$, $\mathcal{F}_1(V, T)$ et $\mathcal{S}(T, V)$ ainsi que le nombre de communautés trouvées pour les différentes méthodes.

1. la percolation de k-cliques [120, 85, 127],
2. BigClam [164],
3. la partition de liens [10].

Nous effectuons cette comparaison sur quatre grands réseaux réels pour lesquels on dispose d'une structure en communautés vérité de terrain. Ces réseaux sont détaillés en annexe : *Wikipédia 2008*, *Wikipédia 2012*, *Google scholar* et *DBLP*. Le tableau 6.3 montre ces résultats. Pour la percolation de k-cliques nous avons choisi le k donnant les meilleurs résultats en termes de $\mathcal{S}(T, V)$. Pour notre méthode, nous avons choisi $k = 10$ (ceci permet de ne pas calculer les proximités pour les k nœuds les plus proches d'un nœud pour lequel les proximités ont déjà été calculées) et $\delta = 0.8$ (pour ne pas prendre en compte une communauté similaire à plus de 80% à une communauté détectée selon le score F_1). Nous avons également sélectionné les communautés trouvées au moins deux fois et donc supprimé celles trouvées une seule fois.

Pour le réseau *Google scholar*, notre méthode donne des résultats similaires aux autres. On peut l'expliquer car d'une part, les communautés "vérité de terrain" sont de piètre qualité du fait de leur définition (annotation manuelle des chercheurs de leur domaine de recherche), d'autre part le réseau lui-même souffre du même problème. C'est pourquoi une méthode médiocre et une méthode parfaite pourraient avoir des résultats similaires.

Nous voyons que pour le réseau *DBLP* notre méthode est bien au-dessus des autres pour les 3 mesures. Soulignons que, de par leur définition, ce réseau et les communautés sont de meilleure qualité que dans le réseau précédent.

Pour les réseaux *Wikipédia 2008* et *Wikipédia 2012*, notre méthode donne de très bons résultats en termes de $\mathcal{F}_1(T, V)$, mais pas vraiment en termes de $\mathcal{F}_1(V, T)$, c'est-à-dire que les communautés que l'on trouve sont très similaires à des communautés "vérité de terrain", mais l'inverse n'est pas vrai. En fait, notre méthode retourne un faible nombre de communautés par rapport au nombre réel de communautés, ainsi beaucoup de communautés "vérité de terrain" ne trouvent pas leur équivalent dans les communautés trouvées et donc le score $\mathcal{F}_1(V, T)$ est faible. Ceci peut s'expliquer car nous avons considéré toute

catégorie (agrégée avec ses sous-catégories) comme étant une communauté, ce qui n'est pas le cas en pratique : beaucoup de catégories agrégées ne sont pas des communautés.

Les trois autres méthodes n'ont pas été capables de traiter ces réseaux Wikipédia : 64 Go de mémoire n'ont pas été suffisants pour la percolation de k-clique et la partition de liens. Quant à BigClam, il n'a pas terminé après deux semaines de calcul avec 4 processeurs (ce problème de BigClam a d'ailleurs déjà été constaté dans d'autres articles [160]).

6.6 Conclusion et perspectives

Nous avons montré que contrairement à ce que l'on pourrait penser, le nombre de communautés pertinentes dans un réseau est faible (de l'ordre du nombre de nœuds, voire plus petit) et la plupart de ces communautés semblent avoir un représentant, c'est-à-dire un nœud appartenant seulement à cette communauté et important en son sein. En prenant en compte ces observations, nous avons mis au point un algorithme permettant de détecter ces groupes pertinents à l'aide de la détection de la communauté ego-centrée de ces représentants. Notre approche exploite une mesure de proximité et un nettoyage intelligent de ces communautés ego-centréees. L'algorithme est plus rapide que ceux de l'état de l'art et donne de meilleurs résultats sur certains jeux de données.

Notre algorithme de nettoyage des communautés trouvées consistant à calculer la similarité des communautés partageant au moins un nœud est suffisamment rapide pour traiter en quelques jours des réseaux contenant des centaines de millions de liens comme le réseau *Wikipédia 2012* (ce que les autres algorithmes de l'état de l'art auxquels nous avons comparé notre méthode ne peuvent pas faire). Cependant notre algorithme de nettoyage (goulot d'étranglement en termes de vitesse d'exécution de notre algorithme) pourrait sûrement être amélioré en utilisant un algorithme de *Locality sensitive hashing* [93] dont l'idée principale est d'utiliser une famille de fonctions de hachage choisies telles que des points proches dans l'espace d'origine aient une forte probabilité d'avoir la même valeur de hachage. Ainsi des ensembles de communautés similaires pourraient être détectés plus rapidement (car elles auraient la même valeur de hachage) et le nettoyage gagnerait en vitesse par rapport à notre heuristique courante (consistant à comparer toutes les communautés partageant au moins un nœud). Nous travaillons actuellement sur cette méthode en utilisant la similarité de Jaccard pour comparer deux communautés et des fonctions de hachage associées à cette similarité : *MinHash* (ou *min-wise independent permutations locality sensitive hashing scheme*) [35].

Bien qu'il semble que beaucoup de communautés aient un représentant (comme par exemple les grandes catégories agrégées de Wikipédia comme "Boxing" ou "Chess" qui en ont au moins un), ce fait est sûrement faux pour certains réseaux ou certains types de communautés. Par exemple il semble que dans un réseau social chaque nœud appartienne à plusieurs communautés (groupe d'amis, famille, collègues). Pour ce type de réseaux une adaptation de la méthode est possible, par exemple en considérant des paires de nœuds comme graine et en cherchant des communautés bi-égocentrées (communauté définie par deux nœuds) en s'inspirant des travaux effectués dans le chapitre 4. Cependant un accrois-

sement du temps de calcul serait observé et une sélection intelligente des paires de nœuds devrait être effectuée. Nous laissons cette généralisation pour des travaux futurs.

Troisième partie

Travaux d'ouverture

Chapitre 7

Les capitalistes sociaux sur Twitter

7.1 Introduction

Contexte. Twitter est un service de micro-blogging très largement utilisé pour partager, rechercher et débattre des informations ou des évènements du quotidien [75]. Sa popularité ne cesse de croître : le nombre de ses utilisateurs est passé de 200 millions en avril 2011 [6] à 500 millions en octobre 2012 [7]. La quantité d'information échangée sur Twitter est considérable : 1 milliard de *tweets* -courts messages de moins de 140 caractères- sont postés tous les deux jours et demi [128]. Twitter est donc à la fois un service de micro-blogging et un outil média. Mais Twitter est également un service de réseau social en ligne. En effet, Twitter inclut de nombreux outils destinés à l'échange entre utilisateurs. Par exemple, pour voir les *tweets* d'autres utilisateurs s'afficher sur son fil d'actualité (*timeline* en anglais), il est nécessaire de s'abonner à ces utilisateurs. Si u s'abonne à v, on dit que u est un *abonné* (*follower* en anglais) de v. Réciproquement, v est un *abonnement* de u ($friend$ en anglais). De plus, un utilisateur peut *retweeter* [144] les *tweets* d'autres utilisateurs, par exemple pour partager avec ses abonnés une information qu'il trouve pertinente. Par ailleurs, les utilisateurs peuvent *mentionner* d'autres utilisateurs pour attirer leur attention en ajoutant `@NomUtilisateur` dans leur message. Il est également possible de répondre à un utilisateur lorsque l'on est mentionné dans un tweet. Une conversation sous forme de tweets est alors engagée.

L'augmentation du nombre d'abonnés, l'explosion du nombre de tweets par jour, l'importance de Twitter dans l'actualité et ses fonctionnalités sociales ont naturellement amené les entreprises et les académiques à s'intéresser à la notion d'influence sur ce réseau [21, 36, 130, 148, 159]. De nombreux paramètres peuvent être pris en compte pour mesurer l'influence sur Twitter. La plupart considère le nombre d'abonnés et d'interactions : les retweets et les mentions. Intuitivement, plus un utilisateur a d'abonnés ou plus il est retweeté et mentionné, plus son influence sur le réseau est considérée comme importante [36]. Différents outils ont ainsi été proposés dans l'industrie dans le but d'associer à chaque utilisateur un *score* qui illustre son influence sur le réseau. Parmi les plus utilisés, on retrouve Klout [2], Kred [3], Tweet Grader [4], Twitalyzer [5]. Dans tous les cas, l'al-

gorithme utilisé pour calculer le score d'un utilisateur est gardé secret, même si Klout et Kred fournissent quelques informations sans vraiment les détailler (voir par exemple [21]). Ce que l'on retient de ces informations, c'est que le nombre d'abonnés n'est pas un paramètre clé de l'algorithme. En effet, ces outils se concentrent sur les interactions entre un utilisateur et le réseau. Kred mesure ainsi deux paramètres différents, à savoir l'*influence* et l'*outreach level* (le niveau de rayonnement i.e., la portée). Selon Kred, l'influence augmente *lorsque quelqu'un vous retweete, vous mentionne ou vous répond.*

Capitalistes sociaux. Il a récemment été observé qu'un grand nombre d'utilisateurs du réseau Twitter tentent d'accroître leur nombre d'abonnés de façon artificielle [56, 69]. Ces utilisateurs appelés capitalistes sociaux détournent le principe d'abonnement sur Twitter en promettant des abonnements réciproques. Ils s'abonnent ainsi uniquement à leurs -futurs- abonnés, sans aucune considération pour le contenu tweeté par ces utilisateurs. Leur objectif est uniquement d'augmenter leur visibilité sur le réseau et sur les moteurs de recherche du réseau [69]. Ces comptes, qui sont des utilisateurs réels et actifs et non des robots ou des comptes automatisés, parviennent par ce moyen à obtenir des scores d'influence élevés. L'efficacité de leur méthode a notamment été illustrée par Dugué et Perez [56] qui ont créé un compte automatique reproduisant leur technique. Ce compte (*@Rain_bow_ash*) a rapidement gagné un grand nombre d'abonnés et a largement été retweeté et mentionné grace aux méthodes artificielles des capitalistes sociaux qui incitent aux retweets et aux mentions en échange d'abonnements.

Une question survient naturellement suite à ces observations : *étant donné que les nombres d'abonnés et les retweets de ces utilisateurs sont obtenus de façon artificielle et indépendante de la pertinence du contenu qu'ils tweetent, doivent-ils être considérés comme influents ?* Il semble évident que la réponse est non, mais nous verrons que les outils actuels de mesure d'influence ne sont pas, pour le moment, en accord avec cette réponse.

Travaux liés. Récemment, Messias et al. [105] ont étudié cette problématique de l'influence sur Twitter. Ils ont ainsi créé deux comptes agissant de façon automatique (des *bots*) selon des stratégies très simples. L'un des bots, qui tweetait de manière automatique à propos de sujets populaires obtint 500 abonnés et des scores Twitalyzer et Klout élevés. Il est intéressant de constater que les outils tels que Twitalyzer et Klout considèrent ces utilisateurs comme influents : ces comptes peuvent pourtant facilement être détectés comme automatiques en étudiant les sources utilisées pour poster les tweets ou le rythme régulier des posts [42].

Dans ces travaux, le problème de l'influence des capitalistes sociaux n'est pas traité. Dugué and Perez [56] observent que ces utilisateurs sont des comptes réels, très souvent administrés manuellement (dans un peu moins de 90% des cas) et qui parviennent néanmoins à gagner un grand nombre d'abonnés et à être largement retweetés.

Contributions. Dans ce chapitre, nous nous intéressons à la problématique de l'influence des capitalistes sociaux. Nous montrons que les outils actuels ne sont pas capables de distinguer les capitalistes sociaux des utilisateurs réguliers et qu'ils leur accordent ainsi à tort des scores d'influence potentiellement élevés. Pour ce faire, nous décrivons tout d'abord un premier jeu de données Twitter obtenu sur des hashtag dédiés au capitalisme social tels que *#TeamFollowBack* (section 7.3). En étudiant ces utilisateurs certifiés comme étant

des capitalistes sociaux selon Dugué et Perez [56], nous observons que certains d'entre eux sont considérés comme très influents par les outils de mesure tels que Klout et Kred (section 7.4.1). Afin de remédier à cela, nous mettons en place un classifieur discriminant les capitalistes sociaux des utilisateurs réguliers. Ce classifieur est basé sur la méthode de régression logistique, un outil d'apprentissage classique décrit dans la section 7.5.1. Cette méthode est capable de retourner la probabilité pour un utilisateur d'être un capitaliste social, ce qui nous permet de pondérer le score d'influence produit par Klout, comme nous le montrerons dans la section 7.5.2. Enfin, nous terminons en présentant dans la section 7.5.3 l'application en ligne que nous avons développée, qui permet d'estimer la probabilité pour un utilisateur d'être un capitaliste social.

7.2 Capitalistes sociaux

Les capitalistes sociaux ont tout d'abord été mis en lumière par Ghosh et al. [69] dans une étude de la capitalisation d'abonnés sur Twitter particulièrement axée sur les spammeurs. Le but des spammeurs, tout comme celui des capitalistes sociaux, est d'obtenir le plus grand nombre d'abonnés possible pour diffuser leurs messages de spam. Les auteurs observent avec surprise que les utilisateurs qui s'abonnent le plus aux spammeurs sont des utilisateurs réels et non des robots ou des spammeurs. Ces utilisateurs qui cherchent à augmenter à tout prix leur nombre d'abonnés mettent en place deux techniques très simples et basées sur des abonnements réciproques :

— Follow Me and I Follow You (FMIFY) : les utilisateurs assurent à leurs abonnés qu'ils s'abonneront à eux en retour.
— I Follow You, Follow Me (IFYFM) : au contraire, ces utilisateurs s'abonnent massivement à d'autres utilisateurs en espérant que ceux-ci s'abonnent à eux en retour.

Ils s'abonnent à d'autres utilisateurs sans considération pour le contenu de leurs tweets et sont donc nocifs pour le réseau. Ils rendent en effet plus difficile la détection d'utilisateurs et de contenus pertinents.

En se basant sur ces observations, Dugué et Perez [56] ont récemment fourni une méthode capable de détecter les capitalistes sociaux en utilisant une mesure de similarité appelée indice de chevauchement [138]. Cette mesure calcule la relation qui existe entre l'ensemble A des abonnements et l'ensemble B des abonnés de la façon suivante :

$$I(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)}$$

Intuitivement, un utilisateur avec un indice de chevauchement supérieur à 1 est abonné à un grand nombre de ses abonnés et a ainsi appliqué les principes mentionnés ci-dessus : **FMIFY** et **IFYFM**. Il est donc susceptible d'être un capitaliste social.

De plus, Dugué et Perez [56] ont montré que les capitalistes sociaux appliquent ces principes de capitalisme social d'une façon particulièrement efficace en insérant dans leur tweets des mots-clés (appelés *hashtags*) dédiés au processus de capitalisme social. Ces

Figure 7.1 – Timeline du capitaliste social "followback_707"

utilisateurs utilisent donc ces hashtags pour interagir entre capitalistes sociaux, par exemple des hashtags au nom explicite *#IFollowBack* ou *#TeamFollowBack* (voir figure 7.1). En utilisant ces hashtags, les capitalistes sociaux demandent explicitement des retweets ou des mentions en échange d'abonnements. Ils encouragent également les utilisateurs qui les retweetent à se suivre entre eux, un comportement illustré avec le hashtag *#FollowTrain*. Nous utilisons ces moyens d'identifier les capitalistes sociaux pour créer le jeu de données que nous détaillons dans la section suivante.

7.3 Jeu de données

Exemples positifs. Dans le but de constituer un jeu de données de capitalistes sociaux certifiés tels que ceux décrits par Dugué and Perez [56], nous avons collecté des tweets postés sur les hashtags *#TeamFollowBack*, *#instantfollowback* et *#teamautofollow* dédiés au capitalisme social. Nous avons identifié les utilisateurs ayant posté au moins trois tweets avec ces hashtags en quelques jours. Nous avons ainsi obtenu un échantillon d'à peu près 25.000 capitalistes sociaux.

Exemples négatifs. La première étape pour obtenir des exemples négatifs consiste à échantillonner Twitter de manière aléatoire. En effet, des utilisateurs choisis aléatoirement sont de façon très peu probable des capitalistes sociaux. Nous avons ainsi choisi aléatoirement 15.000 utilisateurs de ce réseau qui en contient plus de 550 millions d'après Twitter. Pour cela, nous avons choisi 15.000 entiers entre 0 et 550.000.000. Bien que les identifiants Twitter ne soient pas tous consécutifs, cela ne constitue pas un biais. Cependant, puisque la grande majorité de ces utilisateurs possède peu de connexions avec le reste du réseau, ils ne constituent pas un échantillon suffisamment pertinent. Nous avons ainsi choisi aléatoirement 55.000 utilisateurs parmi les abonnements de ceux-ci. Ces utilisateurs plus connectés et plus actifs sont très probablement des utilisateurs réguliers et nous fournissent donc nos exemples négatifs. D'après Dugué and Perez [56], les capitalistes

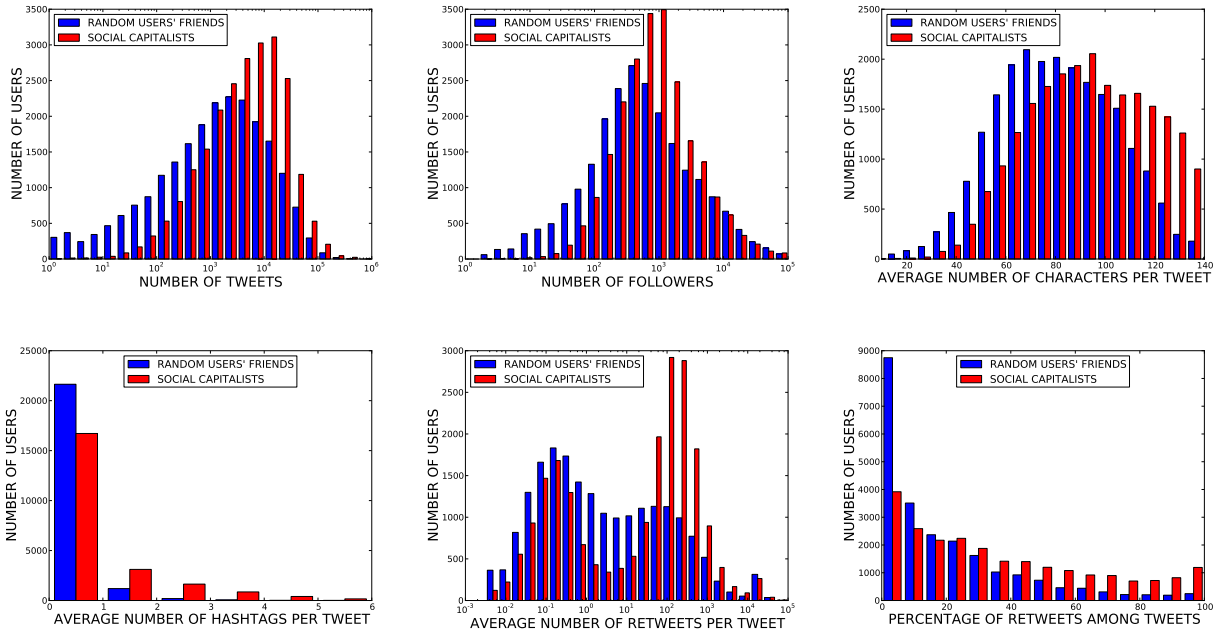


Figure 7.2 – Histogrammes montrant les statistiques sur quelques attributs pour les capitalistes sociaux d'une part et les amis d'un utilisateur aléatoire d'autre part.

sociaux représentent 0.2% du réseau. Ainsi, même en choisissant parmi les abonnements d'utilisateurs aléatoires, ce qui favorise le choix d'utilisateurs bien connectés comme les capitalistes sociaux, notre échantillon contient assurément un nombre négligeable de capitalistes sociaux (0.2% de notre échantillon représente 110 utilisateurs).

Ensuite, nous avons obtenu pour chaque compte un ensemble d'informations pertinentes pour les caractériser. Ces informations sont classées en différentes catégories (voir le tableau 7.1). Nous souhaitons mentionner que toutes ces données sont facilement obtenues par l'API REST fournie par Twitter. Ainsi, par exemple, les fonctionnalités *users/show*, *statuses/user_timeline* de cette API permettent d'obtenir des informations globales sur l'utilisateur (nombre d'abonnés, d'abonnements, de tweets postés, de favoris) mais aussi des informations sur le contenu de ses tweets. Remarquons aussi que les restrictions imposées par Twitter quant à l'utilisation de leur API nous poussent à considérer uniquement les 200 derniers tweets des utilisateurs.

Certaines de ces informations caractéristiques telles que les sources utilisées pour poster les tweets et le nombre d'url moyen par tweet sont efficaces pour séparer les comptes humains des comptes automatisés [42]. Cependant, Dugué et Perez [56] ont montré qu'un peu moins de 90% des capitalistes sociaux détectés sur ces hashtags n'automatisent pas leurs comptes. Il est ainsi nécessaire d'étudier d'autres types d'informations pour construire un classifieur robuste.

On observe que les données choisies semblent pertinentes pour discriminer les capitalistes sociaux des utilisateurs réguliers (voir la figure 7.2). En effet, on peut voir que la

Catégorie	Caractéristiques
Activité	Nombre de : 1 tweets 2 listes Twitter qui contiennent l'utilisateur 3 tweets favoris
Topologie locale	Nombre d' : 4 abonnements 5 abonnés 6 utilisateurs qui sont à la fois des abonnés et des abonnements
Contenu des tweets	Nombre moyen de : 7 caractères par tweet 8 hashtags par tweets 9 url par tweets 10 mentions par tweets
Caractéristiques des tweets	11 nombre moyen de retweets pour un tweet 12 nombre moyen de retweets pour un retweet 13 pourcentage de retweets parmi les tweets
Sources	Proportion d'utilisation d' : 14 application Twitter officielle 15 outil de gestion de compte 16 outil d'abonnement ou de désabonnement automatique 17 outil de post de tweet automatique 18 autres applications (Vine, Wiki, Soundcloud...) 19 applications smartphones ou tablettes

Tableau 7.1 – Description des différentes informations de compte étudiées.

distribution de certains attributs est différente pour ces deux types d'utilisateurs. Il est par exemple intéressant de constater que les capitalistes sociaux sont plus retweetés que les utilisateurs réguliers : le nombre de retweets est au cœur des scores Klout et Kred.

7.4 Mesurer l'influence sur Twitter

Déterminer l'*influence* d'un utilisateur sur Twitter est une question qui a soulevé beaucoup d'intérêt ces dernières années [21, 36, 130, 148, 159]. Puisque cette notion reste assez floue, de nombreux paramètres peuvent être utilisés pour mesurer cette influence : nombre d'abonnés, retweets, mentions, favoris par exemple. Afin de réaliser des scores plus élaborés, il est également possible de les combiner sous forme de ratios. Le plus simple et le plus intuitif est le ratio du *nombre d'abonnements sur le nombre d'abonnés*. Intuitivement, plus le résultat est proche de 0, plus les utilisateurs du réseau sont intéressés par le contenu fourni par l'utilisateur. Au contraire, si le résultat est bien supérieur à 1, l'utilisateur est susceptible d'être considéré comme s'abonnant en masse. Ce ratio peut néanmoins conduire à de mauvaises interprétations et est ainsi associé à des paramètres liés au degré d'*interaction* de l'utilisateur. Deux ratios sont ainsi considérés comme efficaces pour caractériser plus précisément l'influence d'un utilisateur : *le ratio Retweet et Mention* et *le ratio d'interaction* [21]. Le premier compte le nombre de tweets postés par un utilisateur qui sont retweetés ou donnent lieu à une conversation et le divise par le nombre de tweets. Le second considère le nombre d'utilisateurs distincts qui retweetent ou mentionnent l'utilisateur, divisé par son nombre d'abonnés. Anger and Kittl [21] définissent le *Social Networking Potential* d'un utilisateur Twitter comme la moyenne de ces deux ratios.

Dans ce chapitre, nous nous concentrons sur les outils disponibles en ligne les plus largement utilisés par les utilisateurs de Twitter et les entreprises comme mesures d'influence. Remarquons que ces outils restent populaires [1] malgré leurs défauts [111]. C'est en particulier le cas de Klout [2], Kred [3] et Twittalyzer [5], bien que ce dernier ait stoppé son activité commerciale en septembre 2013 (l'outil en ligne reste disponible). Klout mesure l'influence d'un utilisateur en se basant sur les principaux réseaux sociaux (par exemple Facebook, Linkedin et Instagram), pas uniquement sur Twitter. Il est cependant possible de savoir quels réseaux sont utilisés pour obtenir le score Klout d'un utilisateur. Nous préciserons cela lorsque nous utiliserons le score Klout dans ce chapitre. Dans la section suivante, nous montrons les limites des outils classiques de mesure de l'influence en montrant que ces outils considèrent certains capitalistes sociaux comme très influents.

7.4.1 L'impact du capitalisme social

Avec la promesse de suivre en retour les utilisateurs qui les suivent, les capitalistes sociaux parviennent à accroître efficacement leur nombre d'abonnés. Cependant, comme mentionné précédemment, ceci ne conduit pas nécessairement à une augmentation de leur score d'influence puisque que ce paramètre n'est pas considéré comme important par les

1. Klout a récemment été acheté par Lithium Technologies pour 200 millions de dollars [136].

screenname	Abonnements	Abonnés	I	Klout	Kred	Twitalyzer
teamukfollowbac	120,065	134,669	0.99	79	98.9	25.8
berge31	2,522	2,434	0.97	76	77,8	1
TheDrugTribe	26,266	28,832	0.99	69	98.2	27.2
globalsocialm2	5,603	5,624	0.81	69	95.1	3.3
repentedhipster	3,148	2,940	0.98	66	78,2	1
LIGHTWorkersi	112,963	103,475	0.99	66	96,2	22.4
ilovepurple__	49,666	52,448	0.97	65	97.5	22.9
TEAMF0LL0W	13,246	78,615	0.97	65	99.2	21.2
TEEMF0LL0W	10,977	92,412	0.97	64	99.3	21.1

Tableau 7.2 – Scores d'influence des capitalistes sociaux extraits de nos jeux de données : Klout, Kred et Twitalyzer. Indice de chevauchement I.

principaux outils de mesure. Néanmoins, avoir un grand nombre d'abonnés facilite bien entendu les interactions telles que les retweets et les mentions, qui sont quant à eux considérés comme des indicateurs d'influence. C'est surtout le cas pour les capitalistes sociaux qui postent des tweets dont le contenu demande des retweets et des mentions en échange d'un abonnement (voir la figure 7.1). Ce comportement entraîne un niveau élevé d'interaction pour ces utilisateurs, ce qui explique leurs bons scores d'influence.

Nous illustrons cela en calculant les scores Klout, Kref et Twitalyzer pour des capitalistes sociaux certifiés, soit extraits du jeu de données collecté en utilisant les hashtags décrits par Dugué and Perez [56], soit identifiés comme tels à cause de leur biographie ou de leur screenname explicite.

Dugué and Perez [56] ont observé que les utilisateurs avec un indice de chevauchement supérieur à 0.74 sont très fortement susceptibles d'être des capitalistes sociaux, ce qui est vérifié pour tous ces utilisateurs comme le montre le tableau 7.2. Ces capitalistes sociaux sont considérés comme influents par les trois mesures, malgré des comportements, des biographies ou même des noms parfois très explicites.

Le tableau 7.3 présente des résultats similaires pour des capitalistes sociaux détectés par la méthode de Dugué et Perez [56]. Dans chacun des cas, le score Klout est calculé uniquement à partir de l'activité Twitter. Ceci peut expliquer ces scores plus bas que ceux présentés dans le tableau 7.2. Néanmoins, ils restent de loin bien supérieurs à la moyenne. De plus, leurs scores Kred et Twitalyzer sont également relativement élevés.

Pour compléter les observations précédentes, nous comparons dans la figure 7.3 les différents scores de deux comptes populaires et actifs, ceux de **Barack Obama** et **Oprah Winfrey**, à ceux de capitalistes sociaux avérés ou de comptes automatiques (*Carina Santos*, le compte créé par Messias et al. [105]).

Les capitalistes sociaux ont des scores Kred et Twitalyzer similaires à ceux des comptes de référence, alors que leur score Klout est par contre plus faible. Cette différence peut s'expliquer par le fait que Klout utilise l'activité de plusieurs réseaux sociaux, ainsi que celle de Wikipédia. Barack Obama et Oprah Winfrey avec leur page Wikipédia bien documentée

Pseudo	Abonnements	Abonnés	Klout	Kred	Twitalyzer
EcheMadubuike	720, 407	732, 176	69	99	27.2
LarryWentz	600, 196	660, 260	60	90.9	20.2
machavelli7	567, 553	572, 161	68	94.8	20.9
zuandoemkta	566, 971	555, 476	60	94.4	20.6
_Follow_Friends	511, 818	540, 783	56	94.9	23.2
kosma003	438, 050	423, 638	52	95.1	20.2
ceebee308	360, 163	382, 568	60	98.1	22.6
ClimaWorld	384, 365	375, 419	60	90	20.4
radiotabu	306, 066	336, 752	55	97.7	21.5
Nteratedetodo	300, 278	323, 236	56	95.6	26.7

Tableau 7.3 – Scores d'influence (Klout, Kred and Twitalyzer) de capitalistes sociaux détectés par Dugué and Perez.

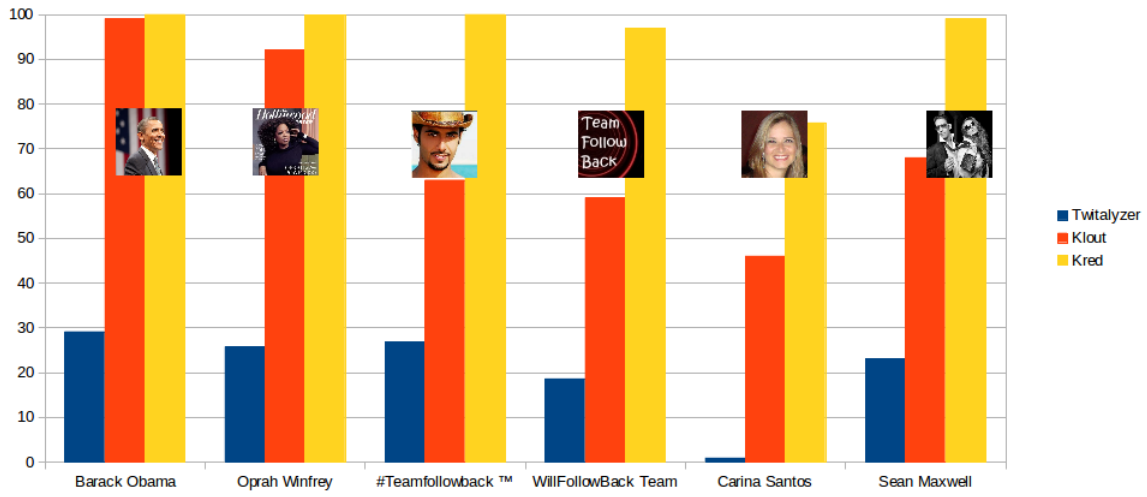


Figure 7.3 – Comparaison des trois mesures pour les comptes de **Barack Obama**, **Oprah Winfrey** et ceux de capitalistes sociaux avérés.

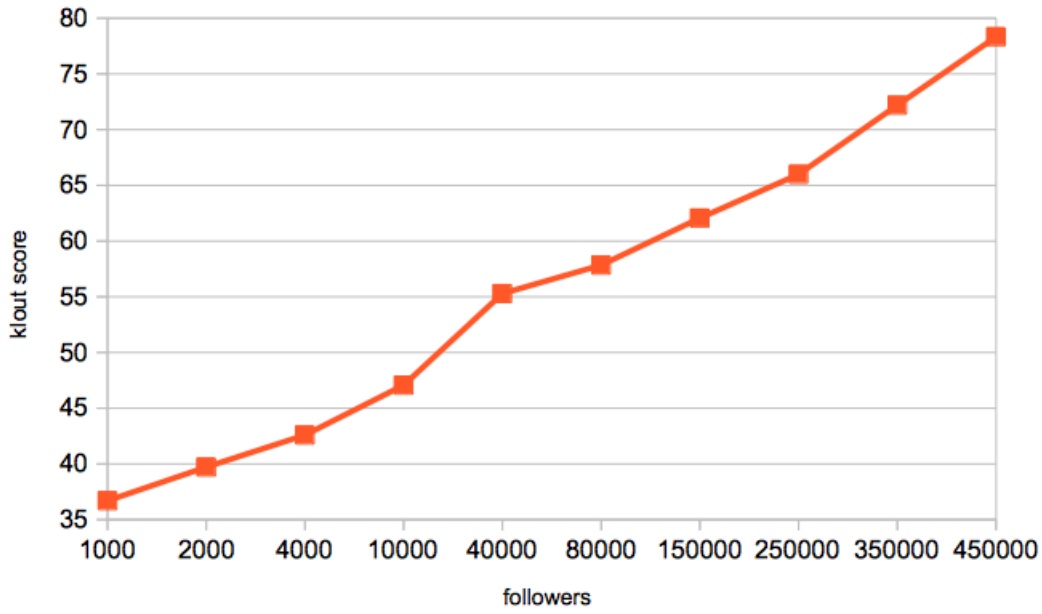


Figure 7.4 – Scores Klout moyens en fonction d'une borne inférieure sur le nombre d'abonnés.

et consultée accèdent donc à un score plus élevé que les capitalistes sociaux.

Excepté Carina Santos, les comptes que nous considérons sont des capitalistes sociaux avérés et très faciles à identifier : leur nom et biographie sont explicites. Par ailleurs, ces comptes tweetent *exclusivement* du contenu lié au capitalisme social et interagissent avec les autres utilisateurs dans l'unique but de gagner des abonnés. Ces utilisateurs ne produisent aucun contenu pertinent et sont des capitalistes sociaux évidents. Pourtant, aucun des outils en ligne n'est capable de détecter ces comportements, et ils sont donc considérés comme influents.

Pour conclure cette section, nous mettons en relation le score Klout moyen d'exemples positifs de notre jeu de données et le nombre d'abonnés de ces utilisateurs. Rappelons que Klout considère le nombre de retweets et de mentions comme plus important que le nombre d'abonnés. Cependant, comme le montre la figure 7.4, le nombre d'abonnés des utilisateurs est fortement corrélé au score Klout. Ceci s'explique par le fait que les capitalistes sociaux réussissent à obtenir plus d'interactions et de visibilité à mesure que leur nombre d'abonnés augmente. Leurs méthodes deviennent donc de plus en plus efficaces.

Nous pensons que ce manque de considération du capitalisme social entraîne la course aux abonnés sur Twitter. Par ailleurs, il semble important d'être capable de distinguer les utilisateurs réellement influents des capitalistes sociaux pour être à même d'accéder à des contenus pertinents et fouiller efficacement la masse de tweets produite par les utilisateurs Twitter.

7.5 Une nouvelle approche pour mesurer l'influence

Nous présentons maintenant la contribution principale de ce chapitre, c'est-à-dire, un classifieur qui permet de discriminer les capitalistes sociaux (exemples positifs) des utilisateurs normaux (exemples négatifs). De plus, l'outil que nous utilisons (la régression logistique) permet d'obtenir, pour un utilisateur donné, une estimation de la probabilité qu'il soit un capitaliste social. Cette probabilité peut également indiquer à quel point l'utilisateur est un capitaliste social, c'est-à-dire une mesure de "capitaliste socialitude". Nous utiliserons cette mesure afin d'équilibrer le score d'influence donné par Klout, cf. équation 7.1, ce qui est un premier pas vers la proposition d'une nouvelle mesure d'influence sur Twitter. Comme nous allons le voir dans la section 7.5.2, un tel équilibrage permet de réduire efficacement le score d'influence des capitalistes sociaux tout en conservant le score d'influence des utilisateurs normaux.

7.5.1 Classification des capitalistes sociaux

Dans un récent travail, Dugué et Perez [56] ont développé un algorithme performant pour discriminer les capitalistes sociaux des utilisateurs normaux. Cependant, leur but était d'utiliser seulement des attributs topologiques. Bien que ces travaux soient intéressants d'un point de vue théorique, il semble clair que l'on peut améliorer les résultats en utilisant également d'autres attributs, par exemple basés sur le contenu des tweets. Nous avons donc utilisé le jeu de données décrit dans la section 7.3 contenant 77 102 utilisateurs, dont 22 845 capitalistes sociaux et 54 257 utilisateur normaux, chaque utilisateur étant décrit par les attributs détaillés dans la section 7.3.

Nous avons partagé aléatoirement le jeu de données en un ensemble d'apprentissage de 70% du jeu total, utilisé pour entraîner le classifieur, et un ensemble de test de 30% des données, utilisé pour évaluer les performances du classifieur. Nous avons utilisé les algorithmes classiques développés en apprentissage, à savoir les K-plus proches voisins (KNN), la machine à vecteurs de support (SVM), les forêts aléatoires (RF) et la régression logistique (LR) [30]. Nous avons implémenté ces algorithmes à l'aide de la bibliothèque python sklearn [123].

Nous avons obtenu une bonne précision [2] sur l'ensemble de test avec tous ces classifieurs, RF donnant des résultats légèrement meilleurs et KNN légèrement moins bons, ce qui corrobore les résultats obtenus dans [103]. Nous avons cependant choisi LR puisque ce classifieur est spécifiquement conçu pour estimer la probabilité qu'un exemple soit positif (résultat que nous utiliserons pour équilibrer le score Klout) et que, une fois ajusté, il est très portable et facile à incorporer dans une application en ligne (cf section 7.5.3) car il nécessite seulement le stockage de quelques paramètres.

Afin d'obtenir de meilleures performances, nous avons transformé les attributs de la manière suivante : (i) nous avons utilisé un attribut constant additionnel, (ii) nous avons passé certains attributs en échelle logarithmique quand cela était nécessaire et (iii) nous

2. La précision est la proportion d'exemples correctement étiquetés.

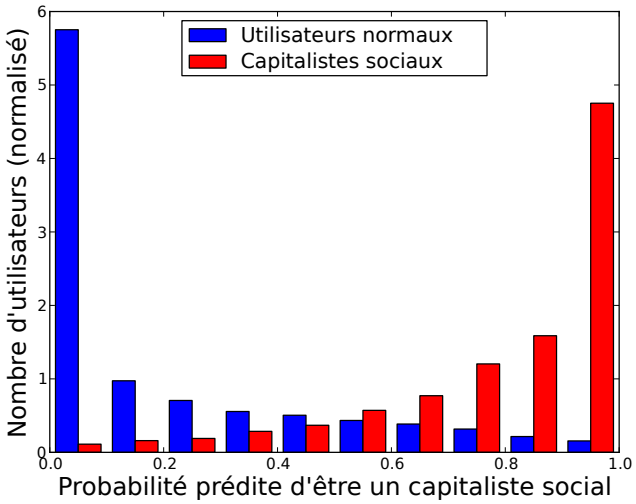


Figure 7.5 – Histogramme des probabilités prédites d'être un capitaliste social pour les utilisateurs normaux et les capitalistes sociaux.

avons utilisé les produits des attributs deux à deux comme attributs supplémentaires.

Après avoir ajusté les paramètres du classifieur sur l'ensemble d'apprentissage, nous avons évalué ses performances sur l'ensemble de test. La figure 7.5 montre l'histogramme des probabilités prédites d'être positif. Nous voyons qu'il y a une très forte corrélation entre la probabilité prédite et le fait qu'un exemple soit effectivement un capitaliste social ou un utilisateur normal. De plus, en coupant les probabilités à 0.5 afin d'obtenir un classifieur binaire, nous obtenons un F-score[3] de 87%. Cela veut dire qu'avec les quelques attributs détaillés dans la section 7.3, on peut prédire efficacement si un utilisateur est un capitaliste social ou non.

Dans le but de caractériser plus précisément la pertinence de chaque attribut, nous les avons évalués indépendamment : aucun attribut n'a permis d'obtenir un F-score de plus de 60%. Cela signifie qu'aucun attribut pris individuellement n'arrive à classifier les utilisateurs beaucoup plus efficacement qu'un classifieur aléatoire. Nous avons aussi évalué indépendamment la pertinence des 5 groupes d'attributs, le plus pertinent étant celui lié aux activités de l'utilisateur.

7.5.2 Rééquilibrage du score Klout

Nous utilisons ici la probabilité prédite P_{Ksoc} qu'un utilisateur soit un capitaliste social pour rééquilibrer le score Klout. On peut en effet assimiler cette probabilité à une évaluation

3. La sensibilité est la proportion d'exemples positifs correctement étiquetés. La spécificité est la proportion d'exemples négatifs correctement étiquetés. Le F-score est la moyenne harmonique entre la sensibilité et la spécificité ; il est meilleur pour quantifier la performance d'un classifieur binaire que la précision dans le cas de classes déséquilibrées.

du degré de capitalisme social d'un utilisateur, et ainsi abaisser le score Klout S_{Klout} des utilisateurs ayant un P_{Ksoc} trop élevé. Comme mentionné plus haut, nous utilisons le score Klout car c'est la mesure d'influence la plus répandue. Cependant, tout autre score peut être utilisé.

$$S_{DDP} = \begin{cases} S_{Klout} & \text{si } P_{Ksoc} \leq 0.5 \\ 2(1 - P_{Ksoc})S_{Klout} & \text{si } P_{Ksoc} > 0.5 \end{cases} \tag{7.1}$$

Nous avons évalué ce nouveau score sur des capitalistes sociaux, des utilisateurs normaux tirés aléatoirement de notre ensemble de test, ainsi que sur d'autres utilisateurs influents de twitter (cf. tableau 7.4).

Nom	**score Klout**	P_{Ksoc}	S_{DDP}
barackobama	99	$8.42 \cdot 10^{-4}$	99
oprah	93	$5.86 \cdot 10^{-9}$	93
followback_707	64	0.999	1
seanmaxwell	69	0.937	9
scarina91	46	0.110	46
teamukfollowbac	80	0.838	26
TEEMF0LLOW	69	0.416	69
zuandoemkta	62	0.861	18
kosma003	53	0.747	27
NicolasDugue	33	0.120	33

Tableau 7.4 – Scores Klout et DDP et P_{Ksoc}.

Comme nous pouvions nous y attendre, le classifieur ne considère pas **Barack Obama** et **Oprah Winfrey** comme étant des capitalistes sociaux. Notons que la méthode de détection de Dugué and Perez [56] considère Barack Obama comme un capitaliste social *passif*, c'est-à-dire, un utilisateur qui a fait du capitalisme social mais a arrêté depuis un certain temps. Cela est donc cohérent avec nos résultats, puisque nous cherchons ici des capitaliste sociaux plutôt actifs et que nous considérons seulement les 200 derniers tweets.

Quand nous considérons des exemples positifs pris de notre ensemble de test (comme **teamukfollowbac**), nous observons que sa probabilité prédite d'être un capitaliste social est élevée. Ainsi, le score Klout rééquilibré reflète cette observation et considère l'utilisateur moins influent que le score Klout. Ceci est encore plus frappant avec **seanmaxwell** et **followback_707**, deux autres utilisateurs positifs extraits de notre ensemble de test. En effet, ces deux utilisateurs tweetent afin de joindre d'autres capitalistes sociaux et d'obtenir des followers ; ce comportement ne devrait pas être considéré comme influent sur Twitter. Le score Klout rééquilibré prend en considération ces remarques et diminue/réduit l'influence de ces utilisateurs. **TEEMF0LL0W** est un exemple positif sur lequel les résultats du classifieur ne sont pas particulièrement convaincants. Cet utilisateur tweete seulement des messages très courts comportant uniquement des hashtags, ce qui diffère de l'activité

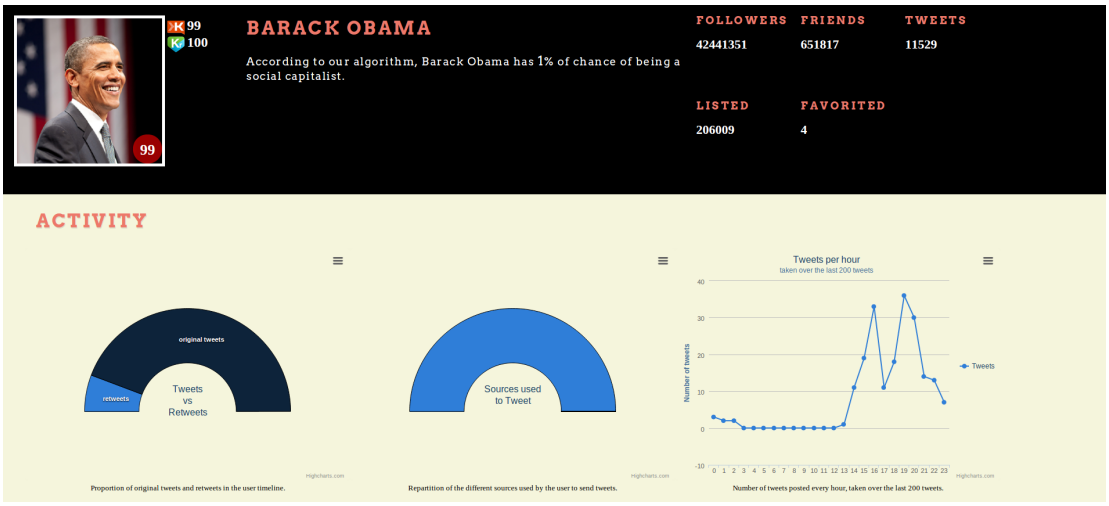


Figure 7.6 – Capture d'écran illustrant notre application en ligne.

des autres capitalistes sociaux et pourrait expliquer les problèmes rencontrés par notre classifieur.

7.5.3 Application en ligne

Pour compléter notre travail, nous avons développé une application en ligne pour calculer la probabilité de n'importe quel utilisateur de Twitter d'être un capitaliste social. Cette application est disponible à l'adresse `http://www.bit.ly/DDPapp` (notons qu'il faut avoir un compte Twitter pour utiliser l'application). La figure 7.6 en montre une capture d'écran.

Étant donné le nom d'utilisateur d'un compte Twitter, l'application calcule les attributs décrits dans le tableau 7.1 et utilise ensuite le classifieur LR afin de calculer la probabilité estimée que l'utilisateur de ce compte soit un capitaliste social. Les attributs sont extraits des 200 derniers tweets de l'utilisateur en question. Cette restriction vient de la limite imposée par l'interface de programmation (API) de Twitter. En haut à gauche de la page web, les scores Klout et Kred sont affichés. Le score Klout rééquilibré est affiché sur la photo du profil de l'utilisateur. En plus des informations basiques du compte Twitter (nombre d'amis, de followers, de tweets, de listes et de favoris), des informations plus complexes sont affichées, telles que la proportion de tweets originaux et de retweets, les sources utilisées pour poster les tweets et l'heure à laquelle les tweets ont été postés.

Afin d'obtenir une réponse rapide, nous avons fait un compromis concernant les attributs utilisés. La modification principale est que l'attribut 6 correspondant au nombre d'utilisateurs qui sont à la fois amis et followers n'est pas utilisé. Cet attribut nécessite d'extraire tous les amis et les followers d'un utilisateur, ce qui est très long à cause des restrictions de l'interface de programmation Twitter. Cependant, cette modification ne change pas significativement la précision de notre classifieur.

7.6 Conclusion et perspectives

Dans ce travail nous avons mis en parallèle différents outils utilisés pour mesurer l'influence des utilisateurs sur Twitter [2, 3, 5] avec de récents travaux menés sur les capitalistes sociaux [56, 69]. Notre étude montre que beaucoup de ces utilisateurs ne tweetent sur aucun autre sujet que le capitalisme social et obtiennent pourtant des scores d'influence élevés. En effet, les mesures d'influence actuelles ne prennent pas en compte l'impact du capitalisme social sur le réseau Twitter. Afin de pallier ces limitations, nous avons développé un classifieur qui prend en compte des attributs topologiques du réseau sous-jacent comme le nombre d'amis, le nombre de followers et le nombre d'utilisateurs à la fois amis et followers ainsi que d'autres attributs extraits des tweets de l'utilisateur, de son activité, etc. Notre classifieur détecte efficacement les capitalistes sociaux et étend des travaux antérieurs basés seulement sur des attributs topologiques [56]. Nous utilisons ensuite les prédictions du classifieur afin de rééquilibrer le score Klout, qui est l'outil le plus utilisé pour mesurer l'influence sur Twitter. Ceci permet de réduire le score Klout des capitalistes sociaux tout en conservant le score d'influence des utilisateurs normaux. Nous avons également développé une application en ligne qui permet d'appliquer nos travaux à n'importe quel utilisateur de Twitter.

Ce travail peut être étendu de plusieurs manières. Avant tout, à cause des limitations de Twitter, notre jeu de données est relativement petit. En construisant un jeu de données plus grand, ce qui est long mais simple, on pourrait construire un classifieur encore plus robuste et également y incorporer d'autres attributs.

De plus, nous nous sommes concentrés dans ce chapitre sur le développement d'un outil pour discriminer les capitalistes sociaux des utilisateurs normaux en fonction de certains attributs. Nous avons ensuite utilisé cet outil pour rééquilibrer le score Klout. Il serait intéressant de construire une nouvelle mesure d'influence en partant de zéro et en utilisant ces attributs et éventuellement d'autres. De plus ces travaux montrent l'importance d'utiliser d'autres attributs que ceux mis en avant dans les mesures d'influence classiques, tels que le nombre de retweets. En effet, du fait de la forte corrélation entre le nombre de retweets et le nombre de followers, le nombre de retweets seul ne peut pas être considéré comme un bon indicateur d'influence. Une piste intéressante pour évaluer l'influence d'un utilisateur serait d'utiliser comme indicateur le nombre de retweets effectués par des utilisateurs qui ne sont pas amis avec lui.

Pour finir, ce travail est ciblé sur le réseau Twitter sur lequel les capitalistes sociaux ont été mis en évidence par Ghosh et al. [69]. D'autres applications avec une composante sociale telles qu'Instagram ou Youtube pourraient avoir les mêmes propriétés et nécessiter une amélioration des outils de mesure d'influence.

Chapitre 8

Génération directe de graphes aléatoires avec distribution de degrés prescrite

8.1 Introduction

Quand des études de réseaux sociaux réels sont effectuées et que des méthodes et métriques d'Analyse de Réseaux Sociaux (Social Networks Analysis ou SNA [156]) sont utilisées, l'évaluation des résultats est un aspect très important de la démarche scientifique. Une méthode d'évaluation consiste à comparer les propriétés des réseaux étudiés à celles de graphes aléatoires. Cette méthodologie a été particulièrement mise en avant en 1998 dans l'article de Watts et Strogatz [157], dans lequel ils ont montré que le "phénomène petit-monde", c'est-à-dire que la faible distance moyenne entre les nœuds, est présent dans tout graphe avec un peu d'aléatoire ; il s'agit donc d'une propriété naturelle des graphes de terrain qui ne permet pas de les discriminer. Ceci a eu un fort impact en promouvant l'évaluation par comparaison aux graphes aléatoires.

Une découverte importante concerne la distribution de degrés des graphes de terrain. Il a en effet été montré que la plupart des graphes de terrain ont une distribution fortement hétérogène, en loi de puissance [62], à l'opposé des distributions en loi de Poisson comme observées dans le modèle de graphes aléatoires d'Erdos-Renyi [60]. Cela incite donc à générer des graphes aléatoires avec une distribution de degrés en loi de puissance puis à les comparer avec les graphes réels. Les propriétés similaires dans les deux cas sont considérées comme triviales, c'est-à-dire qu'elles découlent directement de la distribution de degrés, alors que les propriétés qui diffèrent indiquent l'existence d'une propriété discriminante ou d'une anomalie dans le réseau étudié.

Afin de pouvoir mener ces analyses de manière correcte, il est donc nécessaire de pouvoir générer des graphes aléatoires avec une distribution de degrés fixée. Dans ce chapitre, nous allons étudier des algorithmes existants qui génèrent des graphes aléatoires avec une distribution de degrés fixée et proposer une nouvelle méthode permettant de les générer de

manière directe, c'est-à-dire sans revenir en arrière, et quasi uniformément, c'est-à-dire que chaque graphe a la même probabilité d'être généré. La génération directe permet d'ajuster la génération pour des cas spécifiques, ce qui ouvre de nouvelles possibilités pour une évaluation plus fine et la génération de benchmarks synthétiques.

Après une présentation rapide des notations utilisées, dans la section 8.2, nous dresserons l'état de l'art dans la section 8.3. Nous décrivons ensuite notre algorithme et le discutons dans la section 8.4 avant de présenter des implications sur l'évaluation de réseaux dans la section 8.5. Nous concluons enfin dans la section 8.6.

8.2 Notations

On note $G = (V, E)$ un graphe avec V un ensemble de nœuds et $E \subset (V \times V)$ un ensemble de liens, dirigés ou non. $n = |V|$ est le nombre de nœuds du graphe et $m = |E|$ est le nombre de liens.

Étant donné un lien dirigé $e = (s, t)$, on appelle s la *source* et t la *destination* du lien. La fonction $succ(v)$ retourne tous les successeurs de v, c'est-à-dire tous les nœuds w tels qu'il existe un lien (v, w) et la fonction $pre(v)$ retourne tous les prédécesseurs de v, c'est-à-dire tous les nœuds w tels qu'il existe un lien (w, v).

Avec une distribution de degrés fournie pour tous les nœuds d'un graphe non dirigé, nous attribuons initialement à chaque nœud le nombre d'extrémités (*stubs*) qui doivent y être connectées afin que la distribution de degrés soit réalisée. Durant la génération aléatoire, les extrémités sont reliées pour former des liens. Dans les cas des graphes dirigés, on attache des extrémités entrantes (*in-stubs*) et sortantes (*out-stubs*). Les fonctions $ost(v)$ et $ist(v)$ retournent respectivement le nombre restant d'extrémités sortantes et entrantes de v.

8.3 État de l'art

Dans cette section, nous présentons l'état de l'art dans le domaine des modèles aléatoires et des algorithmes de génération de graphes aléatoires, afin de mieux mettre en lumière les bénéfices de l'approche que nous présenterons par la suite.

8.3.1 Modèles de graphes aléatoires

Dans le cadre du modèle de graphe aléatoire d'Erdos-Renyi [60], les mathématiques et la physique ont été les premières disciplines à étudier les graphes aléatoires et les modèles probabilistes de génération de graphes aléatoires. Ces études avaient initialement pour objectif d'étudier les propriétés locales et globales des graphes quand n tendait vers l'infini. On peut lire [33] pour une étude plus complète des travaux dans cette direction.

Le problème majeur des graphes aléatoires définis par Erdos et Renyi dans le cadre de la modélisation de réseaux sociaux est que ces graphes possèdent une distribution de degrés

en loi de Poisson. Or, des études ont montré que la grande majorité des graphes de terrain ont une distribution fortement hétérogène qui suit asymptotiquement une distribution en loi de puissance. Ces réseaux sont souvent appelés des réseaux sans-échelle (*scale-free networks*) [26].

Ces observations et leurs conséquences pratiques ont mené à l'étude de nouveaux modèles de graphes aléatoires qui peuvent être paramétrés afin de mieux correspondre à des distributions de degrés données [155]. Il existe également des modèles respectant à la fois une distribution de degrés prescrite et d'autres propriétés topologiques [112]. Ces modèles sont très intéressants car ils peuvent être étudiés formellement et permettent d'obtenir des résultats, notamment asymptotiques, sur les propriétés globales ou locales des graphes générés.

À la suite de l'article de Watts et Strogatz [157], les chercheurs en analyse de réseaux sociaux ont pris l'habitude de comparer des réseaux réels à des graphes générés aléatoirement afin d'identifier des différences structurelles non triviales. Le paramétrage pour la génération de ces graphes pour un réseau donné, ainsi que l'étude formelle de nouvelles propriétés d'intérêt sont des problèmes de recherche généralement complexes.

C'est l'une des raisons pour lesquelles les études pratiques utilisent toujours des méthodes exploratoires et descriptives qui sont très utiles pour débuter une étude, mais ne permettent pas de tirer des conclusions fortes. L'état de l'art actuel utilise des instances de graphes générés aléatoirement et compare les propriétés de ces graphes avec celles de graphes réels (avec des méthodes de statistique descriptive). Ceci fournit des résultats suffisamment concluants sous réserve d'avoir assez d'échantillons.

Nous ne souhaitons pas décourager l'étude des modèles aléatoires. Mais, reconnaissant que leur utilisation est très difficile et qu'il y a toujours de nombreuses propriétés pour lesquelles aucun résultat formel n'est disponible, nous nous concentrons sur des évaluations plus classiques avec des graphes générés aléatoirement.

8.3.2 Génération de graphes aléatoires

Tout d'abord il faut distinguer deux types de génération aléatoire. La première, la génération d'instances de modèles, est utilisée pour des questions générales, par exemple l'étude empirique des modèles ou l'impact des paramètres. Pour ces modèles, les réseaux peuvent être générés très efficacement [29] en utilisant les propriétés mathématiques des distributions de degrés. Le second type de génération nécessite une distribution de degrés donnée que l'on souhaite obtenir exactement. Cela sert pour l'évaluation de graphes de terrain particuliers et est l'objectif de ce chapitre. Cela nécessite d'utiliser une approche différente que nous étudions dans la suite.

Milo et al. effectuent un très bon survol de ce domaine [106] ; nous utiliserons leur terminologie pour la présentation et la discussion de notre algorithme.

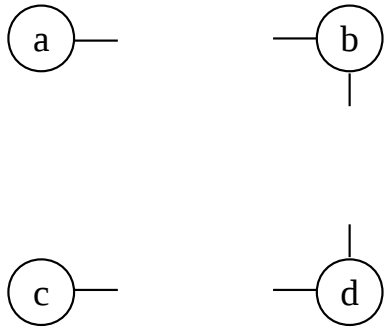


Figure 8.1 – Exemple de graphe avec extrémités pour la génération de liens.

Le modèle configurationnel

L'approche la plus classique est le *modèle configurationnel*, qui est bien résumé par Newman [114]. La génération d'un graphe avec distribution de degrés fixée est assez simple et s'effectue en choisissant aléatoirement des paires d'extrémités et en les connectant pour former des liens. Cet algorithme est la méthode développée par défaut dans la plupart des outils disponibles qui permettent de générer des graphes avec distribution de degrés fixée, par exemple NetworkX `http://networkx.lanl.gov/` pour Python.

Cependant, cette méthode souffre d'un inconvénient majeur pour les études pratiques car elle permet de générer des graphes avec des boucles (si un lien est créé avec deux extrémités d'un même sommet) ou des liens multiples (si deux paires d'extrémités sont choisies pour la même paire de sommets). Dans les réseaux réels ces propriétés sont souvent interdites i.e., les réseaux réels sont souvent des graphes simples [1]. L'utilisation de ce modèle ne permet pas des comparaisons rigoureuses. La figure 8.1 montre un graphe non dirigé avec une distribution de degrés fixée pour lequel on souhaite créer des liens aléatoires. Avec le modèle configurationnel, toute paire d'extrémités peut être choisie, ce qui autorise initialement les 8 connexions différentes. Si on se limite aux graphes simples, par contre, il y a seulement 5 connexions possibles. En effet, (b, b) et (d, d) créeraient une boucle et (a, c) imposerait de créer un lien multiple entre b et d par la suite.

Ce problème décroît quand n augmente mais en utilisant le modèle configurationnel il faut toujours supprimer les boucles et les liens multiples *a posteriori*, ce qui a pour conséquence de modifier la distribution de degrés fournie initialement. Un algorithme basé sur une telle modification est évalué par Milo et. al. [106], sous le nom de *matching algorithm*. La création de liens multiples n'arrête pas l'algorithme mais de tels liens sont rejetés. Cela augmente la probabilité de se rapprocher de la distribution prescrite. Cependant le rejet des liens multiples a pour conséquence la perte de l'uniformité des graphes générés. Viger et Latapy [153] ont montré empiriquement que cela introduit un biais dans les propriétés du graphe généré. Au contraire, Milo et al. défendent que les conséquences sont relativement faibles dans leurs expériences. Ils recommandent cependant d'utiliser une

1. Un graphe simple est un graphe sans boucle et sans lien multiple

Méthode de Monte-Carlo par chaînes de Markov (*Markov Chain Monte Carlo* (MCMC) en anglais).

Méthode de Monte-Carlo par chaînes de Markov

Comme affirmé par Viger et Latapy [153] :

> Although it has been widely investigated, it is still an open problem to directly generate such a random graph, or even to enumerate them in polynomial time [...]

Cette énumération a été accomplie par Snijders [139] mais, à cause de la complexité temporelle exponentielle, la plupart des chercheurs se sont tournés vers les méthodes de *Monte Carlo* pour la génération de graphes aléatoires.

Selon Milo et al. [106], la méthode la plus rapide est *Markov Chain Monte Carlo* (MCMC). Cette méthode a l'avantage supplémentaire de permettre la création de graphes simples, et même connexes si nécessaire au prix d'un temps de génération accru. Ces algorithmes de type MCMC ne permettent pas de créer directement des graphes aléatoires mais procèdent de la manière suivante :

1. générer un graphe simple respectant la distribution de degrés prescrite ;
2. le rendre connexe avec des permutations de liens si nécessaire ;
3. effectuer une suite de permutations de liens jusqu'à ce que le graphe semble suffisamment aléatoire. C'est un mélange (*shuffling*) du graphe.

L'étape 1 peut être réalisée avec l'algorithme de Havel-Hakimi [72] qui réalise exactement la distribution voulue de manière déterministe. Si la connexité du graphe n'est pas obligatoire, Viger et Latapy [153] ont validé empiriquement que $O(m)$ échanges de liens suffisent pour avoir un échantillonage uniforme, mais la preuve est toujours un problème ouvert. Milo et al. [106] ont estimé que le facteur constant de cette borne est environ 100. De plus ils décrivent une implémentation naïve avec connexité garantie dont la complexité est en $O(m^2)$. Cet algorithme naïf est appelé *switching algorithm*. Viger et Latapy [153] proposent une amélioration en $O(m \cdot \log(m))$ pour les graphes non dirigés en se basant sur des validations empiriques mais sans preuve formelle. L'utilisation de l'amélioration pour les graphes dirigés n'est pas étudiée.

La méthode a été généralisée par Tabourier et al. [145] afin de conserver des propriétés du graphe autres que la distribution des degrés, comme par exemple le nombre de triangles. La méthode peut nécessiter de faire des inversions de liens avec plus de 2 liens et de rejeter une inversion si elle change la propriété voulue.

Bien que notre proposition n'offre pas une meilleure complexité que MCMC et qu'elle ne génère pas que des graphes connexes, nous voyons deux avantages à notre méthode par rapport à MCMC. Tout d'abord, notre algorithme est beaucoup plus simple à implémenter et pourrait donc remplacer les *matching algorithms* dans les logiciels. Ensuite, il est beaucoup plus simple d'introduire des règles de connectivité spécifiques dans des algorithmes de génération directe qu'avec des inversions de liens. Avec ces dernières approches il faut

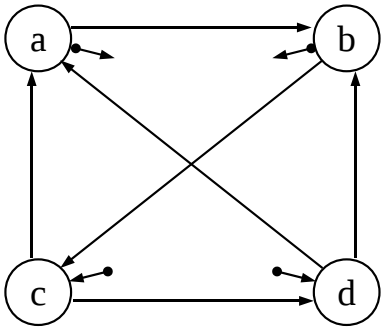


Figure 8.2 – Exemple de graphe durant le processus de génération.

en effet contraindre les échanges autorisés pour garantir que les graphes restent simples. Pour ces raisons nous pensons qu'il reste encore des verrous à lever pour la conception d'algorithmes de génération directe.

Échantillonnage séquentiel

Un algorithme d'échantillonnage séquentiel est proposé par Blitzstein et Diaconis [31]. Comme notre proposition, il génère un graphe séquentiellement en forçant un ordre pour une extrémité des nouveaux liens et en choisissant au hasard l'autre extrémité. En conséquence, il ne permet pas d'échantillonner uniformément. Notre méthode est une alternative viable pour deux raisons. Tout d'abord la méthode de Blitzstein et Diaconis est uniquement décrite et prouvée pour les graphes non dirigés, même si une adaptation semble possible. Ensuite, leur méthode nécessite de vérifier à chaque étape si la création du lien va permettre d'atteindre la distribution de degrés voulue, ce qui ralentit l'exécution à une complexité en $O(m \cdot n^2)$ et est moins efficace que les autres méthodes et en particulier, la nôtre.

Un algorithme d'échantillonnage séquentiel est proposé par Del Genio et al. [54] qui est plus rapide que le précédent : il est en $O(m \cdot n)$; par contre il ne permet pas d'avoir un échantillonage uniforme, mais la probabilité d'avoir une configuration donnée peut être calculée, ce qui permet de faire des moyennes pondérées en générant plusieurs graphes lorsque l'on s'intéresse à des propriétés particulières. L'algorithme est généralisé à des graphes dirigés par Kim et al. [81].

8.4 Priorité sur les nœuds les plus contraints

8.4.1 Principe

Notre objectif est de proposer un algorithme de génération de graphes qui crée les liens dans un ordre tel que le résultat soit nécessairement un graphe simple (i.e., sans boucle ou lien double) et qu'il soit possible d'ajouter des contraintes sur les liens. Cette propriété

peut être obtenue simplement en faisant en sorte que chaque extrémité choisisse une autre extrémité en préférant certaines extrémités à d'autres.

Afin de garantir que le graphe résultant soit un graphe simple, nous allons imposer un ordre sur les nœuds afin d'empêcher qu'il advienne une situation non soluble durant le processus de génération. La figure 8.2 montre un graphe durant le processus de génération avec quatre extrémités non connectées. Les quatre nœuds sont structurellement équivalents, il est donc nécessaire d'avoir une mesure plus fine que le nombre d'extrémités pour l'ordre de priorité.

8.4.2 Degré de liberté

La clé pour résoudre le problème de priorité est le nombre de nœuds potentiels auxquels les extrémités d'un nœud peuvent se connecter. Nous définissons donc pour chaque nœud v deux ensembles de *nœuds candidats* : les candidats cibles $tc(v)$ pour les extrémités sortantes de v et les candidats sources $sc(v)$ pour les extrémités entrantes de v, s'il reste des extrémités à connecter pour v.

$$tc(v) = \{w \in V | (w \neq v) \wedge (w \notin succ(v)) \wedge (ist(w) > 0)\} \tag{8.1}$$

$$sc(v) = \{w \in V | (w \neq v) \wedge (w \notin pre(v)) \wedge (ost(w) > 0)\} \tag{8.2}$$

Ayant ces listes de candidats à disposition, nous pouvons maintenant calculer pour chaque nœud v son degré de liberté par rapport aux nœuds candidats. Cela est fait indépendamment pour les liens sortant avec $odf(v)$ et pour les liens entrants avec $idf(v)$ comme suit [2] :

$$odf(v) = \begin{cases} |tc(v)| - ost(v) & \text{si } ost(v) > 0 \\ \infty & \text{sinon} \end{cases} \tag{8.3}$$

$$idf(v) = \begin{cases} |sc(v)| - ist(v) & \text{si } ist(v) > 0 \\ \infty & \text{sinon} \end{cases} \tag{8.4}$$

Avec ces deux valeurs on peut fixer l'ordre en choisissant les nœuds les plus contraints (*most constrained nodes* ou MCN), i.e., ceux ayant le plus faible degré de liberté dans les deux cas. Ceci est exprimé par $df(v)$:

$$df(v) = min(odf(v), idf(v)) \tag{8.5}$$

Le tableau 8.1 liste les degrés de liberté pour le graphe exemple de la figure 8.2 comme défini par nos formules. Avec ces valeurs il apparaît que soit le nœud b soit le nœud c doit être prioritaire pour choisir un nœud destination ou source respectivement. Cela amènera à la création du lien (b, d) ou (a, c) et évitera la création de liens qui empêcheraient la création d'un graphe simple. Ce principe est maintenant détaillé sous forme d'algorithme.

2. *odf* pour *outgoing degree of freedom* et *idf* pour *incoming degree of freedom*

	a	**b**	**c**	**d**
idf	∞	∞	0	1
odf	1	0	∞	∞
df	1	0	0	1

Tableau 8.1 – Degrés de liberté pour le graphe exemple de la figure 8.2

8.4.3 L'algorithme

Nous décrivons ici une implémentation naïve de notre algorithme. Nous utilisons des crochets pour distinguer les deux cas différents qui peuvent se produire dans la boucle. À chaque itération, seule la partie gauche ou droite s'applique selon le cas.

1. calculer les valeurs initiales `idf` et `odf` pour tous les sommets
2. répéter `m` fois :
   (a) choisir un nœud `mcn` de `df` minimum
   (b) distinguer [`odf(mcn) < idf(mcn)` / sinon]
   (c) créer une liste `candidates` = `[tc(mcn)/sc(mcn)]`
   (d) sélectionner `peer` au hasard dans `candidates`
   (e) créer le lien `mcn` [`->` / `<-`] `peer`
   (f) corriger `idf` de tous les nœuds si [`mcn` / `peer`] a utilisé sa dernière extrémité source
   (g) corriger `odf` de tous les nœuds si [`peer` / `mcn`] a utilisé sa dernière extrémité destination

Nous avons également implémenté une version légèrement optimisée de l'algorithme dans notre package open-source `SNA::Network`, qui est disponible sur la plateforme Perl : `http://www.cpan.org/`.

8.4.4 Complexité spatiale et temporelle

Comme nous utilisons une seule instance du graphe sans structure de données multidimensionnelles, la complexité pour stocker les données est linéaire, soit en :

$$C_s(n, m) = O(n + m).$$

L'étape 1 peut être effectuée en temps $O(n)$, en comptant tout d'abord toutes les sources et destinations possibles, puis en itérant sur tous les nœuds pour mettre à jour `idf` et `odf`. Les étapes (f) et (g) sont exécutées au plus n fois chacune indépendamment à partir de la boucle de l'étape 2, car chaque nœud peut utiliser sa dernière extrémité entrante ou sortante une seule fois. La correction des valeurs `idf` et `odf` requiert le même coût que l'étape 1 dans chaque cas, le coût total de ces deux étapes est donc $O(n^2)$. La boucle de

l'étape 2 est exécutée exactement m fois, et les étapes (b), (d) et (e) peuvent toutes être exécutées en $O(1)$. L'étape (a) requiert une recherche linéaire dans la liste des nœuds, soit $O(n)$. L'étape (c) nécessite une itération sur tous les nœuds, avec un coût en $O(n)$. Pour la boucle complète, le total des temps d'exécution est en $O(m \cdot n)$, et la complexité globale de l'algorithme complet appliqué sur un graphe sans sommet isolé est donc :

$$C_t(n, m) = O(n + n^2 + m \cdot n) = O(m \cdot n).$$

En utilisant un tas pour maintenir le minimum des `idf` et `odf`, en ne créant pas explicitement la liste `candidates` et en utilisant une liste pour se rappeler quels sont les nœuds pour lesquels `idf` et `odf` ont été modifiés, l'algorithme pourrait également être implémenté en temps

$$C_t(n, m) = O(m \cdot log(n)).$$

Il pourrait ainsi être utilisé pour générer des graphes ayant des milliards de liens.

8.4.5 Uniformité

En restreignant l'ordre aléatoire dans lequel les nœuds sont sélectionnés pour la création des connexions, nous brisons l'uniformité de la génération aléatoire. Cela pose surtout problème dans la phase initiale de l'algorithme quand les valeurs *df* sont très hétérogènes. Ces valeurs vont s'équilibrer durant l'exécution pour atteindre une sélection presque aléatoire.

Nous évaluons l'uniformité des échantillons avec les méthodes utilisées par Milo et al. [106]. La figure 8.3 montre les deux topologies possibles pour un graphe jouet consistant en un hub avec dix liens sortants, un hub avec dix liens entrants, et dix sommets intermédiaires ayant chacun un lien entrant et un lien sortant. Il y a une seule manière d'obtenir la topologie en (a) mais 90 manières différentes d'obtenir celle en (b). Un algorithme générant les graphes de manière uniforme devrait donc aboutir à chacune des 91 configurations dans 1.099% des cas.

Nous avons généré 1 000 000 de graphes avec cette distribution de degrés et compté combien de fois chaque configuration était obtenue. Restreindre la sélection aux deux hubs pour la création du premier lien introduit un biais mesurable dans la fréquence d'échantillonnage de la topologie (a), ce qui représente une sur-sélection d'environ 32%. Les 90 configurations de la topologie (b) sont quant à elles échantillonnées uniformément. La sur-sélection peut être expliquée par la priorité accordée aux deux hubs, ce qui ne donne pas aux nœuds internes la même chance de se connecter les uns avec les autres au début de l'algorithme, mais seulement plus tard quand ils atteignent les mêmes valeurs basses que les deux hubs.

L'algorithme de Milo et al. générait la configuration (a) dans environ 0.3% des cas [106], ce qui représente une sur-élection d'environ 70%. Mais, même pour cet algorithme, le biais mesurable pour cette distribution de degrés extrême ne conduit généralement pas à un fort biais pour des réseaux réels. Dans les tests que nous avons menés, nous n'avons pas observé de problèmes avec notre algorithme.

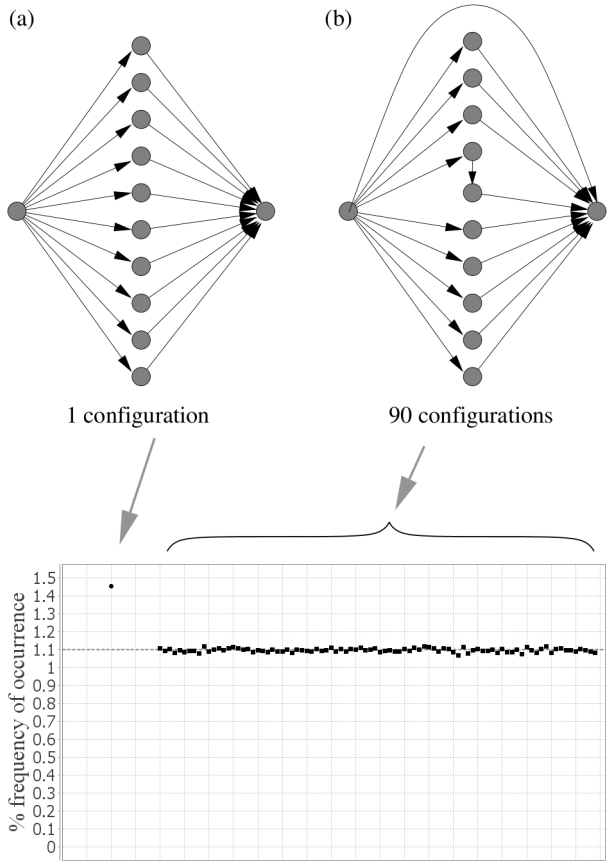


Figure 8.3 – Test d'uniformité de notre algorithme (à comparer avec les résultats de [106], p.3)

8.4.6 Connexité

Tout comme la méthode de Milo et al., notre algorithme ne garantit pas de connexité faible pour le graphe obtenu. Cependant, l'ensemble des configurations connexes est échantillonné uniformément quand on ignore les échantillons non connexes pour relancer l'algorithme.

Dans les situations où nous avons besoin d'un nombre spécifique d'échantillons connectés, il est possible de contourner ce problème. La probabilité pour un graphe aléatoire échantillonné uniformément peut être calculée exactement, comme montré par Molloy et Reed [109]. En ayant une telle estimation *a priori* sur le nombre d'itérations nécessaires, un tel ensemble de graphes connexes peut être généré en itérant l'échantillonnage jusqu'à ce que le nombre requis soit obtenu. Cependant, cela dépend fortement de la distribution de degrés.

8.4.7 Preuve de correction

Notre algorithme utilise une heuristique pour l'ordre de génération des liens qui garantit la réalisation de la distribution de degrés si elle est faisable. Cependant, nous n'avons pas de preuve formelle de cette assertion. Tous nos tests empiriques sur des graphes de terrain ont été couronnés de succès et nous n'avons pas réussi à construire un graphe qui, en obéissant à l'heuristique de priorité, a conduit à un état bloquant. De fait, tous ces liens problématiques, définis par Snijders [139], sont effectivement évités car les extrémités ont toujours des scores *df* relativement élevés au début de l'algorithme.

8.5 Sélection aléatoire non uniforme

La motivation initiale de ce travail était de pousser un peu plus loin les évaluations statistiques de graphes de terrain par le biais de graphes générés aléatoirement. Comme nous l'avons montré dans la section 8.3, toutes les méthodes de l'état de l'art pour la génération de graphes avec distribution de degrés fixée fonctionnent de manière non déterministe, avec des retours en arrière ou génèrent des liens indirects comme pour les méthodes à base de chaînes de Markov.

Ces méthodes ne permettent pas de rendre la sélection non uniforme de manière vraiment contrôlée. Avec une sélection directe de voisins, cela devient beaucoup plus facile car différents groupes de nœuds peuvent être pondérés différemment au moment de la sélection aléatoire, étape (2d) de notre algorithme (voir section 8.4.3).

Dans cette section, nous donnons un bref aperçu de l'utilisation et des bénéfices de cette idée sur deux problèmes réels.

8.5.1 Homophilie dans les réseaux de blogs politiques

Un premier exemple d'application vient d'une étude d'Adamic et Glance [9] qui ont analysé les connexions dans la blogosphère politique américaine en 2004. Ils ont agrégé un

réseau de blogrolls (une blogroll est une liste de blogs conseillés que l'on peut trouver sur un blog) comportant plus de 1000 blogs, conservateurs et libéraux. Ils ont observé que les deux groupes forment des communautés dans lesquelles chaque groupe recommande des blogs de sa propre classe dans environ 90% des cas.

Ce phénomène est appelé un *mixing pattern* en sociologie des réseaux [113], et homophilie (ou *assortative mixing*) dans les réseaux sociaux. La préférence dans la sélection des liens peut être dérivée de la *mixing matrix* et doit permettre de générer un graphe aléatoire dans lequel le coefficient d'homophilie est supposé être le même que dans le graphe original.

De cette façon, les propriétés du réseau de blogs politiques peuvent être comparées statistiquement aux réseaux aléatoires générés avec la même distribution de degrés et le même coefficient d'homophilie. Cela peut permettre de mettre en avant des différences structurelles qui auraient été cachées autrement. Cela peut montrer l'impact de l'homophilie dans ce contexte.

8.5.2 Structure communautaire

Un second exemple est lié à la génération de graphes aléatoires avec une structure communautaire prédéfinie. Une communauté est généralement définie comme un groupe de nœuds qui sont plus connectés entre eux que vers l'extérieur. La structure communautaire peut être vue comme une partition, une hiérarchie ou un recouvrement. Être capable de générer des graphes avec une structure communautaire tout en gardant d'autres propriétés classiques comme la distribution de degrés est nécessaire pour évaluer les algorithmes de détection de communautés.

Cependant, nous n'avons pu trouver qu'une seule méthode pour générer de tels réseaux avec une structure non recouvrante [89]. Cette méthode utilise le modèle configurationnel avec une contrainte supplémentaire pour connecter de préférence les nœuds d'une même communauté. La méthode a été étendue aux réseaux dirigés avec structure communautaire recouvrante dans [87]. Cependant dans ce modèle la majorité des nœuds doit appartenir à une seule communauté, et les autres doivent appartenir à un même nombre de communautés entré en paramètre. Bien que ces méthodes soient les seules pour générer des graphes avec une structure communautaire et une distribution de degrés fixée, elles sont limitées du fait qu'elles utilisent le modèle configurationnel comme brique de base. Elles conservent donc la distribution de degrés de manière approchée et l'addition de contraintes (pour obtenir des communautés) peut aggraver ce fait. Elles sont aussi limitées par les contraintes sur le nombre de communautés auxquelles un nœud peut appartenir et par le recouvrement des communautés, ce qui constitue un paramètre non modifiable.

Au contraire, notre algorithme ne souffre pas de ces défauts ; il peut générer toute combinaison de communautés (hiérarchiques et recouvrantes) qui peuvent être décrites par des groupes de sommets et des préférences de connexions entre eux.

8.6 Conclusion

Dans ce chapitre, nous avons présenté un algorithme efficace et très simple pour générer directement des graphes aléatoires respectant exactement une distribution de degrés fixée et ce de manière uniforme, ce qui était un problème ouvert : les algorithmes de matching avaient des problèmes pour obtenir exactement la distribution souhaitée, et les algorithmes à base d'inversion de liens opéraient indirectement ce qui rendait les changements de connexions difficiles à mettre en œuvre.

Pouvoir modifier ces modes de connexion de manière fine permet de comparer plus précisément les réseaux réels aux graphes aléatoires et peut permettre d'avancer dans leur compréhension. Contrairement aux modèles de graphes aléatoires, l'évaluation basée sur la génération de graphes aléatoires permet d'étudier directement toute propriété qui est calculable sur une instance d'un graphe. Ceci ouvre de nouvelles perspectives pour les utilisateurs de cette méthodologie.

L'algorithme est aussi utilisable pour la génération de réseaux avec une structure communautaire de tout type, dès lors qu'elle peut être décrite avec des connexions préférentielles entre groupes. Ceci ouvre des perspectives de nouvelles applications pour la création de benchmarks.

Deux perspectives sont ouvertes, en plus de la preuve de correction de l'algorithme. Bien que plusieurs travaux connexes indiquent que la non uniformité n'est pas un problème pour des distributions de degrés réelles, une évaluation précise telle que celle effectuée par Milo et al. [106] doit être effectuée. Concernant l'efficacité, des améliorations sont encore possible en exploitant la monotonie et la stabilité relative des valeurs df avec des structures de données plus sophistiquées que des tableaux simples. La contrepartie serait une implémentation plus complexe.

Chapitre 9

Analyse de graphes bipartis pour détecter l'activité pédophile dans les réseaux Pair à Pair (P2P)

9.1 Introduction

Depuis une quinzaine d'années, les réseaux pair-à-pair (P2P) permettent à des millions d'utilisateurs d'échanger facilement des contenus entre des points très éloignés du monde. À l'aide d'un logiciel dédié, chaque membre du réseau peut partager des fichiers stockés sur sa machine et en obtenir d'autres en effectuant une recherche par mots-clefs.

La pédopornographie – c'est-à-dire la représentation d'actes sexuels impliquant des enfants – est un problème d'importance cruciale pour la société. Il est essentiel de faire cesser la commission d'abus sexuels sur des enfants visant à produire de tels contenus, en anéantissant les réseaux de criminels qui participent à la production et à la diffusion de ces contenus. Récemment, une méthodologie d'étude de la pédopornographie dans les réseaux P2P a été développée, aboutissant à la constitution de jeux de données de très grande taille et au développement d'un filtre sémantique aux performances évaluées par des experts du domaine [92]. Les études préliminaires menées par les auteurs ont permis d'observer des estimations statistiques fiables sur la quantité de requêtes liées à la pédopornographie dans des réseaux P2P, lesquelles ont été corroborées par d'autres équipes [129].

Les réseaux P2P, de grande taille, peuvent également être vus comme des réseaux sociaux, dans lesquels les utilisateurs forment des communautés d'intérêt autour de thématiques recherchées et de fichiers partagés. Ainsi, les travaux récents en analyse de graphes offrent un formalisme et des méthodes que nous nous proposons d'utiliser pour interroger les relations entre les catégorisations sémantiques d'un filtre de requêtes et la topologie du graphe d'utilisateurs sous-jacent. En particulier, nous examinons les deux questions suivantes :

— dans quelle mesure une analyse centrée sur les utilisateurs permet-elle de découvrir de nouveaux mots-clefs liés à une thématique donnée ?

— les catégorisations sémantiques du filtre peuvent-elles être retrouvées à partir de la seule analyse topologique d'un graphe ?

Dans les deux cas, on espère que l'on pourra d'une part corroborer des résultats obtenus de manière indépendante – des mots-clefs permettant de détecter des requêtes pédophiles et les catégorisations associées – et d'autre part proposer des pistes d'améliorations du filtre. En particulier, dans le cas de la thématique pédophile, mettre à jour un tel filtre est nécessaire, les combinaisons de mots-clefs évoluant sensiblement au fil du temps. Pourtant, l'expertise nécessaire pour cette mise à jour est rare et peu accessible, y compris à un niveau mondial : on peut dès lors espérer que des mesures topologiques adéquates permettent de limiter significativement le recours à des experts humains.

9.2 Méthodologie

Les données que nous utilisons sont des requêtes par mots-clefs envoyées par des utilisateurs au moteur de recherche d'edonkey. La collecte des données a eu lieu pendant 10 semaines en continu sur l'un des serveurs les plus importants du réseau edonkey [11]. Le jeu de données est constitué de 107 226 021 requêtes, provenant de 23 892 531 adresses IP différentes, anonymisées à la volée. Chaque requête est horodatée et comporte un port de communication ainsi que la liste des mots-clefs de la recherche.

Dans [92], les auteurs ont présenté leur méthodologie pour détecter les requêtes effectuées dans l'intention d'obtenir des contenus à caractère pédopornographique (appelées dans la suite *requêtes pédophiles*). Nous utilisons leur filtre de détection de requêtes dont le schéma de la figure 9.1a illustre le fonctionnement. Celui-ci identifie quatre catégories de requêtes pédophiles et repose sur la constitution de groupes de mots-clefs et de leurs traductions dans différentes langues ; les combinaisons de ces mots-clés permettent de décider si une requête est pédophile ou non (cf. figure 9.1a). Avec les données utilisées, le filtre identifie 207 340 requêtes comme pédophiles et 151 545 requêtes sont classées en catégorie 1, 27 753 en catégorie 2, 35 264 en catégorie 3 et 4 299 en catégorie 4 (une requête peut appartenir à plusieurs catégories).

Dans notre contexte d'étude, où l'on ne dispose que d'une adresse IP et d'un port de communication, il est difficile de passer de la granularité des requêtes à celle des utilisateurs. Différents mécanismes sont en effet susceptibles d'agir sur l'adresse IP des utilisateurs, ce qui nous conduirait à mélanger les requêtes d'utilisateurs distincts (typiquement, parce qu'ils disposent d'une adresse IP publique commune) ou à considérer comme provenant de deux utilisateurs les requêtes d'un même utilisateur (par exemple, parce qu'une allocation dynamique d'adresse lui propose deux adresses IP distinctes à des moments différents). Dans la suite de ce travail, on associe cependant une adresse IP à un utilisateur unique.

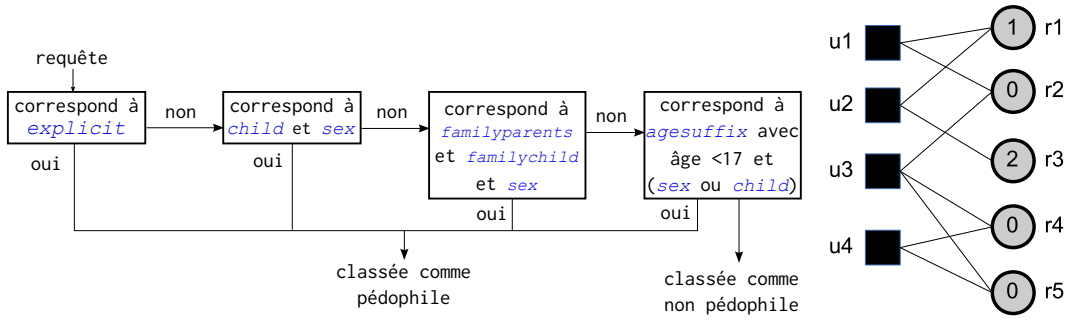
Nous proposons d'améliorer les résultats du filtre en effectuant des mesures sur le graphe biparti comportant d'une part les requêtes effectuées, de l'autre les utilisateurs, identifiés par un couple (*adresse IP*, *port*), comme l'illustre la figure 9.1b. En utilisant la classification existante des requêtes, nous définissons une similarité entre celles-ci. On cherche à obtenir que les requêtes classées non pédophiles auparavant, mais qui auraient dû l'être (des *faux*

négatifs) aient un score élevé. Et, réciproquement, on espère que les *faux positifs* du filtre obtiennent un score faible.

Soit U l'ensemble des utilisateurs et R celui des requêtes. On se donne la fonction f qui, à un couple $(u, r) \in U \times R$, associe le nombre de fois que l'utilisateur u a saisi la requête r. Pour chaque requête r, nous définissons son voisinage $V(r)$ tel que : $V = \{u \in U | f(u, r) \geq 1\}$. Pour chaque utilisateur, notons $R(u)$ l'ensemble de ses requêtes. Pour une classe C de requêtes, le score $s_C(r)$ est défini par l'expression 9.1 :

$$s_C(r) = \frac{\sum_{u \in V(r)} |C \cap R(u) \setminus \{r\}|}{\sum_{u \in V(r)} |R(u) \setminus \{r\}|} \quad (9.1) \qquad s_C(r) = \sum_{u \in V(r)} \frac{|\{C \cap R(u) \setminus \{r\}|}{|\{R(u) \setminus \{r\}|} \quad (9.2)$$

Les utilisateurs n'ayant effectué qu'une requête ne contribuent pas à relier des requêtes entre elles, ils sont écartés. La nature de la requête considérée n'intervient pas dans le calcul de son score. L'expression 9.1 est sensible au biais introduit par les déséquilibres entre les nombres de requêtes des utilisateurs. Nous avons aussi essayé le score 9.2 qui normalise les contributions de chaque utilisateur, quel que soit le nombre de requêtes, mais les résultats obtenus ne sont pas significativement différents de ceux fournis par 9.1.



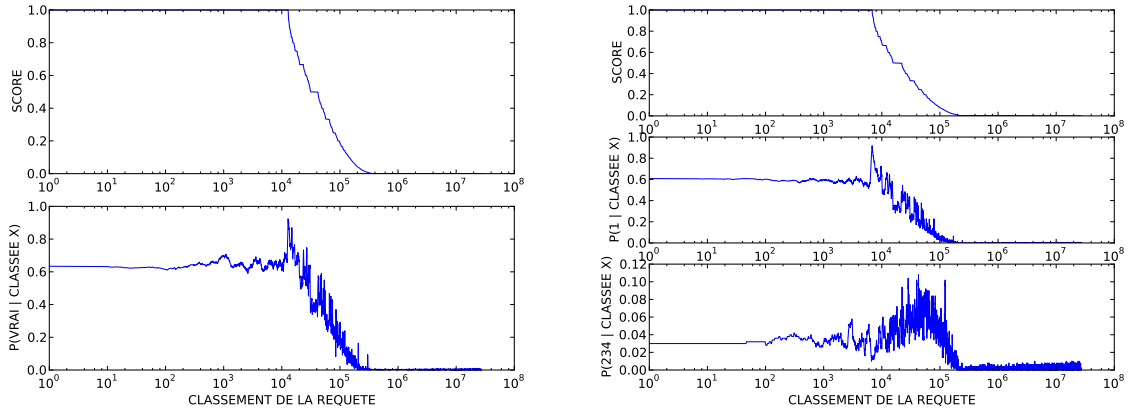
(a) (b)

Figure 9.1 – (9.1a) : L'outil de détection présenté dans [92] (en bleu et italique, les noms des groupes de mots-clefs). Dans la suite, nous utiliserons, de gauche à droite, les numéros 1, 2, 3 et 4 pour désigner ces catégories, et 0 pour les requêtes non pédophiles. (9.1b) : le graphe biparti *utilisateurs-requêtes*. Les utilisateurs, représentés par des carrés, sont liés aux requêtes qu'ils ont effectuées. Les numéros dans chaque cercle indiquent la catégorie des requêtes. Exemple : l'utilisateur *u1* a effectué une requête pédophile r_1 (de catégorie 1) et une requête non pédophile, r_2.

9.3 Résultats et validation

9.3.1 Étude des faux positifs et faux négatifs

La figure 9.2a (haut) présente les scores obtenus par les requêtes du jeu de données, pour la classe C incluant toutes les requêtes pédophiles, ordonnées par score. Pour la figure 9.2a



(a) (b)

Figure 9.2 – 9.2a : score obtenu à partir de l'ensemble C de toutes les requêtes étiquetées 1, 2, 3 ou 4 (haut) et probabilité pour la requête d'être étiquetée pédophile en fonction du classement par score (bas). 9.2b : score obtenu à partir de l'ensemble C des requêtes de la catégorie 1 (haut), probabilité pour la requête d'être étiquetée 1 (milieu) et probabilité d'être étiquetée 2, 3 ou 4 en fonction du classement par score (bas).

(bas), les requêtes sont groupées en paquets glissants de k [1] : pour une requête classée en X$^{\text{ème}}$ position, la valeur en ordonnée correspond à la proportion de requêtes étiquetées comme pédophiles par le filtre sémantique entre la $X^{\text{ème}}$ et la $X + k^{\text{ème}}$ requête.

On obtient 12 858 requêtes de score 1, c'est-à-dire provenant d'utilisateurs n'ayant fait, hormis la requête considérée, que des requêtes pédophiles. Parmi celles-ci, on constate que 4 518 (soit 35,1%) sont des requêtes que le filtre sémantique a étiquetées comme non pédophiles. Ce sont pour la plupart des requêtes très proches de la thématique pédophile, comportant soit des nouveaux mots-clefs spécifiques, soit des combinaisons de mots-clefs non prises en compte mais pertinentes. Bien sûr, avant d'intégrer ces mots-clefs directement dans le filtre, une étude générale des requêtes qui leur sont associées sera nécessaire, afin de ne pas rajouter de faux positifs. En revanche, il semble que les résultats pour les faux positifs soient moins concluants : les requêtes étiquetées comme pédophiles avec un score de 0, pourtant effectuées par des utilisateurs n'ayant fait *aucune* autre requête pédophile, ne semblent pas être en grande partie des *faux positifs*.

9.3.2 Pertinence de la segmentation existante

Le filtre initial définit 4 catégories de requêtes et nous examinons ici, avec notre score, si celles-ci paraissent pertinentes au regard de la topologie. En particulier, on regarde si la catégorie 1, composée par des requêtes comportant des mots-clefs très spécifiques (et

1. Nous avons pris $k = 500$ pour éviter des oscillations trop importantes.

supposés peu connus du grand public) peut se retrouver à partir des requêtes des catégories 2, 3 et 4, qui elles sont assez proches sémantiquement.

La figure 9.2b présente notre expérimentation : on calcule cette fois le score à partir de la seule classe des requêtes étiquetées 1 (9.2b-haut). On retrouve massivement les requêtes de type 1 en haut du classement (9.2b-milieu), avant de voir que les requêtes de catégorie 2, 3 et 4 ont des scores faibles (9.2b-bas). Cette dernière courbe laisse penser que la catégorie 1 est peu liée aux autres : les utilisateurs saisissant des requêtes de type 1 ont peu tendance à proposer des requêtes de type 2, 3 ou 4. Il semble donc que les différents types de requêtes pédophiles sémantiques sont aussi différents au niveau topologique.

Pour une analyse plus fine, on compare pour chaque catégorie, le nombre de paires de requêtes effectuées par un même pair pour voir si les utilisateurs ont tendance à rester cantonnés aux requêtes d'une seule catégorie.

i	j	V_{ij}	$E(V_{ij})$	$V_{ij}/E(V_{ij})$
1	1	205,986	177,595.62	1.15
1	2	21,296	35,266.72	0.60
1	3	33,239	73,718.54	0.45
1	4	1,905	4,235.47	0.44
2	2	8,525	1,752.94	4.86
2	3	7,432	7,315.32	1.01
2	4	731	421.04	1.73
3	3	27,995	7,649.96	3.65
3	4	552	879.19	0.62
4	4	1,199	25.14	47.69

Tableau 9.1 – Statistique des paires de requêtes effectuées par le même pair et comparaison à un *modèle nul* ($E(V_{ij})$).

Le tableau 9.1 présente nos résultats : V_{ij} représente le nombre de paires de requêtes de type i et j effectuées par le même pair. $E(V_{ij})$ représente la valeur attendue de V_{ij} en supposant qu'une requête de type i est en moyenne dans $\sum_{k=1}^{4} V_{ik}$ paire de requêtes. Le ratio des deux précédentes valeurs est plus grand que 1 pour $i = j$ et plus petit pour $i \neq j$. Ainsi la comparaison à ce *modèle nul*[2] montre que deux requêtes de même type ont bien plus de chance d'avoir été effectuées par le même pair que deux requêtes de type différent. La catégorisation au niveau sémantique se retrouvant donc aussi au niveau topologique, elle est donc pertinente.

9.4 Conclusion et perspectives

Nous avons montré que l'étude de la topologie du graphe biparti utilisateurs-requêtes d'un réseaux P2P permet une amélioration et une réactualisation du filtre sémantique de

2. En anglais : *null model*.

[92] pour détecter les requêtes pédophiles. L'étude peut en effet permettre – par ajout et suppression de mots-clefs et par une meilleure combinaison de ces derniers – de diminuer le nombre de faux négatifs, mais également, dans une moindre mesure, de faux positifs. Nous avons également corroboré les résultats de ces auteurs en mettant en évidence quatre types de requêtes topologiques correspondant aux quatre catégories sémantiques.

Une perspective à court terme de notre travail est de permettre la mise à jour du filtre sémantique, sous réserve de disposer d'une évaluation adaptée des performances. Dans un deuxième temps, il est pertinent d'envisager d'appliquer d'autres méthodes de clustering pour segmenter plus finement les requêtes et les utilisateurs pédophiles : éventuellement ajouter des sous-catégories aux catégories existantes, ou en trouver de nouvelles. Enfin, la méthodologie de notre étude, reposant sur l'élaboration d'un premier filtre sémantique *ad hoc* puis son amélioration à l'aide d'une étude topologique, est généralisable et il est donc envisageable de l'utiliser dans d'autres contextes.

Chapitre 10

Conclusion

Dans cette thèse nous avons proposé une approche pour identifier des communautés. Elle constitue une alternative à l'approche classique consistant en l'optimisation gloutonne d'une fonction de qualité : elle est basée sur la notion de proximité entre les nœuds et plus précisément sur la détection de discontinuités dans la décroissance des valeurs de cette proximité. Nous avons montré comment décliner notre approche pour résoudre efficacement trois problèmes différents : "trouver toutes les communautés auxquelles un nœud donné appartient", "compléter un ensemble de nœuds en une communauté" et "trouver des communautés recouvrantes dans un réseau".

Nous avons appliqué la détection de communautés à deux problèmes concrets et cela a donné lieu à deux travaux d'ouverture : (i) en essayant d'améliorer un filtre sémantique de détection de requêtes pédophiles sur des réseaux pair à pair, nous avons commencé une adaptation de la méthode pour travailler avec des graphes bipartis, (ii) après avoir postulé l'existence d'une communauté (au sens "groupe de nœuds fortement liés les uns aux autres et peu liés vers l'extérieur") de capitalistes sociaux sur Twitter, nous avons tenté de la détecter sans grand succès. Nous avons alors proposé une méthode pour les détecter en utilisant des attributs en plus de la topologie du réseau. Nous avons également proposé une méthode permettant de mettre à jour les outils de mesure d'influence sur Twitter que les capitalistes sociaux arrivent à tromper.

Un autre travail d'ouverture est né de notre initiative visant à combler le manque d'algorithmes pour générer des réseaux réalistes ayant des communautés recouvrantes : nous avons proposé une méthode de génération de graphe respectant une distribution de degrés fixée. Notre méthode, contrairement aux autres méthodes existantes, est directe et exacte. Elle constitue ainsi une piste très intéressante vers la création d'un tel algorithme. Elle pourrait permettre de générer des réseaux avec davantage de contraintes (correspondant à la structure communautaire) de façon aléatoire et "plus uniforme" que les autres méthodes.

Nous allons maintenant présenter plus en détail les différentes contributions de la thèse, puis les perspectives.

10.1 Contributions

Cette thèse a donné lieu à deux articles publiés dans des journaux internationaux [50, 52], quatre articles publiés dans des conférences internationales [51, 45, 118, 65], un article publié dans un atelier international [48], un chapitre de livre international [41], un résumé étendu publié dans une conférence internationale [124], quatre articles publiés dans des conférences nationales [46, 146, 49, 53] et un article dans un atelier national [47].
Les contributions de la thèse peuvent être détaillées comme suit, partie I :

- Un état de l'art sur les communautés ego-centrées et les communautés multi-ego-centrées (chapitre 2).
- L'idée d'utiliser une approche à base de mesure de proximité pour détecter une communauté ego-centrée en s'affranchissant d'une fonction de qualité (chapitre 3 et article [52]).
- La conception de deux nouvelles mesures de proximité, l'une sans paramètre appelée Opinion propagée et l'autre paramétrée appelée Katz+(chapitre 3 et articles [52, 51, 49]).
- La possibilité de savoir grossièrement à combien de communautés appartient un nœud d'intérêt grâce à la forme de la courbe de décroissance des proximités à ce nœud (chapitre 3 et articles [52, 50]).

Ainsi que, partie II :

- L'observation que, en général, un nœud appartient à plusieurs communautés et ne peut donc pas caractériser une communauté à lui seul. Cependant, quelques nœuds (voire deux nœuds) bien choisis peuvent caractériser une communauté (chapitre 4 et les articles [52, 50]).
- L'idée de trouver toutes les communautés d'un nœud d'intérêt en cherchant les communautés bi-ego-centrées sur ce nœud et des nœuds candidats qui ne sont ni trop loin ni trop près de manière à ce qu'ils ne partagent qu'une seule communauté avec le nœud d'intérêt (chapitre 4 et les articles [52, 50]).
- La nécessité d'utiliser des paramètres pour les mesures de proximité afin de prendre en compte les deux compromis que nous avons jugés nécessaires : (i) distance/redondance et (ii) popularité/intimité. (chapitre 3 et articles [51, 49]).
- L'idée que, dans le cas de la complétion de communauté, on peut utiliser les nœuds qui sont initialement dans la communauté afin de calibrer les paramètres de la mesure de proximité (chapitre 5 et articles [51, 49]).
- La découverte que dans certains réseaux, il existe pour la plupart des communautés au moins un nœud qui peut caractériser la communauté à lui seul (un représentant de communauté), c'est-à-dire un nœud qui est seulement dans la communauté en question et est important en son sein (chapitre 6). Ceci n'est pas en contradiction avec le premier point : on montre que, en général, un nœud appartient à plusieurs communautés, mais que pour chaque communauté, il existe un nœud qui appartient à cette communauté seulement (le nombre de communautés est en général bien plus petit que le nombre de nœuds).

Les travaux d'ouverture à la thèse ont mené aux contributions suivantes, partie III :

— La découverte que, sur Twitter, des utilisateurs malveillants (ou inutiles), appelés *capitalistes sociaux* arrivent à tromper les outils classiques pour mesurer l'influence (Klout, Kred et Twittaliser) et à être reconnus comme influents. La conception d'un algorithme pour détecter ces capitalistes sociaux et la mise au point d'un score reflétant plus justement leur influence. Ainsi que la mise en place d'une application en-ligne : `http://www.bit.ly/DDPapp` (chapitre 7 et articles [45, 46]).
— Un nouveau benchmark plus efficace pour générer des graphes avec une distribution de degrés donnée dans l'optique de générer des graphes avec communautés plus réalistes (chapitre 8 et article [118]).
— L'utilisation du réseau biparti requêtes/utilisateurs extrait des requêtes faites sur un réseau P2P afin d'améliorer la détection sémantique des requêtes à caractère pédophile (chapitre 9 et article [65]).

Les travaux auxquels nous avons participé, mais non détaillés dans la thèse consistent en :

— L'étude de grandeurs d'intérêt afin de quantifier le recouvrement dans les réseaux bipartis, en particulier les réseaux bipartis de type communautés/nœuds ainsi que la proposition de deux nouvelles grandeurs simples : *le monopole* et *la dispersion* pour quantifier ce recouvrement (article [146]).
— Le développement d'une application Twitter pour suggérer des utilisateurs à mentionner dans les tweets afin d'accroître le nombre de retweets (résumé étendu [124]). L'application est essentiellement basée sur des expériences faites en temps réel.

10.2 Perspectives

10.2.1 Retour sur la notion de communauté

Étude des liens inter-communautaires et du recouvrement entre communautés

La définition informelle de ce qu'est topologiquement une communauté, "un groupe de nœuds fortement liés les uns aux autres et peu liés vers l'extérieur", a du sens lorsqu'il s'agit de communautés vues comme une partition du réseau, mais elle perd un peu son essence lorsque l'on considère des communautés recouvrantes. En effet, si chacun des nœuds d'une communauté d'intérêt appartient également à d'autres communautés, comment la communauté d'intérêt pourrait-elle avoir peu de liens vers l'extérieur ?

Il est urgent de rompre avec cette définition des communautés et d'en proposer une plus juste. Un premier pas pourrait être de garder la première partie de la définition "groupe de nœuds fortement liés les uns aux autres", mais de modifier la deuxième "peu liés vers l'extérieur". Les nœuds dans les communautés "vérité de terrain" ne sont en général pas "peu liés vers l'extérieur", au contraire ils ont beaucoup de liens vers l'extérieur. Cependant, localement, ces liens ont tendance à s'éparpiller aléatoirement et indépendamment dans le réseau : ils vont vers d'autres communautés qui sont différentes ; sans pour autant être complètement indépendant : il existe souvent des petits paquets de nœuds à l'intersection des communautés.

Des travaux ont déjà été réalisés dans ce sens comme dans [66] où la fonction de qualité appelée cohésion et basée sur les triangles (déjà discutée dans le chapitre 2) autorise les communautés à avoir des liens sortants du moment qu'ils sont aléatoires. Un autre exemple de fonction de qualité faisant de même est la permanence proposée dans [39] (bien que définie pour une structure communautaire en partition) : les communautés se concurrencent pour gagner des nœuds en tirant dessus (au travers des liens), une communauté qui ne tire pas fort un nœud (par l'intermédiaire de peu de liens) n'aura que peu d'impact.

Cependant ces fonctions de qualité sont basées sur une intuition physique de ce qu'est une communauté, puis elles sont validées sur des communautés vérité de terrain. Je propose de faire l'inverse : je propose d'étudier des communautés vérité de terrain et en particulier de caractériser leurs liens inter-communautaires et leurs parties recouvrantes afin d'en tirer des principes généraux bien plus fins que ceux de la définition actuelle d'une communauté.

Cette perspective permettra de proposer une meilleure définition de ce qu'est une communauté et possiblement de caractériser différents types de communautés et différents types de réseaux (en fonction de leur structure communautaire).

Une communauté comme groupe pertinent de nœuds

Dans le même ordre d'idée, si l'on définit une communauté comme "un groupe de nœuds qu'il est pertinent de rassembler", alors une communauté devient plus large qu'un simple "groupe de nœuds fortement liés les uns aux autres et peu liés vers l'extérieur". En effet, un groupe de nœuds peut être une communauté, non seulement parce que ses nœuds sont fortement liés les uns aux autres et peu liés vers l'extérieur, mais plus généralement parce que ses nœuds ont un certain profil de connexion : parce qu'il y a "quelque chose" dans leur profil de connexion qui fait qu'il est pertinent de les rassembler.

Je propose ainsi d'essayer de capturer ce "quelque chose" par l'étude rigoureuse de communautés vérité de terrain et de mettre en place un modèle de structure du réseau plus général et plus complet qu'une simple structure communautaire telle qu'elle est définie à l'heure actuelle.

Remarquons qu'une fois ce modèle mis en place, il sera possible de trouver ces groupes dans un réseau pour lequel la vérité de terrain n'est pas fournie, en optimisant une fonction de qualité, ou plutôt des fonctions de qualité, si différentes classes de communautés sont mises en évidence. Ou bien, de la même façon que nous l'avons exposé au cours de ce manuscrit, de les détecter en utilisant une approche à base de mesures de proximité, ou plutôt de mesures de similarité, puisqu'il ne s'agirait plus de l'évaluation de la proximité topologique des nœuds, mais plutôt d'une similarité au regard de certains critères.

Recours à la visualisation

Au cours de cette thèse, nous avons montré à quel point les mesures de proximité et la détection de communautés sont liées. En effet, nous avons vu qu'il est possible d'isoler une partie locale du réseau proche d'un ou de plusieurs nœuds grâce à une mesure de proximité. Ces parties locales sont petites (par rapport à la taille du réseau complet) et présentent

une structure consistant en un petit nombre de communautés recouvrantes. En termes de visualisation, les applications des mesures de proximité sont très importantes, puisque ces parties locales que les mesures de proximité permettent d'isoler ont une structure plus discernable que celle du réseau complet. En effet, le dessin d'un grand réseau mène souvent à une représentation en forme de pelote de laine dont on ne peut extraire aucune connaissance (même si des techniques de coarse-graining permettent parfois de voir une certaine structure du réseau en agrégeant les nœuds en communautés). Au contraire, le dessin de parties locales isolées à l'aide de mesures de proximité pourrait être très informatif.

Je propose ainsi d'étudier comment utiliser des mesures de proximité à bon escient afin d'isoler des parties locales d'un réseau pour les visualiser et non pas directement pour y détecter des communautés. Ceci pourra être suivi de l'incorporation de ces méthodes dans des plateformes comme Tulip [22] ou Gephi [27]. Je pense que la visualisation de ces structures locales pourrait permettre de mieux comprendre la structure des communautés elles-mêmes et ainsi être d'une grande utilité pour progresser dans les deux perspectives précédentes.

10.2.2 Calcul effectif de ces communautés

Retour aux fonctions de qualité

L'approche classique pour détecter des communautés consistant en l'optimisation gloutonne d'une fonction de qualité peut être inefficace dans certains cas. En effet, à cause de minima locaux, l'optimisation peut donner de mauvais résultats. De plus, la méthode peut également être trompeuse, car elle donne chaque fois une communauté même s'il n'y en a pas. Au contraire, notre méthode ne souffre pas de ces défauts. Par contre, une comparaison directe des deux méthodes pour la détection d'une communauté contenant un nœud représentant sur le réseau *Wikipédia 2012* (en comparant les communautés obtenues à la vérité de terrain) montre que les deux méthodes se valent, cf. tableau 10.1. La meilleure valeur obtenue par l'optimisation de fonction de qualité est même presque chaque fois meilleure que celle obtenue par notre approche.

L'optimisation de fonction de qualité est effectuée de la façon suivante :

— **initialisation :** on part de la communauté contenant seulement le nœud d'intérêt,
— **optimisation :** chaque nœud à l'extérieur de la communauté ayant au moins un voisin dans la communauté est ajouté à la communauté s'il augmente sa qualité (mesurée par le ratio liens internes/liens externes) ; chaque nœud dans la communauté est enlevé si cela augmente la qualité de la communauté sauf pour le nœud d'intérêt qui ne peut pas être enlevé.
— **arrêt :** on s'arrête lorsque l'on ne peut plus augmenter la fonction de qualité par l'ajout ou la suppression d'un nœud.

Je propose ainsi de combiner les deux méthodes afin d'additionner les forces de chacune. En particulier, appliquer d'abord la méthode à base de mesure de proximité afin d'être sûr d'éviter les minima locaux évidents, de vérifier qu'une communauté est bien présente et d'en avoir une première approximation. Ensuite, utiliser l'optimisation de fonction de qualité

Catégorie	candidat représentant	meilleur score F_1 conductance	score F_1 moyen conductance	score F_1 proximité
Chess (3850 nœuds)	Chess	0.919	0.870	0.89
	Magnus Carlsen	0.919	0.917	0.90
	Queen's Gambit	0.888	0.284	0.91
	World Chess Championship	0.918	0.840	0.90
	Chess Boxing	0.919	0.045	0.00
Sumo (445 nœuds)	Sumo	0.909	0.763	0.88
	Taiho Koki	0.906	0.895	0.91
	Yokozuna	0.908	0.102	0.88
	Lyoto Machida	0.000	0.000	0.08
Boxing (7289 nœuds)	Boxing	0.842	0.815	0.79
	Cruiserweight	0.829	0.322	0.78
	Vitali Klitschko	0.833	0.370	0.02
	Chess Boxing	0.818	0.020	0.01

Tableau 10.1 – Résultats obtenus sur le réseau *Wikipédia 2012*.

afin d'affiner le résultat semble être une perspective intéressante.

Équation de la chaleur avec conditions aux limites

Je pense qu'il serait aussi intéressant d'étudier des communautés multi-ego-centrées pondérées (possiblement avec des poids négatifs). Dans ce cas, on cherche une communauté sachant qu'un petit ensemble de nœuds donnés lui appartient avec un certain poids. Nous avons déjà bien avancé sur cette piste en mettant au point une solution inspirée de la physique : le réseau est vu comme une structure métallique où les nœuds sont des boules de pétanque et les liens sont des barres de fer liant les nœuds (plus ou moins grosses selon la pondération du lien), le système étant plongé dans le vide et les rayonnements négligés. La température de certains nœuds est fixée à une certaine valeur : 100 pour un nœud que l'on sait dans la communauté et 0 pour un nœud que l'on sait hors de la communauté (toute valeur entre 0 et 100 étant également possible ; par exemple, si l'on veut qu'un nœud soit à la frontière de la communauté, on le fixe à 50). La température d'un nœud reflète donc son degré d'appartenance à la communauté. On résout l'équation de la chaleur avec ces conditions aux limites. Cela nous donne la température de tous les nœuds dans le réseau et donc la proportion selon laquelle ils font partie de la communauté définie par les nœuds à température fixée. Notre code pour résoudre le système utilise la méthode du gradient conjugué sur la matrice définie positive correspondant au laplacien avec les lignes et les colonnes des nœuds à température fixée enlevées. Il peut traiter, en quelques secondes, des graphes ayant des centaines de millions de liens.

Le "locality sensitive hashing" pour la détection de communautés

Comme détaillé en conclusion du chapitre 6, nous avons pensé à appliquer un algorithme de *Locality sensitive hashing* (LSH) [93] après l'obtention d'un grand ensemble de communautés dont certaines sont très similaires (ce sont les mêmes à un peu de bruit près). L'idée principale du LSH est d'utiliser une famille de fonctions de hachage choisies telles que des points proches dans l'espace d'origine aient une forte probabilité d'avoir la même valeur de hachage. Ainsi, des ensembles de communautés similaires pourront être détectés plus rapidement (car elles ont la même valeur de hachage) et le nettoyage gagnerait en vitesse par rapport à notre heuristique courante (consistant à comparer toutes les communautés partageant au moins un nœud). Nous travaillons d'ailleurs actuellement sur cette méthode en utilisant la similarité de Jaccard pour comparer deux communautés et des fonctions de hachage associées à cette similarité : *MinHash* (ou *min-wise independent permutations locality sensitive hashing scheme*) [35].

J'envisage à présent d'appliquer directement le MinHash sur chaque nœud identifié par l'ensemble de ses voisins. Ainsi, des nœuds ayant un profil de connexion similaire (dont les ensembles de voisins ont une similarité de Jaccard élevée) seraient identifiés très rapidement (en temps quasiment linéaire). Les ensembles de nœuds identifiés ne seraient pas des communautés au sens de "groupe de nœuds fortement liés les uns aux autres et peu liés vers l'extérieur", mais des ensembles de nœuds ayant tendance à se connecter aux mêmes nœuds.

Ce type de communautés se rapprocherait du Stochastic Block Model [74] qui peut être vu comme une généralisation des communautés : les nœuds sont partitionnés de telle sorte que les nœuds d'un même groupe aient des profils de connexion similaires. Cela se rapprocherait aussi d'un clustering des nœuds identifiés par le vecteur correspondant à leur profil de connexion. Les deux méthodes ayant souvent un problème pour passer à l'échelle, l'approche que je propose a clairement un intérêt potentiel.

10.2.3 Vers des méthodes pour la dynamique

Suivi temporel de communautés dans les réseaux dynamiques

Pour des réseaux dynamiques (où des nœuds et des liens apparaissent et disparaissent), la structure du réseau ainsi que celle de ses communautés changent au cours du temps. Des communautés apparaissent, disparaissent, fusionnent et se séparent, comme illustré par la figure 10.1.

Nous montrons ici que dans le cadre de la détection de communautés et de leur suivi temporel dans les réseaux dynamiques, la méthode de détection par l'approche mesure de proximité présente deux atouts par rapport à l'approche fonction de qualité :

1. la méthode est déterministe et peu sensible aux perturbations, comme montré sur la figure 10.2 : même en appliquant de grosses perturbations au réseau, une communauté bien dessinée est encore apparente.
2. on peut exploiter le fait qu'une communauté peut être caractérisée par un nœud

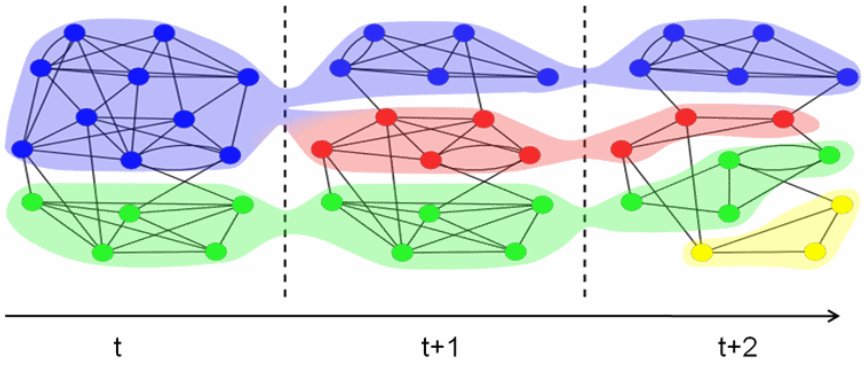


Figure 10.1 – Suivi temporel de communautés

(ou un petit ensemble de nœuds) comme montré sur les figures 10.3 et 10.4 où l'on détecte dans le réseau *Wikipédia 2008*, puis dans le réseau *Wikipédia 2012* la communauté "Chess" (catégorie agrégée, comme détaillé en annexe) en partant du nœud "Magnus Carlsen".

Les autres méthodes de détection de communautés étant souvent non-déterministes et peu stables : lorsque des communautés sont détectées à différents instants du réseau dynamique, retrouver quelle communauté à t correspond à quelle communauté à $t+1$ est problématique [23]. Ainsi je propose d'adapter les techniques exposées au cours de la thèse au cas dynamique en mettant à profit les deux atouts que nous venons de mettre en avant.

Le formalisme des flots de liens

De nombreux travaux visent à intégrer la dynamique dans la théorie des graphes. On peut, par exemple, étudier une série de graphes $G(t_1, t_2)$ consistant en une agrégation des interactions ayant eu lieu entre t_1 et t_2 [28]. Ces travaux sont très pertinents pour modéliser des systèmes dynamiques mais où les interactions sont relativement longues. Comme par exemple, des pages web connectés par des liens hypertextes ou bien Facebook, où des profils sont associés par des liens d'amitié. Cependant les systèmes qui, en plus d'être dynamiques, ont des interactions qui sont ponctuelles (ou de courte durée) ne peuvent pas être étudiés simplement sans perte d'information. Pour cela le formalisme des flots de liens a été introduit.

Un flot de liens est une séquence de triplets (t, u, v) où chaque triplet décrit une interaction qui a eu lieu entre les nœuds u et v au temps t (une durée d'interaction peut également être ajoutée). Ils sont donc utiles pour modéliser des systèmes où les interactions entre entités sont ponctuelles, ou de courte durée tels que : des échanges de courriels, des tweets/retweets sur Twitter, des transactions bancaires ou des contacts physiques entre personnes.

Des travaux ont récemment été proposés pour étudier comment des notions classiques en théorie des graphes s'adaptent aux flots de liens, comme la notion de densité ou de clique [151, 152]. Un autre travail a montré que la définition classique de communautés

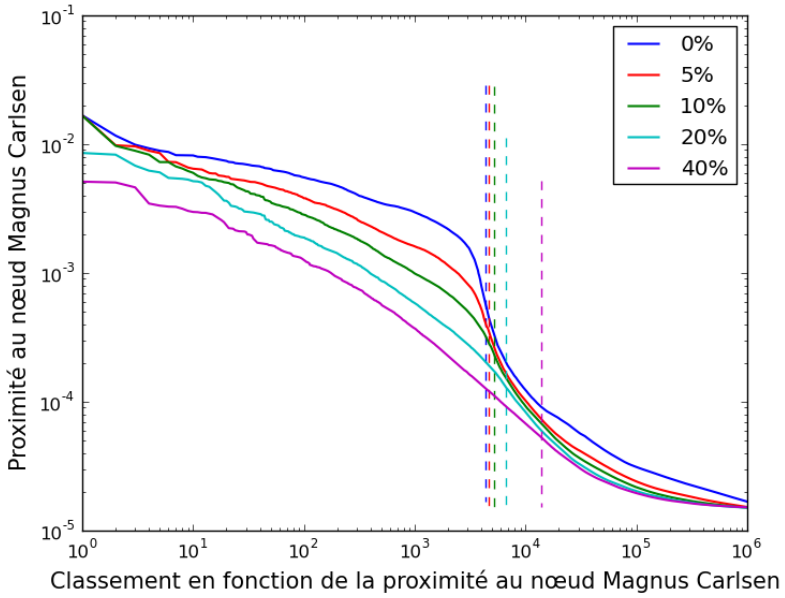


Figure 10.2 – Perturbations sur le réseau *Wikipédia 2012*. La proximité est calculée avec l'opinion propagée à partir du nœud "Magnus Carlsen". $x\%$ quantifie les perturbations et signifie que l'on a effectué "$x\%$ fois le nombre de liens dans le graphe" échanges aléatoires (sélectionner deux liens aléatoirement, les couper et connecter l'un avec l'autre et inversement).

se généralise difficilement à des systèmes avec des interactions ponctuelles et nécessite la détection d'autres types de structures [154]. Une notion de temps, différente de la notion de temps absolu mesuré en seconde, a également été mise en avant dans ce type de système [15]. Il s'agit d'un temps intrinsèque basé sur les modifications apparaissant dans le système lui-même.

Dans le cadre du formalisme des flots de liens et de la détection de communautés, je propose d'adapter l'approche à base de mesure de proximité présentée dans cette thèse afin de détecter un certain type de communauté (plutôt des communautés de liens que de nœuds) dans ces systèmes avec des interactions ponctuelles. Je propose ainsi de répéter le protocole mis en place dans cette thèse : (i) construire une mesure de proximité entre deux liens incorporant à la fois des informations topologiques et temporelles, (ii) mesurer la proximité d'un lien d'intérêt à tous les autres liens du flot et (iii) détecter des irrégularités ("plateau / décroissance / plateau") dans la décroissance de ces valeurs de proximité. Une irrégularité traduirait la présence d'un groupe de liens proches du lien d'intérêt et ayant du sens.

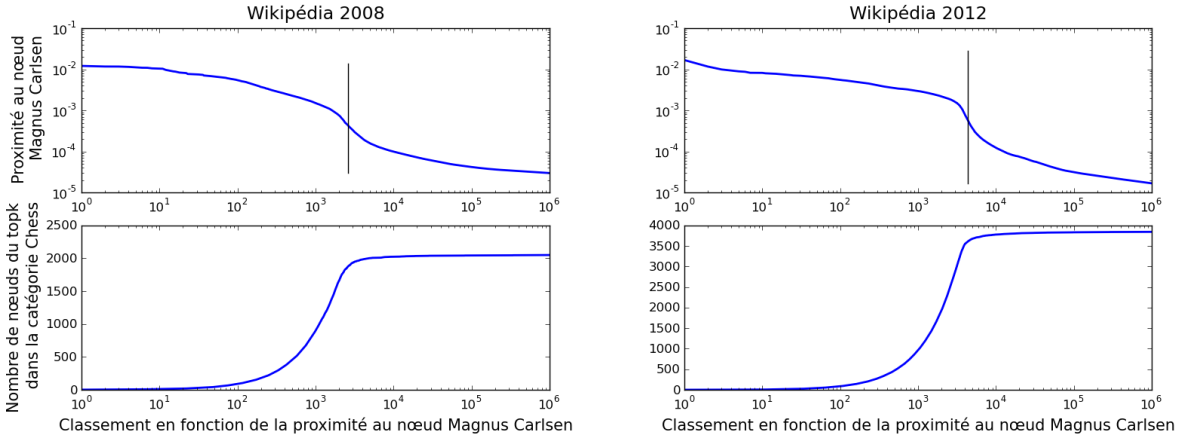


Figure 10.3 – *Wikipédia 2008* (gauche) VS *Wikipédia 2012* (droite). La proximité est calculée dans les deux réseaux avec l'opinion propagée à partir du nœud "Magnus Carlsen".

Dynamique de triangles dans les réseaux dirigés

La plupart des réseaux que nous avons étudié sont dirigés, pourtant nous avons complètement ignoré cette orientation des liens en travaillant avec des versions rendues symétriques. Cette perte d'information, classique quand on étudie des communautés, devrait être évitée. Cependant, quand on parle de communautés en les définissant par "un ensemble de nœuds fortement connectés entre eux et peu connectés vers l'extérieur", il est difficile de prendre en compte cette orientation autrement qu'en mettant un poids de deux pour les liens bidirectionnels.

Des méthodes de détections de communauté dans les réseaux dirigés existent cependant, comme en témoigne l'article d'état de l'art [102]. Ces méthodes consistent par exemple à (i) adapter des méthodes existantes (ii) transformer, de façon intelligente, le réseau dirigé en un autre réseau non dirigé (parfois pondéré ou biparti) et à appliquer des méthodes de détection de communautés existantes ou (iii) considérer qu'une communauté est un ensemble de nœuds ou l'information a tendance à rester, plutôt "qu'un ensemble de nœuds fortement connectés entre eux et peu connectés vers l'extérieur".

Afin d'étendre ces méthodes et d'aller plus loin pour prendre en compte du mieux possible cette orientation, je proposons d'étudier des structures simples, mais plus complexes que les liens : les triangles et les V-liens [1]. En effet, les triangles (non orientés) ont été montrés comme des structures importantes pour les communautés [66, 125], pourtant l'étude des triangles dirigés n'est que peu présente dans la littérature [107] et l'étude de leurs relations avec les communautés l'est encore moins [55].

Je suis persuadé que leur étude, particulièrement l'étude de leur formation à l'intérieur des communautés et entre les communautés, pourra donner des connaissances utiles pour leur détection et en particulier pour la prise en considération de l'orientation dans les

1. Un V-lien est un triplet de nœuds dont seulement 2 sont connectés

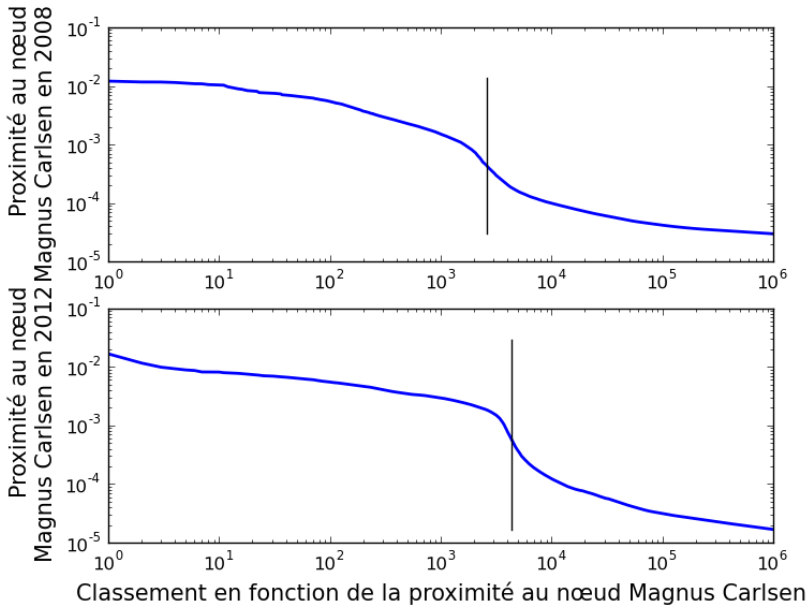


Figure 10.4 – *Wikipédia 2008* VS *Wikipédia 2012*. Nous voyons ici que la structure en "plateau / décroissance / plateau" de la courbe des proximité en 2012 est plus marquée que celle obtenue en 2008. On peut en déduire que (i) la communauté "Chess" est mieux dessinée en 2012 quelle ne l'était en 2008 ou bien "Magnus Carlsen" est plus central à la communauté "Chess" en 2012 qu'il ne l'était en 2008. On voit également que la communauté "Chess" a grossi.

méthodes de détection de communautés et dans la conception de mesures de proximité.

Afin de commencer cette étude, nous avons modifié le code d'énumération de triangles non dirigés de [90] (algorithme appelé *compact-forward*) pour pouvoir compter les 6 types de V-liens dirigés différents (énumérés figure 10.5) et les 7 types de triangles dirigés différents (énumérés figure 10.6). L'algorithme a une complexité temporelle en $O(m^{\frac{3}{2}})$ (m correspond au nombre de liens dans le réseau) et peut aisément traiter des graphes avec des centaines de millions de liens en quelques minutes.

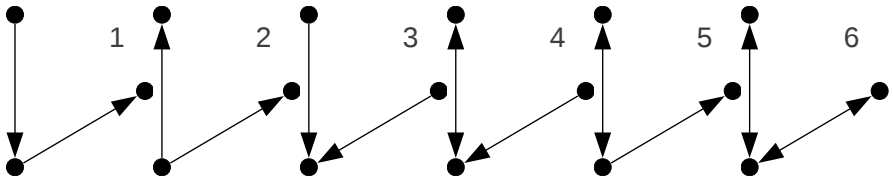


Figure 10.5 – Les six types de V-liens différents

Un premier résultat non trivial intéressant est que le nombre de triangles de type 7 est 12 fois (resp. 45 fois) plus important que ce qu'il devrait être pour le réseau *Wikipédia*

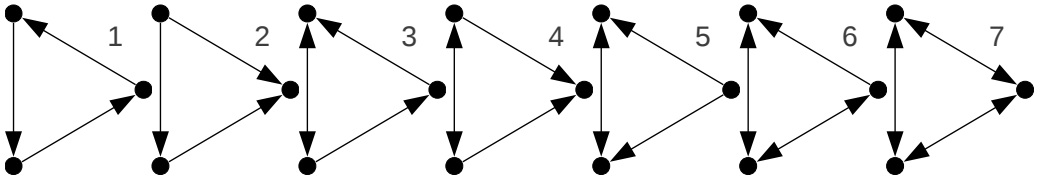


Figure 10.6 – Les sept types de triangles différents

2008 (resp. *Twitter 2009*) comme montré par le tableau 10.2. Ceci peut être apparenté à une sorte de clustering en version orientée.

Réseau	*Wikipédia 2008*	*Twitter 2009*
liens simples	64.95%	77.89%
liens doubles	35.05%	22.11%
triangle 1	0.011	0.021
triangle 2	0.448	0.266
triangle 3	0.028	0.172
triangle 4	1.539	0.733
triangle 5	1.418	0.436
triangle 6	0.797	2.224
triangle 7	12.608	45.9634

Tableau 10.2 – Nombre de triangles de chaque type divisé par le nombre de triangles espéré en supposant que le graphe est le même, mais que les liens sont aléatoirement doubles ou simples (et orientés aléatoirement) en gardant les proportions de liens simples et doubles.

L'étude de la dynamique de ces triangles orientés et leur comparaison à l'intérieur et à l'extérieur des communautés reste à faire et est une perspective de la thèse.

Annexe A

Jeux de données utilisés

Afin de valider les mesures et applications proposées dans cette thèse, nous avons réalisé des expériences sur des réseaux synthétiques et réels, de diverses tailles. Nous présentons tous ces réseaux dans cette annexe.

A.1 Graphes synthétiques

A.1.1 Petits graphes

Nous avons tout d'abord utilisé des graphes de petite taille afin de pouvoir facilement les représenter avec des outils de visualisation. Ceci nous a permis de tester plus avant nos intuitions et nous présentons ici les idées générales pour les construire.

Le modèle de graphe aléatoire $G(n, p)$ introduit par Erdos et Renyi [59] permet de générer des graphes aléatoires caractérisés par deux paramètres : (i) le nombre de nœuds n et (ii) la probabilité d'avoir un lien entre deux nœuds donnés p. Les graphes générés avec le modèle configurationnel [116] sont également utilisables. Outre le nombre de nœuds et le nombre de liens, ce modèle permet de choisir la distribution de degrés souhaitée et ainsi de se rapprocher de la structure des graphes de terrain qui ont généralement une distribution de degrés fortement hétérogène. Ces deux modèles de graphes aléatoires n'ont pas de structure communautaire *a priori*. Ils peuvent néanmoins servir à vérifier qu'un algorithme de détection de communautés ne trouve pas de communautés dans ce type de graphe.

En combinant ces graphes, on peut également en construire de nouveaux avec une pseudo-structure communautaire. Par exemple, si l'on génère un graphe formé de deux sous-graphes d'Erdos-Renyi, tous deux caractérisés par les paramètres (n, p) et que l'on connecte les deux sous-graphes avec une probabilité $p_2 < p$, on obtient un graphe avec deux groupes d'autant plus marqués que p_2 est petit devant p.

On peut également construire des structures en pseudo-communautés recouvrantes. Par exemple, on peut générer deux graphes avec le modèle d'Erdos-Renyi (n, p). On peut ensuite les superposer sur n_2 nœuds : on obtient alors deux communautés de taille n

se recouvrant sur n_2 nœuds. Plusieurs règles sont possibles pour connecter deux nœuds situés dans la partie recouvrante. Par exemple, on peut les connecter si et seulement si ils étaient connectés dans au moins l'un des deux graphes de départ ; leur probabilité de connexion est alors $p_{ov} = 1 - (1-p)^2 = 2p - p^2$, qui est comprise entre p et $2p$. On peut également les connecter si et seulement si ils étaient connectés dans un seul des deux graphes d'Erdos-Renyi ; leur probabilité de connexion est alors $p_{ov} = 2p(1-p) = 2p - 2p^2$, qui est un peu moins élevée que la probabilité précédente, mais reste comprise entre p et $2p$. Enfin, on peut simplement choisir $p_{ov} = p$, c'est-à-dire que deux nœuds ont la même probabilité d'être connectés, qu'ils soient dans un seul groupe ou à l'intersection de deux groupes, ou toute valeur supérieure ou inférieure à p suivant que l'on veut que les nœuds à l'intersection soient plus ou moins connectés. Ces différents choix permettent de définir plus précisément la notion de recouvrement souhaitée et de tester les algorithmes dans des conditions différentes.

On peut également utiliser le Stochastic Block Model (SBM) [74] qui généralise les approches précédentes. Par exemple, on peut prendre trois groupes de nœuds notés G_1, G_2 et G_3 de taille respective n_1, n_2 et n_3 et connecter les paires de nœuds avec les probabilités p_{ij} si un nœud est dans le groupe i et l'autre dans le groupe j comme sur la figure A.1, avec $p_{11} = p_{22} = p_{13} = p_{23} = p$, $p_{33} = p_{ov}$ et $p_{12} = 0$. On obtient la même structure que précédemment avec les graphes Erdos-Renyi recouvrants, mais avec la possibilité de fixer p_{ov} à n'importe quelle valeur. Ce modèle peut bien entendu être utilisé avec un nombre de groupes et des probabilités quelconques.

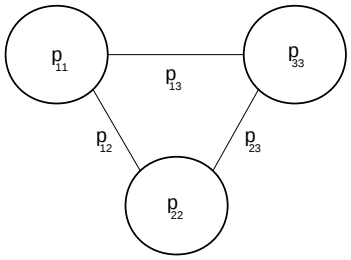


Figure A.1 – Schéma représentant les trois ensembles de nœuds et les probabilités de connexion pour générer le graphe jouet de deux communautés recouvrantes à l'aide du SBM.

En allant un peu plus loin, nous voyons ici qu'avec le SBM il est possible de générer n'importe quelle structure en communautés recouvrantes imaginable. Cependant le SBM n'est que peu utilisé à ces fins ; en effet, une telle structure serait laborieuse à créer. De plus le SBM ne génère pas naturellement une distribution de degrés en loi de puissance (comme classiquement observé dans les réseaux réels) et donnerait donc des réseaux peu réalistes. Un premier pas combinant les avantages du SBM tout en gardant le réalisme de la distribution en loi de puissance des degrés est abordé dans le chapitre 8.

La génération de ces graphes synthétiques avec une structure de communautés recouvrantes soulève également une question importante, peu étudiée dans la littérature. Elle concerne la structure en communautés des réseaux réels : *comment varie la probabilité de connexion d'une paire de nœuds en fonction du nombre de communautés partagées par la paire de nœuds et le nombre de communautés de chacun de ces deux nœuds ?* Bien que quelques éléments de réponse à cette question figurent dans [164] (les auteurs montrent que dans les réseaux avec communautés "vérité de terrain" qu'ils ont rassemblés, les nœuds qui partagent plus de communautés sont plus connectés entre eux que ceux qui en partagent moins), cette question reste très complexe. La réponse peut dépendre du type de réseau ou même du type de communautés. Nous avons d'ailleurs effectué des travaux préliminaires dans ce sens en étudiant des métriques et en en proposant de nouvelles [146]. Mais cette question n'étant pas centrale dans cette thèse et pouvant constituer un sujet de thèse à elle seule, nous nous contenterons d'aller au plus simple et de générer des graphes jouets suivant les premiers modèles que nous avons exposés.

A.1.2 Benchmarks LFR et LF

Le modèle le plus utilisé pour générer des réseaux réalistes avec des communautés non-recouvrantes est benchmark de Lancicineti, Fortunato et Radici (LFR) [89]. Il permet de générer des réseaux avec une distribution de degrés et des tailles de communautés en loi de puissance. La génération d'un tel réseau est contrôlée par les paramètres suivants :

- N le nombre de nœuds,
- k le degré moyen,
- $maxk$ le degré maximum,
- mu le paramètre de mixage permettant de contrôler le niveau de définition des communautés,
- $t1$ le coefficient de la loi de puissance pour la distribution de degrés (la valeur utilisée est $-t1$),
- $t2$ le coefficient de la loi de puissance pour la distribution des tailles de communautés (la valeur utilisée est $-t2$),
- $minc$ la taille minimum des communautés et
- $maxc$ la taille maximum des communautés.

Le modèle a été généralisé aux communautés recouvrantes dans [87] avec benchmark de Lancicineti et Fortunato (LF). Deux autres paramètres contrôlent l'étendue des recouvrements :

- on le nombre de nœuds appartenant à plusieurs communautés et
- om le nombre de communautés auxquelles ces nœuds appartiennent.

Ce modèle est limité : les nœuds qui appartiennent à plusieurs communautés sont tous dans le même nombre de communautés. Il est cependant l'un des seuls permettant de générer des réseaux réalistes avec une structure en communautés recouvrantes et est en conséquence le modèle le plus utilisé. D'autres modèles permettent de générer des réseaux avec des communautés recouvrantes, par exemple le SBM présenté précédemment. Ces autres modèles sont cependant moins réalistes ou plus difficiles à calibrer et nous utiliserons

donc souvent les benchmarks LFR et LF au cours de cette thèse. Nous regrettons toutefois l'absence d'un modèle plus facile à paramétrer qui permettrait de générer des communautés recouvrantes plus réalistes.

A.2 Réseaux réels

A.2.1 Petits réseaux classiques

Il existe de nombreux réseaux réels de petite taille, et certains sont très souvent utilisés dans le domaine des réseaux complexes, comme *karaté club* qui comporte 34 nœuds et dont l'article qui le présente [165] est cité plus de 1800 fois. Ils sont utiles pour plusieurs raisons : (i) comme ils sont issus du monde réel, on peut disposer de connaissances complémentaires comme par exemple une structure communautaire explicite, (ii) étant petits, ils sont facilement visualisables et (iii) ils ont certaines propriétés que les modèles ont parfois du mal à capturer. Dans le cadre de cette thèse, ils nous ont permis de tester plus avant les intuitions que nous avons eues, en analysant nos résultats avec des outils de visualisation de graphe. Voici le détail de ces réseaux :

- *netscience* : un réseau où les nœuds sont des scientifiques travaillant dans le domaine des réseaux et les liens témoignent des collaborations que ces scientifiques ont pu avoir (1589 nœuds, 2742 liens) [115]. Nous avons extrait et travaillé avec sa composante géante (379 nœuds, 914 liens).
- *dauphins* : un réseau social de 62 dauphins (nœuds) vivant au large du Doubtful Sound en Nouvelle-Zélande, liés entre eux s'ils ont fréquemment été observés ensemble (159 liens) [100].
- *blogs politiques* : un réseau où les nœuds sont des blogs politiques et les liens sont des citations consistant en des hyperliens entre ces blogs [9]. Ce réseau contient 1222 nœuds et 16717 liens. De plus, les nœuds ont été étiquetés manuellement : il y a 759 blogs libéraux et 443 blogs conservateurs.
- *livres politiques* : un réseau où les nœuds sont des livres politiques et les liens sont des citations entre ces livres. Il contient 105 nœuds et 441 liens [1].
- *mots adjacents* : un réseau d'adjectifs et de noms fréquents écrits dans le roman "David Copperfield" de Charles Dickens. Il comprend 112 mots (nœuds) et deux mots sont liés lorsqu'ils apparaissent fréquemment côte à côte (425 liens) [115].
- *les misérables* : un réseau de personnages (77 nœuds) présents dans "Les Misérables" de Victor Hugo. Deux personnages sont reliés s'ils apparaissent dans la même scène (254 liens) [83].
- *C-elegans* : le réseau de neurones d'un Caenorhabditis elegans (un nématode de la famille des Rhabditidae, c'est-à-dire un certain type de ver) : 297 neurones liés par 2359 synapses [158].
- *karaté club* : relations sociales entre les personnes fréquentant un club de karaté (34 nœuds), après une dispute au sein de celui-ci ayant entraîné une séparation des membres en deux groupes (78 liens) [165].

— *football Américain* : un réseau d'équipes de football Américain dans lequel deux équipes sont liées si elles ont joué un match pendant le championnat (115 nœuds, 616 liens) [70].

A.2.2 Réseaux SNAP avec communautés "vérité de terrain"

Dans [163], les auteurs présentent des jeux de données de plusieurs réseaux avec des communautés connues :

— Un réseau d'achats couplés de produits Amazon : les nœuds sont des produits vendus sur Amazon et il y a un lien entre deux produits qui sont achetés fréquemment ensemble. Les communautés sont définies par une hiérarchie de catégories de produits.
— Un réseau de collaborations de scientifiques DBLP : les nœuds sont des auteurs d'articles scientifiques et un lien relie deux co-auteurs. Les communautés sont des conférences et les auteurs sont inclus dans les conférences où ils ont publié.
— Des réseaux sociaux en ligne où les nœuds sont des profils liés par des liens d'amitié. Les communautés sont des catégories créées par les utilisateurs.

Cependant ces réseaux sont très incomplets, les communautés sont souvent très partiellement étiquetées. Les auteurs proposent même d'utiliser les nœuds ayant les mêmes étiquettes et étant dans la même composante connexe (les nœuds ayant les mêmes étiquettes n'étant parfois pas connectés), ou d'utiliser les 5000 communautés ayant la meilleure qualité suivant des fonctions ad hoc. Pour ces raisons, nous n'utiliserons que peu ces réseaux.

A.2.3 Réseau de pages Wikipédia catégorisées

Wikipédia est une encyclopédie en ligne fonctionnant sur le principe du wiki, c'est-à-dire que les utilisateurs peuvent eux-mêmes créer et modifier les pages. Ces pages sont liées entre elles par des liens hypertextes. L'historique des modifications est archivé. Il existe de plus des catégories auxquelles les pages peuvent appartenir et l'affectation des pages aux catégories est faite par les utilisateurs.

Nous avons utilisé le réseau de [121] qui a été construit en 2008 avec la totalité des pages Wikipédia en anglais et les liens entre celles-ci. Ce réseau est composé de :

— 2 070 486 pages Wikipédia correspondant aux nœuds du réseau,
— 46 092 182 liens hypertextes entre ces pages ; nous utiliserons principalement la version non-orientée contenant 42 336 692 liens,
— 265 432 catégories avec, de plus, une structure hiérarchique de ces catégories illustrée par un réseau de 543 722 liens dirigés entre ces catégories. Il existe des cycles dans ce réseau. En effet, ce réseau étant créé par des utilisateurs, on arrive qu'une catégorie A soit une sous-catégorie d'une catégorie B, que la catégorie B soit une sous-catégorie de la catégorie C et que C soit à son tour une sous-catégorie de A. Ce type de boucle peut être enlevé efficacement, par exemple, à l'aide de l'algorithme détaillé

dans [121]. On obtient alors un réseau dirigé sans cycle (Directed Acyclic Graph ou DAG en anglais) de catégories mères et filles : 265 432 catégories et 539 745 liens.

Au cours de la thèse, ce réseau sera appelé : réseau *Wikipédia 2008*. Nous avons également ré-extrait ce réseau de manière à en avoir une version plus récente, le réseau équivalent daté du 2 juillet 2012 est composé de :

— 4 030 943 nœuds,
— 253 128 051 liens orientés, ce qui donne 187 433 442 liens pour le réseau symétrisé,
— 874 761 catégories et un DAG de 2 157 687 liens orientés.

Au cours de la thèse, ce réseau sera appelé : réseau *Wikipédia 2012*.

A.2.4 Réseau social google scholar

Nous avons crawlé google scholar en septembre 2014 de manière à avoir le profil de chaque utilisateur avec ses co-auteurs et ses domaines de recherche. Nous avons fait une exploration BFS (Breath First Search) en partant de l'utilisateur "NVX8nHgAAAAJ", soit "Maximilien Danisch". Les co-auteurs et les domaines de recherche d'un utilisateur sont renseignés par l'utilisateur lui-même. La figure A.2 montre une capture d'écran d'un profil d'utilisateur.

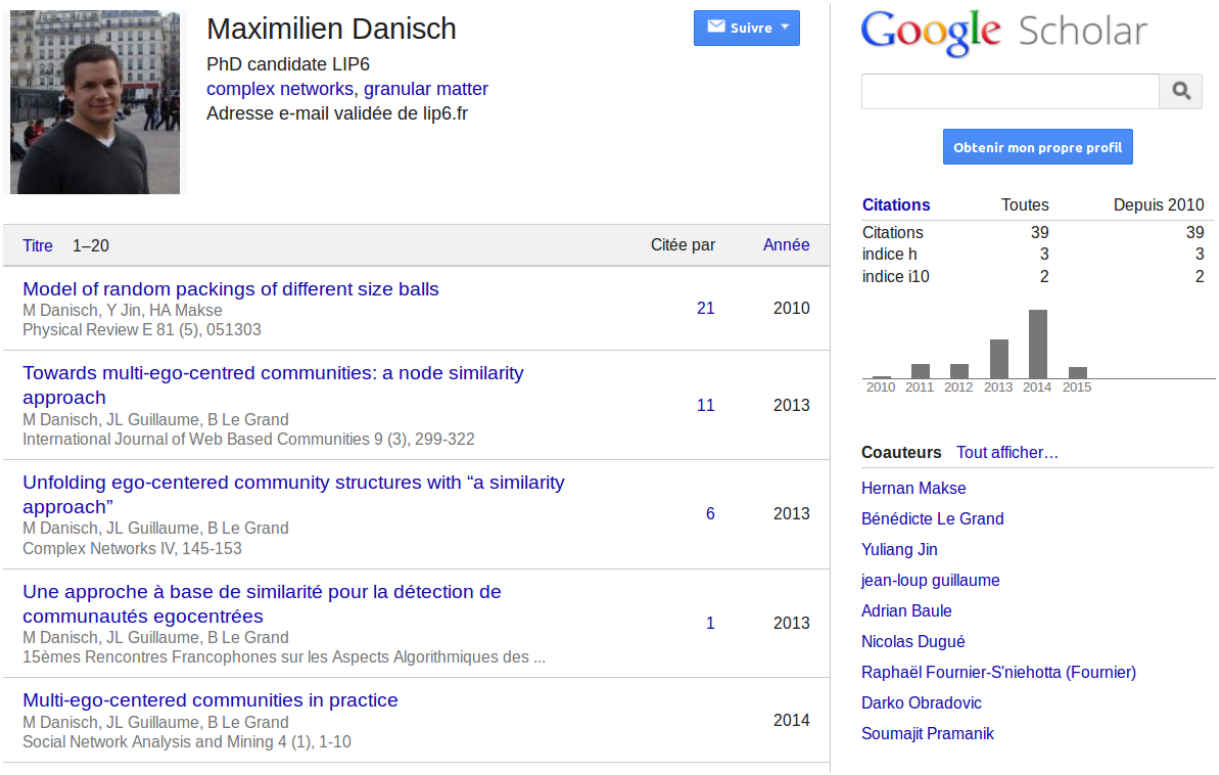


Figure A.2 – Capture d'écran d'un profil d'utilisateur google scholar. Les domaines de recherche sont situés en haut sous le nom de l'utilisateur et les co-auteurs sont situés à droite.

Nous avons ainsi obtenu un réseau de :
— 287 426 nœuds,
— 2 153 962 liens dirigés, ce qui donne 871 001 liens pour le réseau rendu symétrique (en effet un utilisateur indique lui-même ses coauteurs ce qui donne lieu à un réseau dirigé bien que les liens de co-auteurs soient intrinsèquement dirigés),
— 16 773 domaines de recherche.

Nous assimilerons les domaines de recherche renseignés par les utilisateurs à des communautés.

Au cours de la thèse, ce réseau sera appelé *Google scholar*.

A.2.5 Réseau DBLP

Le Digital Bibliography & Library Project (DBLP, littéralement "Projet de bibliothèque et de bibliographie numérique") est un site web publiant un catalogue de bibliographies en informatique. Les articles, leurs citations et leurs domaines de recherche associés ont été crawlés et rendus publics [38]. Le réseau comporte :
— 233 659 nœuds (articles),
— 865 794 liens de citation (que l'on rend symétrique),
— 24 domaines de recherche.

Au cours de la thèse, ce réseau sera appelé : réseau *DBLP*.

A.2.6 Réseau de requêtes pair-à-pair

Nous avons également travaillé avec un réseau de requêtes pair-à-pair. Le jeu de données consiste en des requêtes par mots-clefs envoyées par les utilisateurs d'eDonkey au moteur de recherche de leur application cliente. La collecte des données a eu lieu pendant 10 semaines en continu sur l'un des serveurs les plus importants du réseau eDonkey [12]. Le jeu de données est constitué de 107 226 021 requêtes, provenant de 23 892 531 adresses IP différentes, anonymisées à la volée. Chaque requête est horodatée et comporte un port de communication ainsi que la liste des mots-clefs utilisés pour la recherche.

Afin d'obtenir des étiquettes pour ces données correspondant à une communauté "vérité de terrain", nous utiliserons le travail de [91], où les auteurs ont présenté une méthodologie pour détecter les requêtes effectuées dans l'intention d'obtenir une classification des contenus à caractère pédopornographique basée uniquement sur la sémantique des requêtes.

A.2.7 Réseau social Twitter

Twitter est un outil très largement utilisé pour partager, rechercher et débattre des informations ou des évènements du quotidien. Twitter compte plus de 200 millions d'utilisateurs actifs. La quantité d'information échangée sur Twitter est considérable : 1 milliard de tweets (courts messages de moins de 140 caractères) sont postés tous les deux jours et demi [128]. Twitter est donc à la fois un service de micro-blogging et un outil média. Mais Twitter est également un service de réseau social en ligne. En effet, Twitter inclut de

nombreux outils destinés à l'échange entre utilisateurs. Par exemple pour voir les *tweets* d'autres utilisateurs s'afficher sur son fil d'actualité, il est nécessaire de s'abonner à ces utilisateurs. Si u s'abonne à v, on dit que u *suit* v et on dit que u est un *abonné* de v. Réciproquement, v est un abonnement de u.

Nous avons travaillé sur un réseau extrait de Twitter en 2009 et présenté dans [37]. Ce réseau consiste en 52 579 682 nœuds (comptes Twitter) et 1 963 263 821 liens dirigés entre ces utilisateurs, ce qui donne un réseau asymétrisé de 1 614 106 500 liens non-dirigés.

Au cours de la thèse, ce réseau sera appelé *Twitter 2009*.

Nous n'avons pas pu extraire d'autres réseaux plus récents suffisamment grands pour être intéressants à cause des limitations imposées par l'interface de programmation Twitter. En effet, actuellement, avec une requête d'abonnés (resp. d'abonnements), on peut obtenir 5000 abonnés (resp. abonnements) d'un utilisateur donné. Une requête doit être effectuée au travers d'un compte Twitter valide et le nombre de requêtes est limité à une requête par minute. Ainsi avec plus de 200 millions d'utilisateurs, cela prendrait autour de 500 ans pour crawler tout Twitter. Même en utilisant 5000 comptes, il faudrait un mois pour crawler tout Twitter. Cette opération étant cependant laborieuse à mettre en œuvre et contraire à la charte d'utilisation de l'API Twitter, nous ne nous sommes pas attardés dessus.

Bibliographie

[1] Books about us politics. http://networkdata.ics.uci.edu/data.php?d=polbooks.

[2] Klout, the standard for influence. http://www.klout.com.

[3] Kred story. http://www.kred.com.

[4] Review your Twitter account — free Twitter analyzer — Twitter grader - http ://twittergrader.mokumax.com/. http://twittergrader.mokumax.com/.

[5] Twitalyzer, serious analytics for social business. http://www.twitalyzer.com - As of September 28, 2013 Twitalyzer has decided to no longer sell new subscriptions.

[6] Twitter : We now have over 200 million accounts, 2011. http://www.huffingtonpost.com/2011/04/28/twitter-number-of-users_n_855177.html.

[7] The telegraph : Twitter in numbers, 2013. http://www.telegraph.co.uk/technology/twitter/9945505/Twitter-in-numbers.html.

[8] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3) :211–230, 2003.

[9] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election : divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.

[10] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307) :761–764, 2010.

[11] Frédéric Aidouni, Matthieu Latapy, and Clémence Magnien. Ten weeks in the life of an eDonkey server. *International Workshop on Hot Topics in P2P Systems*, 2009.

[12] Frédéric Aidouni, Matthieu Latapy, and Clémence Magnien. Ten weeks in the life of an edonkey server. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–5. IEEE, 2009.

[13] Morteza Alamgir and Ulrike Von Luxburg. Multi-agent random walks for local clustering on graphs. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 18–27. IEEE, 2010.

[14] Richard D Alba. A graph-theoretic definition of a sociometric clique†. *Journal of Mathematical Sociology*, 3(1) :113–126, 1973.

[15] Alice Albano, Jean-Loup Guillaume, Sébastien Heymann, and Bénédicte Le Grand. A matter of time-intrinsic or extrinsic-for diffusion in evolving complex networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 202–206. IEEE, 2013.

[16] Reid Andersen, Christian Borgs, Jennifer Chayes, John Hopcraft, Vahab S Mirrokni, and Shang-Hua Teng. Local computation of pagerank contributions. In *Algorithms and Models for the Web-Graph*, pages 150–165. Springer, 2007.

[17] Reid Andersen and Fan Chung. Detecting sharp drops in pagerank and a simplified local partitioning algorithm. In *Theory and Applications of Models of Computation*, pages 1–12. Springer, 2007.

[18] Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 475–486. IEEE, 2006.

[19] Reid Andersen, Fan Chung, and Kevin Lang. Local partitioning for directed graphs using pagerank. *Internet Mathematics*, 5(1-2) :3–22, 2008.

[20] Reid Andersen and Kevin J Lang. Communities from seed sets. In *Proceedings of the 15th international conference on World Wide Web*, pages 223–232. ACM, 2006.

[21] I. Anger and C. Kittl. Measuring influence on Twitter. In *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies*, pages 1–4. ACM, 2011.

[22] David Auber. Tulip—a huge graph visualization framework. In *Graph Drawing Software*, pages 105–126. Springer, 2004.

[23] Thomas Aynaud, Eric Fleury, Jean-Loup Guillaume, and Qinna Wang. Communities in evolving networks : Definitions, detection, and analysis techniques. In *Dynamics On and Of Complex Networks, Volume 2*, pages 159–200. Springer, 2013.

[24] Lars Backstrom and Jure Leskovec. Supervised random walks : predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 635–644. ACM, 2011.

[25] James P Bagrow and Erik M Bollt. Local method for detecting communities. *Physical Review E*, 72(4) :046108, 2005.

[26] Albert L. Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, 286 :509–512, 1999.

[27] Mathieu Bastian, Sebastien Heymann, Mathieu Jacomy, et al. Gephi : an open source software for exploring and manipulating networks. *ICWSM*, 8 :361–362, 2009.

[28] Prithwish Basu, Amotz Bar-Noy, Ram Ramanathan, and Matthew P Johnson. Modeling and analysis of time-varying graphs. *arXiv preprint arXiv :1012.0260*, 2010.

[29] Vladimir Batagelj and Ulrik Brandes. Efficient generation of large random networks. *Physical Review E*, 71(3) :036113, March 2005.

[30] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. Springer, 2006.

[31] Joseph Blitzstein and Persi Diaconis. A sequential importance sampling algorithm for generating random graphs with prescribed degrees. *Internet Mathematics*, 6(4) :489–522, 2011.

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

[33] B. Bollobas. *Random Graphs*. London : Academic Press, 1985.

[34] Béla Bollobás and Oliver Riordan. Clique percolation. *Random Structures & Algorithms*, 35(3) :294–322, 2009.

[35] Andrei Z Broder. On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997. Proceedings*, pages 21–29. IEEE, 1997.

[36] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in Twitter : The million follower fallacy. In *Proceedings of ICWSM*. AAAI, 2010.

[37] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and P Krishna Gummadi. Measuring user influence in twitter : The million follower fallacy. *ICWSM*, 10 :10–17, 2010.

[38] Tanmoy Chakraborty, Sandipan Sikdar, Vihar Tammana, Niloy Ganguly, and Animesh Mukherjee. Computer science fields as ground-truth communities : Their impact, rise and fall. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, pages 426–433. IEEE, 2013.

[39] Tanmoy Chakraborty, Sriram Srinivasan, Niloy Ganguly, Animesh Mukherjee, and Sanjukta Bhowmick. On the permanence of vertices in network communities. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1396–1405. ACM, 2014.

[40] Jiyang Chen, Osmar Zaïane, and Randy Goebel. Local community identification in social networks. In *Social Network Analysis and Mining, 2009. ASONAM'09. International Conference on Advances in*, pages 237–242. IEEE, 2009.

[41] Hocine Cherifi. *Complex Networks and Their Applications*. Cambridge Scholars Publishing, 2014.

[42] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Who is tweeting on Twitter : Human, Bot, or Cyborg ? In *ACSAC*, pages 21–30. ACM, 2010.

[43] Aaron Clauset. Finding local community structure in networks. *Physical review E*, 72(2) :026132, 2005.

[44] Sara Cohen, Benny Kimelfeld, and Georgia Koutrika. A survey on proximity measures for social networks. In *Search Computing*, pages 191–206. Springer, 2012.

[45] Maximilien Danisch, Nicolas Dugué, and Anthony Perez. On the importance of considering social capitalism when measuring influence on twitter.

[46] Maximilien Danisch, Nicolas Dugué, and Anthony Perez. Prendre en compte le capitalisme social dans la mesure de l'influence sur twitter.

[47] Maximilien Danisch, Jean-Loup Guillaume, and Bénédicte Le Grand. Déplier les structures communautaires egocentrées-une approche à base de similarité.

[48] Maximilien Danisch, Jean-Loup Guillaume, and Bénédicte Le Grand. Unfolding ego-centered community structures with "a similarity approach". In *Complex Networks IV*, pages 145–153. Springer, 2013.

[49] Maximilien Danisch, Jean-Loup Guillaume, and Bénédicte Le Grand. Complétion de communautés par l'apprentissage d'une mesure de proximité. In *ALGOTEL 2014–16èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, pages 1–4, 2014.

[50] Maximilien Danisch, Jean-Loup Guillaume, and Bénédicte Le Grand. Multi-ego-centered communities in practice. *Social Network Analysis and Mining*, 4(1) :1–10, 2014.

[51] Maximilien Danisch, Jean-Loup Guillaume, Bénédicte Le Grand, et al. Learning a proximity measure to complete a community. In *The 2014 International Conference on Data Science and Advanced Analytics (DSAA2014)*.

[52] Maximilien Danisch, Jean-Loup Guillaume, Bénédicte Le Grand, et al. Towards multi-ego-centered communities : a node similarity approach. *Int. J. of Web Based Communities*, 2012.

[53] Maximilien Danisch, Jean-Loup Guillaume, Bénédicte Le Grand, et al. Une approche à base de similarité pour la détection de communautés egocentrées. *15èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel)*, pages 1–4, 2013.

[54] Charo I Del Genio, Hyunju Kim, Zoltán Toroczkai, and Kevin E Bassler. Efficient and exact sampling of simple graphs with given arbitrary degree sequence. *PloS one*, 5(4) :e10012, 2010.

[55] Nicolas Dugué, Tennessy Kolubako, and Anthony Perez. Détection de zones denses en triplets significatifs dans un réseau orienté. *Atelier Fouille de Grands Graphes : Application à la bioinformatique*.

[56] Nicolas Dugué and Anthony Perez. Social capitalists on twitter : detection, evolution and behavioral analysis. *Social Network Analysis and Mining*, 4(1) :1–15, 2014.

[57] John Ellson, Emden Gansner, Lefteris Koutsofios, Stephen C North, and Gordon Woodhull. Graphviz—open source graph drawing tools. In *Graph Drawing*, pages 483–484. Springer, 2002.

[58] David Eppstein, Maarten Löffler, and Darren Strash. *Listing all maximal cliques in sparse graphs in near-optimal time*. Springer, 2010.

[59] P ERDdS and A R&WI. On random graphs i. *Publ. Math. Debrecen*, 6 :290–297, 1959.

[60] P. Erdos and A. Renyi. On random graphs. *Publ. Math. Debrecen*, 6 :290, 1959.

[61] Christos Faloutsos, Kevin S McCurley, and Andrew Tomkins. Fast discovery of connection subgraphs. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 118–127. ACM, 2004.

[62] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM '99 : Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication*, pages 251–262. ACM, 1999.

[63] Gary William Flake, Steve Lawrence, and C Lee Giles. Efficient identification of web communities. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–160. ACM, 2000.

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

[65] Raphaël Fournier and Maximilien Danisch. Mining bipartite graphs to improve semantic pedophile activity detection. In *Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference on*, pages 1–4. IEEE, 2014.

[66] Adrien Friggeri, Guillaume Chelius, and Eric Fleury. Triangles to capture social cohesion. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 258–265. IEEE, 2011.

[67] Adrien Friggeri and Eric Fleury. Finding cohesive communities with c^3. 2012.

[68] Thomas MJ Fruchterman and Edward M Reingold. Graph drawing by force-directed placement. *Software : Practice and experience*, 21(11) :1129–1164, 1991.

[69] S. Ghosh, B. Viswanath, F. Kooti, N. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K. Gummadi. Understanding and combating link farming in the Twitter social network. In *WWW*, pages 61–70, 2012.

[70] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12) :7821–7826, 2002.

[71] David F Gleich and C Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 597–605. ACM, 2012.

[72] S. L. Hakimi. On the realizability of a set of integers as degrees of the vertices of a linear graph. *Journal of the Society of Industrial and Applied Mathematics*, 10(3) :496–506, 1962.

[73] Magnus Rudolph Hestenes. *Methods of conjugate gradients for solving linear systems*, volume 49. 1952.

[74] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels : First steps. *Social networks*, 5(2) :109–137, 1983.

[75] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter : understanding microblogging usage and communities. In *Workshop on Web mining and social network analysis*, WebKDD/SNA-KDD '07, pages 56–65, 2007.

[76] Rushed Kanawati. Licod : Leaders identification for community detection in complex networks. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*, pages 577–582. IEEE, 2011.

[77] Rushed Kanawati. Seed-centric approaches for community detection in complex networks. In Gabriele Meiselwitz, editor, *Social Computing and Social Media*, volume 8531 of *Lecture Notes in Computer Science*, pages 197–208. Springer International Publishing, 2014.

[78] Rushed Kanawati. Yasca : A collective intelligence approach for community detection in complex networks. *arXiv preprint arXiv :1401.4472*, 2014.

[79] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1) :39–43, 1953.

[80] Reihaneh Rabbany Khorasgani, Jiyang Chen, and Osmar R Zaïane. Top leaders community detection approach in information networks. Citeseer, 2010.

[81] Hyunju Kim, Charo I Del Genio, Kevin E Bassler, and Zoltán Toroczkai. Constructing and sampling directed graphs with given degree sequences. *New Journal of Physics*, 14(2) :023012, 2012.

[82] Jon Kleinberg. An impossibility theorem for clustering. *Advances in neural information processing systems*, pages 463–470, 2003.

[83] Donald Ervin Knuth, Donald Ervin Knuth, and Donald Ervin Knuth. *The Stanford GraphBase : a platform for combinatorial computing*, volume 37. Addison-Wesley Reading, 1993.

[84] Yehuda Koren, Stephen C North, and Chris Volinsky. Measuring and extracting proximity in networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–255. ACM, 2006.

[85] Jussi M Kumpula, Mikko Kivelä, Kimmo Kaski, and Jari Saramäki. Sequential algorithm for fast clique percolation. *Physical Review E*, 78(2) :026109, 2008.

[86] KIM Kyungbaek. Two-step boosting for osn based sybil-resistant trust value of non-sybil identities. *IEICE TRANSACTIONS on Information and Systems*, 97(7) :1918–1922, 2014.

[87] Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80(1) :016118, 2009.

[88] Andrea Lancichinetti, Santo Fortunato, and János Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3) :033015, 2009.

[89] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4) :046110, 2008.

[90] Matthieu Latapy. Main-memory triangle computations for very large (sparse (power-law)) graphs. *Theoretical Computer Science*, 407(1) :458–473, 2008.

[91] Matthieu Latapy, Clémence Magnien, and Raphaël Fournier. Quantifying paedophile queries in a large p2p system. In *INFOCOM, 2011 Proceedings IEEE*, pages 401–405. IEEE, 2011.

[92] Matthieu Latapy, Clémence Magnien, and Raphaël Fournier. Quantifying paedophile activity in a large P2P system. *Information Processing and Management*, 2012.

[93] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

[94] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7) :1019–1031, 2007.

[95] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3) :503–528, 1989.

[96] Fabrizio Luccio and Mariagiovanna Sami. On the decomposition of networks in minimally interconnected subnetworks. *Circuit Theory, IEEE Transactions on*, 16(2) :184–188, 1969.

[97] R Duncan Luce. Connectivity and generalized cliques in sociometric group structure. *Psychometrika*, 15(2) :169–190, 1950.

[98] R Duncan Luce and Albert D Perry. A method of matrix analysis of group structure. *Psychometrika*, 14(2) :95–116, 1949.

[99] Feng Luo, James Z Wang, and Eric Promislow. Exploring local community structures in large networks. *Web Intelligence and Agent Systems*, 6(4) :387–400, 2008.

[100] David Lusseau. The emergent properties of a dolphin social network. *Proceedings of the Royal Society of London. Series B : Biological Sciences*, 270(Suppl 2) :S186–S188, 2003.

[101] Linyuan Lv, Ci-Hang Jin, and Tao Zhou. Effective and efficient similarity index for link prediction of complex networks. *arXiv preprint arXiv :0905.3558*, 2009.

[102] Fragkiskos D Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks : A survey. *Physics Reports*, 533(4) :95–142, 2013.

[103] M McCord and M Chuah. Spam detection on twitter using traditional classifiers. In *Autonomic and Trusted Computing*, pages 175–186. Springer, 2011.

[104] Qiaozhu Mei, Dengyong Zhou, and Kenneth Church. Query suggestion using hitting time. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 469–478. ACM, 2008.

[105] J. Messias, L. Schmidt, R. Oliveira, and F. Benevenuto. You followed my bot ! transforming robots into influential users in Twitter. *First Monday*, 18(7), 2013.

[106] R. Milo, N. Kashtan, S. Itzkovitz, M. E. J. Newman, and U. Alon. On the uniform generation of random graphs with prescribed degree sequences. *Arxiv preprint cond-mat/0312028*, 2003.

[107] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs : simple building blocks of complex networks. *Science*, 298(5594) :824–827, 2002.

[108] Robert J Mokken. Cliques, clubs and clans. *Quality & Quantity*, 13(2) :161–173, 1979.

[109] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structures and Algorithms*, 6 :161–179, 1995.

[110] John W Moon and Leo Moser. On cliques in graphs. *Israel journal of Mathematics*, 3(1) :23–28, 1965.

[111] Jon Nathanson. How Klout Finally Matters, 2014. `http://www.slate.com/articles/business/the_bet/2014/05/klout_is_basically_dead_but_it_finally_matters.html`.

[112] M Newman, D Watts, and S Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences USA*, 99 :2566–2572, 2002.

[113] M. E. J. Newman. Mixing patterns in networks. *Physical Review E*, 67, 2003.

[114] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45 :167–256, 2003.

[115] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3) :036104, 2006.

[116] Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(2) :026118, 2001.

[117] Blaise Ngonmang, Maurice Tchuente, and Emmanuel Viennet. Local community identification in social networks. *Parallel Processing Letters*, 22(01), 2012.

[118] Darko Obradovic and Maximilien Danisch. Direct generation of random graphs exactly realising a prescribed degree sequence. In *Computational Aspects of Social Networks (CASoN), 2014 6th International Conference on*, pages 1–6. IEEE, 2014.

[119] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking : Bringing order to the web. 1999.

[120] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043) :814–818, 2005.

[121] Gergely Palla, Illés J Farkas, Péter Pollner, Imre Derényi, and Tamás Vicsek. Fundamental statistical features and self-similar properties of tagged networks. *New Journal of Physics*, 10(12) :123026, 2008.

[122] Symeon Papadopoulos, Yiannis Kompatsiaris, and Athena Vakali. A graph-based clustering scheme for identifying related tags in folksonomies. In *Data Warehousing and Knowledge Discovery*, pages 65–76. Springer, 2010.

[123] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn : Machine learning in Python. *JMLR*, 12 :2825–2830, 2011.

[124] Soumajit Pramanik, Maximilien Danisch, Qinna Wang, and Bivas Mitra. An empirical approach towards an efficient "whom to mention ?" twitter app.

[125] Arnau Prat-Pérez, David Dominguez-Sal, Josep M Brunat, and Josep-Lluis Larriba-Pey. Shaping communities out of triangles. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1677–1681. ACM, 2012.

[126] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9) :2658–2663, 2004.

[127] Fergal Reid, Aaron McDaid, and Neil Hurley. Percolation computation in complex networks. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 274–281. IEEE Computer Society, 2012.

[128] Simon Rodgers. Twitter blog, august 2013. `https://blog.twitter.com/2013/behind-the-numbers-how-to-understand-big-moments-on-twitter`.

[129] Moshe Rutgaizer, Yuval Shavitt, Omer Vertman, and Noa Zilberman. Detecting pedophile activity in BitTorrent networks. In Nina Taft and Fabio Ricciato, editors, *PAM*, volume 7192 of *Lecture Notes in Computer Science*, pages 106–115. Springer, 2012.

[130] A. Sameh. A Twitter analytic tool to measure opinion, influence and trust. *Journal of Industrial and Intelligent Information*, 1(1) :37–45, 2013.

[131] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1) :27–64, 2007.

[132] Stephen B Seidman and Brian L Foster. A graph-theoretic generalization of the clique concept*. *Journal of Mathematical sociology*, 6(1) :139–154, 1978.

[133] Devavrat Shah and Tauhid Zaman. Community detection in networks : The leader-follower algorithm. *arXiv preprint arXiv :1011.0774*, 2010.

[134] Ron Shamir, Roded Sharan, and Dekel Tsur. Cluster graph modification problems. In *Graph-Theoretic Concepts in Computer Science*, pages 379–390. Springer, 2002.

[135] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8) :888–905, 2000.

[136] Catherine Shu. Tech Crunch, march 2014.

[137] Jiří Šíma and Satu Elisa Schaeffer. On the np-completeness of some graph cluster measures. In *SOFSEM 2006 : Theory and Practice of Computer Science*, pages 530–537. Springer, 2006.

[138] George Gaylord Simpson. Mammals and the nature of continents. *Am. J. of Science*, (241) :1–41, 1943.

[139] Tom Snijders. Enumeration and simulation methods for 0-1 matrices with given marginals. *Psychometrika*, 56(3) :397–417, September 1991.

[140] Mauro Sozio and Aristides Gionis. The community-search problem and how to plan a successful cocktail party. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 939–948. ACM, 2010.

[141] Daniel A Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 81–90. ACM, 2004.

[142] Daniel A Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly-linear time graph partitioning. *arXiv preprint arXiv :0809.3232*, 2008.

[143] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3 :583–617, 2003.

[144] B. Suh, Lichan H., P. Pirolli, and E. H. Chi. Want to be retweeted ? large scale analytics on factors impacting retweet in Twitter network. In *SocialCom*, pages 177–184, 2010.

[145] Lionel Tabourier, Camille Roth, and Jean-Philippe Cointet. Generating constrained random graphs using multiple edge switches. *Journal of Experimental Algorithmics (JEA)*, 16 :1–7, 2011.

[146] Raphaël Tackx, Maximilien Danisch, and Fabien Tarissan. Structures biparties et communautés recouvrantes des graphes de terrains.

[147] Nikolaj Tatti and Aristides Gionis. Discovering nested communities. In *Machine Learning and Knowledge Discovery in Databases*, pages 32–47. Springer, 2013.

[148] R. Tinati, L. Carr, W. Hall, and J. Bentwood. Identifying communicator roles in Twitter. In *International Conference Companion on WWW*, pages 1161–1168. ACM, 2012.

[149] Hanghang Tong and Christos Faloutsos. Center-piece subgraphs : problem definition and fast solutions. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 404–413. ACM, 2006.

[150] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. 2006.

[151] Jordan Viard and Matthieu Latapy. Identifying roles in an ip network with temporal and structural density. In *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pages 801–806. IEEE, 2014.

[152] Jordan Viard, Matthieu Latapy, and Clémence Magnien. Computing maximal cliques in link streams. *arXiv preprint arXiv :1502.00993*, 2015.

[153] Fabien Viger and Matthieu Latapy. Efficient and simple generation of random simple connected graphs with prescribed degree sequence. In *Proceedings of the 11th international conference on Computing and Combinatorics*, volume 3595 of *LNCS*, pages 440–449. Springer, 2005.

[154] Qinna Wang. Link prediction and threads in email networks. In *The 2014 International Conference on Data Science and Advanced Analytics (DSAA2014)*, 2014.

[155] S Wasserman and Garry L Robins. An introduction to random graphs, dependence graphs, and p*. In Peter J. Carrington, John Scott, and Stanley Wasserman, editors, *Models and methods in social network analysis*, pages 148–161. Cambridge University Press, 2005.

[156] Stanley Wasserman, Katherine Faust, and Dawn Iacobucci. *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, 1994.

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

[158] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684) :440–442, 1998.

[159] B. Waugh, M. Abdipanah, O. Hashemi, S. A. Rahman, and D. M. Cook". The influence and deception of Twitter : The authenticity of the narrative and slacktivism in the australian electoral process. In *Proceedings of the 14th Australian Information Warfare Conference*, 2013.

[160] Joyce Jiyoung Whang, David F Gleich, and Inderjit S Dhillon. Overlapping community detection using seed set expansion. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2099–2108. ACM, 2013.

[161] Kevin A Wilson, Nathan D Green, Laxmikant Agrawal, Xibin Gao, Dinesh Madhusoodanan, Brian Riley, and James P Sigmon. Graph-based proximity measures. *Practical Graph Mining with R*, page 135, 2013.

[162] Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. Overlapping community detection in networks : the state of the art and comparative study. *arXiv preprint arXiv :1110.5813*, 2011.

[163] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, page 3. ACM, 2012.

[164] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale : a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM, 2013.

[165] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.

[166] Zeyuan A Zhu, Silvio Lattanzi, and Vahab Mirrokni. A local algorithm for finding well-connected clusters. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 396–404, 2013.