Analyse du capitalisme social sur Twitter

Nicolas Dugué

UNIVERSITÉ D'ORLÉANS

ÉCOLE DOCTORALE MIPTIS
MATHÉMATIQUES, INFORMATIQUE, PHYSIQUE THÉORIQUE ET INGÉNIEURIE DES SYSTÈMES

Laboratoire d'Informatique Fondamentale d'Orléans

THÈSE

présentée par :

Nicolas DUGUÉ

pour obtenir le grade de : **Docteur de l'université d'Orléans**

Discipline/ Spécialité : **Informatique**

Soutenue le **29 juin 2015**

Analyse du capitalisme social sur Twitter

Directeur :

Jean-Michel Couvreur - Professeur des Universités ..... Université d'Orléans

Co-encadrants :

Anthony Perez - Maître de conférence ..... Université d'Orléans
Frédéric Moal - Maître de conférence ..... Université d'Orléans

Rapporteurs :

Jean-Loup Guillaume - Professeur des Universités ..... Université de La Rochelle
Emmanuel Viennet - Professeur des Universités ..... Université Paris 13 Nord

Présidente du Jury :

Clémence MAGNIEN - Directrice de recherche .... CNRS et Université Pierre et Marie Curie

Examinateur :

Julien VELCIN - Maître de Conférences HDR ..... Université Lyon 2

"J'ai des tas d'idées brillantes et nouvelles, mais les nouvelles ne sont pas brillantes, et les brillantes ne sont pas nouvelles."
Marcel Achard

Remerciements

Je remercie Clémence Magnien d'avoir présidé mon jury de thèse, Emmanuel Viennet et Jean-Loup Guillaume d'avoir rapporté ma thèse et Julien Velcin d'avoir accepté d'être examinateur. Leur évaluation a entraîné des discussions pertinentes et soulevé des questionnements intéressants.

Je remercie Jean-Michel d'avoir accepté de diriger ma thèse et de m'avoir laissé une grande liberté.

Je remercie Frédéric d'avoir monté le projet INEX et de m'avoir donné l'opportunité d'entrer en thèse. Je le remercie également pour son co-encadrement. Enfin, je suis très heureux d'avoir partagé certains enseignements avec lui, il m'a énormément appris et motivé.

Je remercie Anthony Perez pour son travail de co-encadrement, les nombreuses discussions scientifiques, les encouragements, les relectures d'articles et les conseils prodigués tout au long de mon doctorat. Mais je retiendrai également de notre collaboration l'amitié qui en est née et les soirées passées à jouer aux fléchettes. Merci peranth !

En parlant de fléchettes et d'amitié, je remercie Maxime et Sylvain aka Daill ball, pour tous les moments passés ensemble à décompresser et à profiter de ces dernières années d'étude. C'est avec nostalgie que je pense à tous ces bons moments lorsque j'écris ces lignes. Mes pensées m'amènent alors vers Simon, mon premier ami au LIFO, avec qui j'ai partagé l'aventure de la thèse mais aussi celle de Slippy Charms. Ce fût épique :smileyému : !

Je remercie Hélène, qui m'a supporté quotidiennement pendant ma dernière année de thèse, sans jamais se plaindre. Elle a supporté mes sautes d'humeur, la musique hypnotique et incessante que je lui imposais, mon manque de temps pour elle et mes complaintes permanentes. Elle m'a encouragé lorsque j'étais fatigué et démotivé. Elle m'a soutenu lorsque j'étais stressé, et elle sait à quel point je l'ai été. Merci Hélène !

Je remercie Catherine Leborgne, qui m'a aidé à devenir qui je suis, semaine après semaine.

Je remercie Maximilien Danisch pour notre collaboration fructueuse, les bons moments passés à Berlin et MARAMI, ses encouragements et sa passion pour les K soc !

Je remercie Vincent Labatut pour toutes nos collaborations, les conseils donnés, les méthodes qu'il m'a faites découvrir et les bons moments passés à Avignon ou en conférence.

Je remercie Charlotte Laclau pour les discussions scientifiques mais aussi pour m'avoir fourni une oreille compatissante lors des nombreux coups de blues de ma thèse.

Je remercie à nouveau Jean-Loup Guillaume, pour son invitation à RNSC, ses encouragements mais surtout pour toutes les discussions débiles que nous avons eues.

Je remercie Yohan Boichut, auprès duquel j'ai beaucoup appris au cours de mes enseignements d'IHM et de MAC.

Je remercie l'équipe ERA du laboratoire ICARE et notamment Wahid Mellouki, François Bernard, Matthieu Cazaunau, Véronique Daële et Benoit Grosselin pour m'avoir fait découvrir le monde de la recherche et la passion qui peut y vibrer. Ce sont eux qui m'ont donné la vocation !

Je remercie Florence et Isabelle, ma famille d'adoption au LIFO.

Je remercie Wadoud Bousdira pour m'avoir guidé dans les enseignements que nous avons partagés durant ces années de thèse.

Je remercie le LIFO qui m'a chaleureusement accueilli. J'y ai passé des années merveilleuses.

Je remercie Adrien Guille pour les discussions autour de Twitter et les bonnes soirées de conférence.

Je remercie ma famille. Nous avons traversé des moments très difficiles, mais qui ont resserré nos liens.

Je remercie Joris Falip, Tennessy Kolubako, Amélie Daviau, Florian Bridoux, Simon Munier pour le travail accompli ensemble. J'ai eu beaucoup de chance de pouvoir travailler avec des étudiants aussi brillants et passionnés.

Je remercie Glory owl et Cyanide and Happiness de m'avoir fourni d'agréables moments de détente pendant mon travail.

Je remercie Slippy Charms pour tout ce que nous avons vécu qui a rythmé mes dernières années de thèse : les répétitions, enregistrements et concerts sur relents de Pontarlier.

Je remercie Christian Scott et Electric Wizard que j'ai écoutés plus qu'intensivement pendant la rédaction de ce manuscrit.

Je remercie tous mes étudiants, parce qu'enseigner fût une expérience particulièrement gratifiante.

Table des matières

Introduction

Réseaux sociaux numériques

Les réseaux sociaux numériques sont définis par Boyd et Ellison [15] comme des services en ligne qui permettent aux individus de :

1. Construire un profil public ou semi-public ;
2. Définir une liste d'utilisateurs auxquels ils sont connectés ;
3. Voir et naviguer à travers leurs connexions mais aussi celles des autres utilisateurs.

Les vingt dernières années ont été marquées par l'apparition et la popularisation de ces services de réseaux sociaux en ligne (Figure A.1). Depuis l'apparition de **SixDegrees**, un grand nombre de réseaux en tous genres ont émergé. Alors que **Dogster** fait la part belle aux amoureux de nos fidèles compagnons canins, **Piczo** cible les adolescents et **Zaadz** permet de connecter les gens qui souhaitent un monde et une vie plus sains.

Parmi les plus utilisés et étudiés se trouvent **Facebook** [108, 120], **Google+** [44] et **LiveJournal** [85, 127] qui sont dédiés au partage de tous types de contenus, **LinkedIn** [108] destiné à la construction d'un réseau social professionnel, **Youtube** [85, 127] et **Vine** spécialisés dans le partage de contenus vidéos, et enfin **Flickr** [85, 127], **Pinterest** et **Instagram** (voir Figure 0.1) où les utilisateurs mettent en ligne leurs photographies.

Tous ces réseaux sociaux numériques ont prouvé leur succès avec des chiffres impressionnants :

— **Youtube** a dépassé en 2010 deux milliards de vues chaque jour pour les vidéos que le site héberge [113] ;

FIGURE 0.1 – Quelques uns des réseaux sociaux les plus utilisés.

— En octobre 2010, l'ensemble des chaînes **Youtube** a dépassé le milliard d'abonnements [9] ;
— En décembre 2014, **Instagram** annonce avoir dépassé 300 millions d'utilisateurs actifs. Instagram déclare également que 30 milliards de photos ont été partagées. Par ailleurs, le service constate 2,5 milliards de "J'aime" chaque jour [86] ;
— En janvier 2015, on dénombre 359 millions d'utilisateurs actifs sur **Google+** [87] ;
— En 2015, 90 des 332 millions d'inscrits sont actifs sur leur compte **LinkedIn** [87] et **Viadeo** recense 65 millions d'utilisateurs [87] ;
— Le réseau **Pinterest** recense quant à lui 50 millions d'utilisateurs en début d'année 2015.

Nous n'avons pour le moment pas évoqué **Twitter** qui est également l'un de ces services : **Twitter** inclut en effet de nombreux outils destinés à l'échange entre utilisateurs. Dans la mesure où une grande partie des travaux décrits dans ce manuscrit repose sur des données issues de ce dernier, nous allons maintenant en faire une présentation détaillée.

Twitter. Ce service en ligne est un outil de *micro-blogging* qui permet de discuter publiquement des évènements médiatiques et du quotidien [54] en utilisant des *tweets*, courts messages de maximum 140 caractères. Pour voir les *tweets* postés par d'autres utilisateurs s'afficher sur son fil d'actualité (*timeline* en anglais, voir Figure 3.5), il est nécessaire de s'*abonner* à ces derniers. Si l'utilisateur u s'abonne à l'utilisateur v, on dit que u *suit* v, et également 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* les *tweets* d'autres utilisateurs [110], par exemple pour partager avec ses abonnés une information qu'il trouve pertinente, marquer son accord ou engager une forme de conversation [14].

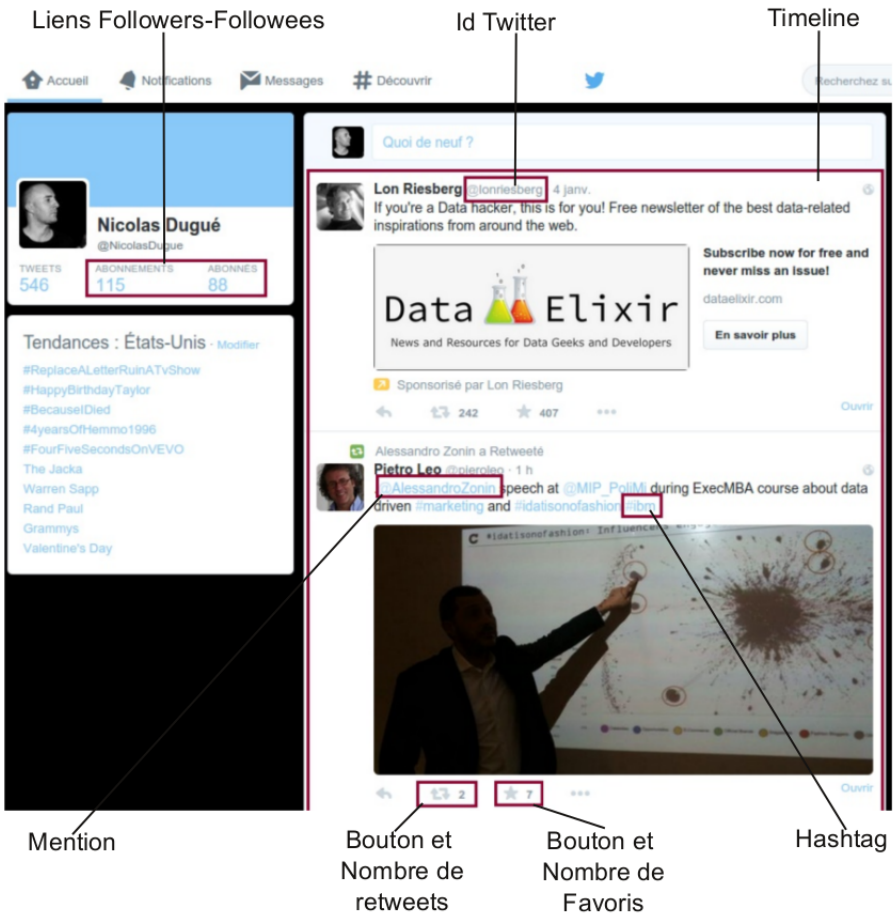

FIGURE 0.2 – L'écran classique de l'application web Twitter.

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. Ainsi par exemple, d'après l'entreprise *Simply Measured* [83], 70% des tweets postés par les 100 plus grandes compagnies (classement Interbrand [52]) sont des réponses à un tweet.

Nous venons ainsi de voir les fonctionnalités sociales inclues dans **Twitter** : le système d'abonnements qui connectent les utilisateurs, et les mentions et retweets qui créent des connexions le temps d'un tweet. Par ailleurs, les *hashtags* sont des mots-clé débutant par le symbole # qui permettent de *taguer* un tweet. En cliquant sur un hashtag, on peut accéder à une page qui recense les tweets tagués avec ce même mot-clé (voir Figure 0.3).

FIGURE 0.3 – En haut en rouge, on voit la distinction entre les "Top" messages postés sur le hashtag *#twitter* et ceux affichés dans la section "Tout". Par défaut, c'est la page "Top" qui est affichée.

On peut voir sur cette Figure qu'une liste des *top* tweets liés au *hashtag* est visible ainsi qu'une liste qui les contient tous dans l'ordre chronologique. Cela permet d'avoir une classification des tweets par mots-clé et de naviguer dans l'information plus efficacement.

Le nombre d'utilisateurs de **Twitter** est passé de 200 millions en avril 2011 [97] à 500 millions en octobre 2012 [111]. D'après Myers et al. [89], en 2012, les 175 millions d'utilisateurs actifs sont connectés par près de 20 milliards d'arcs. En 2015, **Twitter** recense 284 millions d'utilisateurs actifs mensuels. **Twitter** a donc vécu une forte croissance et est actuellement un outil très largement utilisé. La quantité d'information qui y est échangée est réellement importante : un milliard de tweets sont postés tous les deux jours et demi [99]. Le service a ainsi attiré l'attention des politiciens, entreprises, célébrités et spécialistes du marketing. En effet, **Twitter** est vu comme un outil au potentiel considérable pour diffuser l'information. **Barack Obama** s'en est emparé lors de sa campagne présidentielle [50, 40] et des célébrités comme **Britney Spears** [40] ou certaines organisations [18] utilisent également l'outil pour leur communication. D'après *Simply Measured* [83], 94% des 100 plus grandes compagnies (classement Interbrand [52]) ont tweeté au moins une fois par jour, et 75% de celles-ci le font au moins trois fois.

Des utilisateurs malicieux (tels que les spammeurs [123], qui voient **Twitter** comme une mine d'or pour diffuser leurs messages) font également leur apparition sur le réseau. Ces utilisateurs essayent d'être aussi visibles, voire influents que possible sur le réseau [4]. Pour être visible sur **Twitter**, il s'agit d'avoir un grand nombre d'abonnés, d'apparaître dans le

top des résultats du moteur de recherche de **Twitter** ou dans le top des messages pour un *hashtag* donné (Figure 0.3). Le **Larousse** définit ainsi l'influence :

> *Pouvoir social et politique de quelqu'un, d'un groupe, qui leur permet d'agir sur le cours des événements, des décisions prises.*

Pour les spammeurs, être influent peut ainsi signifier réussir à faire efficacement la publicité de leurs produits (pornographie, médicaments, services, contrefaçons, produits illicites), inciter leur audience à cliquer sur les liens hypertextes contenus dans leurs publicités, parvenir à faire de l'hameçonnage. Il est intéressant de noter que la notion d'influence constitue un réel enjeu académique et industriel, de nombreux outils (**Klout** [56], **Kred** [62]) ayant récemment été développés afin de permettre aux utilisateurs de mesurer (voire d'améliorer) leur influence.

Les spammeurs sur **Twitter** ont été intensivement étudiés au cours des années précédentes notamment par Lee et al. [68, 69, 70, 71]. Les auteurs proposent principalement des approches basées sur la récolte de données sur ces profils, puis sur la construction d'un modèle capable de séparer les spammeurs des utilisateurs réguliers à l'aide d'un algorithme de classification supervisée.

De nombreux utilisateurs de **Twitter** (dont les spammeurs) détournent les fonctionnalités sociales offertes par le réseau dans le but de maximiser leur *capital social*. Dans son article *Le Capital social* [13], le sociologue Bourdieu définit le *capital social* comme :

> *L'ensemble des ressources actuelles ou potentielles qui sont liées à la possession d'un réseau durable de relations plus ou moins institutionnalisées d'interconnaissance et d'interreconnaissance.*

Sur **Twitter**, les relations d'abonnements, de mentions et de retweets créent un *réseau de relations* pour chaque utilisateur. Les *ressources* liées à la possession d'un tel réseau sont les possibilités offertes par les liens créés sur le réseau : l'obtention d'informations pertinentes de ses abonnements, la possibilité d'être lu, d'assouvir un besoin narcissique [28, 82] (Figure 0.4), de diffuser efficacement des messages, informations ou publicités. Pour maximiser leur capital social, ces utilisateurs cherchent avant tout à maximiser la taille de leur réseau social et donc à acquérir un maximum d'abonnés. Non seulement cela a pour effet de naturellement augmenter leur visibilité, mais cela rend leurs tweets plus visibles sur le moteur de recherche de **Twitter** [40, 126].

Ces utilisateurs particuliers du média social sont appellés *amis infiltrés* [69] ou *capitalistes sociaux* [40]. Ce sont sur ces derniers que vont principalement porter les travaux présentés dans ce manuscrit. Nous allons donc maintenant réaliser une mise en situation détaillée de ces utilisateurs, en définissant notamment les différentes techniques qu'ils utilisent pour

FIGURE 0.4 – Le Yeti, un capitaliste social raté : 0 abonné.

détourner les fonctionnalités sociales proposées par **Twitter**, et ainsi accroître leur capital social.

Les capitalistes sociaux

Les capitalistes sociaux ont tout d'abord été mis en lumière par Ghosh et al. [40] pendant une étude sur l'échange de liens sur **Twitter** concentrée sur les spammeurs. Entre autres résultats, les auteurs observent avec surprise que les utilisateurs qui réagissent le plus aux sollicitations des spammeurs ne sont ni des robots, ni des spammeurs, mais des utilisateurs *réels*. Ceux-ci mettent en effet à jour leur image de profil et le contenu tweeté n'est pas considéré comme du spam. S'il est surprenant de prime abord que de tels utilisateurs soient des comptes réels et non des robots, cela peut être expliqué par la présence de capitalistes sociaux sur le réseau. Ces utilisateurs cherchent à augmenter à tout prix leur nombre d'abonnés et ainsi à maximiser leur *capital social*. Pour cela, ils mettent en place deux techniques très simples basées sur l'échange d'abonnement réciproque comme illustré sur la Figure 0.5 :

Définition 0.1 (Follow Me and I Follow You (FMIFY))**.** *L'utilisateur assure à ses abonnés qu'il s'y abonnera en retour.*

Définition 0.2 (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 ainsi à d'autres utilisateurs sans considération pour le contenu de leurs tweets. Cela conduit donc tout naturellement à ce qu'ils répondent plus aux sollicitations

FIGURE 0.5 – L'ensemble *in* symbolise les abonnés d'un compte et *out* ses abonnements. A gauche, les utilisateurs appliquant **FMIFY** avec des abonnés qui attendent un abonnement en retour. À droite, les utilisateurs qui appliquent **IFYFM** et qui attendent des abonnements de leurs abonnés.

des spammeurs que des utilisateurs dits *réguliers* ne le feraient. Afin d'être clairs, nous précisons que sous l'appellation *capitalistes sociaux*, nous regroupons tous les utilisateurs qui cherchent à maximiser leur capital social par l'augmentation de leur nombre d'abonnés : les spammeurs peuvent donc être considérés comme capitalistes sociaux.

Méthodes appliquées. Les capitalistes sociaux ont souvent recours à des techniques d'abonnements et désabonnements *agressives*. Bien que proscrites par **Twitter** [117], ces dernières sont pratiquées efficacement par les capitalistes sociaux appliquant le principe **IFYFM**. Ceci est particulièrement visible sur la Figure 0.6. Le processus est le suivant :

1. Abonnement en masse jusqu'à ce que la limite autorisée par Twitter sur le ratio Abonnements / Abonnés [1] soit atteinte [116] ;
2. Les utilisateurs répondent ou non aux abonnements en s'abonnant en retour ;
3. Désabonnement massif afin de pouvoir reprendre la phase 1.

Pour gérer plus facilement des comptes **Twitter**, certains outils comme **HootSuite** et **TweetDeck** existent. Leurs interfaces graphiques et fonctionnalités simplifient la navigation dans ses abonnés, abonnements et contenus tweetés. Néanmoins, avec l'apparition des pratiques d'abonnements et de désabonnement *agressives* telles que celles utilisées par les capitalistes sociaux (**IFYFM**), d'autres outils sont apparus comme Unfollowers.me ou justunfollow.com permettant de faire du désabonnement massif. Certains outils de gestion de comptes **Twitter** comme **Manage Flitter** ont également intégré la fonctionnalité de désabonnement massif.

1. Lorsqu'un compte atteint 2000 abonnements, il existe des restrictions sur son ratio Abonnements/Abonnés. Bien que non explicitées par Twitter [118], si celui-ci est trop élevé, l'utilisateur ne peut plus s'abonner.

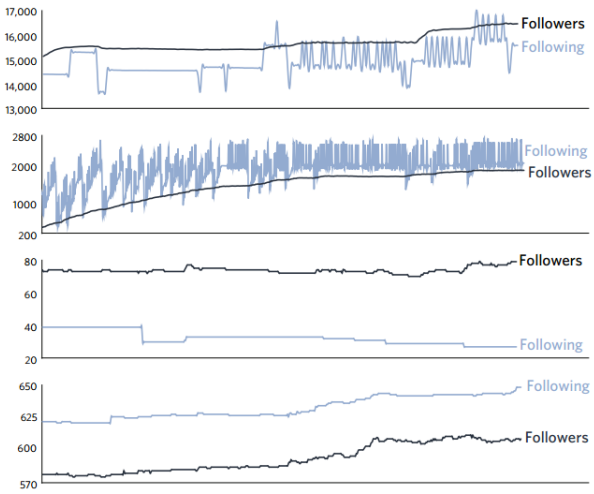

FIGURE 0.6 – Figure issue de Lee et al. [70]. Les deux courbes du haut représentent l'évolution du nombre d'abonnés et d'abonnements de spammeurs utilisant des méthodes de capitalisme social ; celles du bas représentent la même évolution pour des utilisateurs réguliers.

Certains comptes affichent littéralement leur pratique du *capitalisme social* à travers leurs images de profil ou encore leurs noms d'utilisateurs comme le montre la Figure 0.7. Comme on peut le lire sur le compte visible à gauche sur la Figure 0.7, le commerce d'abonnés **Twitter** est devenu monnaie courante : *DM for info & prices*, DM signifiant *message direct*.

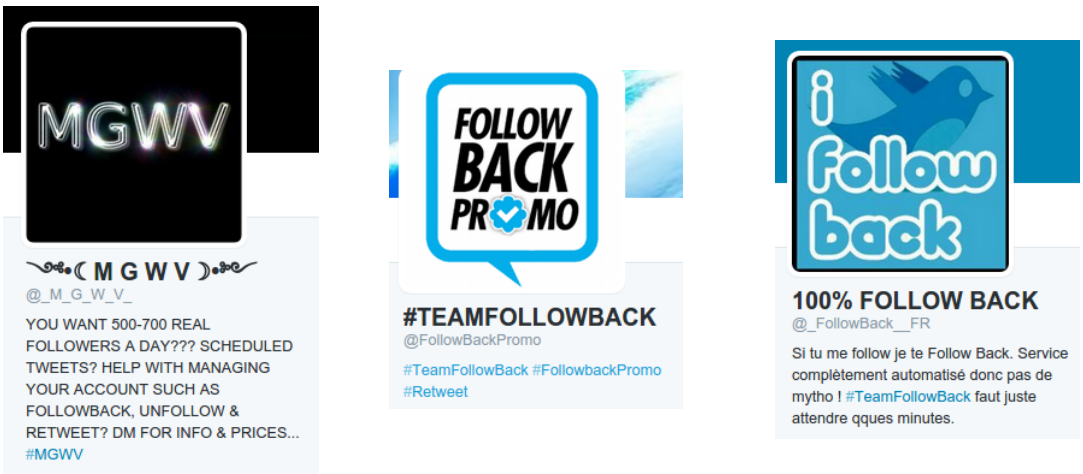

FIGURE 0.7 – Photos de profil, screen name et user names explicites.

Certains comptes **Twitter** sont même dédiés à la promotion de services de vente d'abonnés (Figure 0.8).

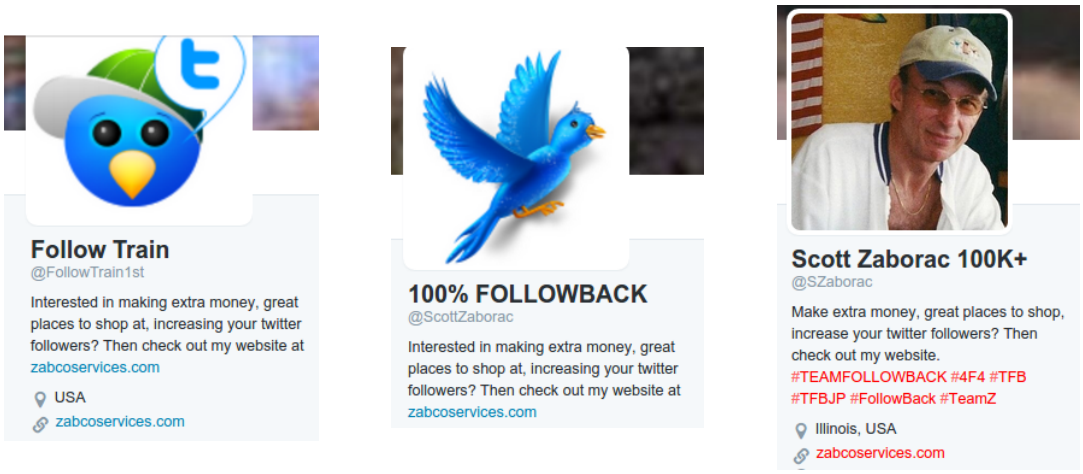

FIGURE 0.8 – Profils qui font la promotion de zabcoservices.com en proposant la possibilité d'augmenter son nombre d'abonnés.

On constate ainsi l'existence de services webs complètement dédiés à l'achat d'abonnés **Twitter** dont voici une liste non exhaustive :

- http://twitterstar.com/buy-twitter-retweets/
- http://www.paywithatweet.com/
- http://followers-and-likes.com/twitter/buy-twitter-retweets/
- http://retweets.pro/buy-twitter-tweets
- http://www.cittadiniditwitter.com/
- https://fastfollowerz.co/
- http://buyfollowersguide.com/
- https://buyaccs.com/en/buy-bulk-twitter-accounts.php
- http://www.retweets.pro/
- http://howdoigetfollowers.net/

Certains services sont également gratuits, comme celui proposé par le site teamfollowback.fr, qui assure à ses utilisateurs de gagner des followers, mais également des *shoutout* (fait de tweeter à ses abonnés de s'abonner à un compte précis). Obtenir des *shoutout* permet donc d'obtenir des abonnements.

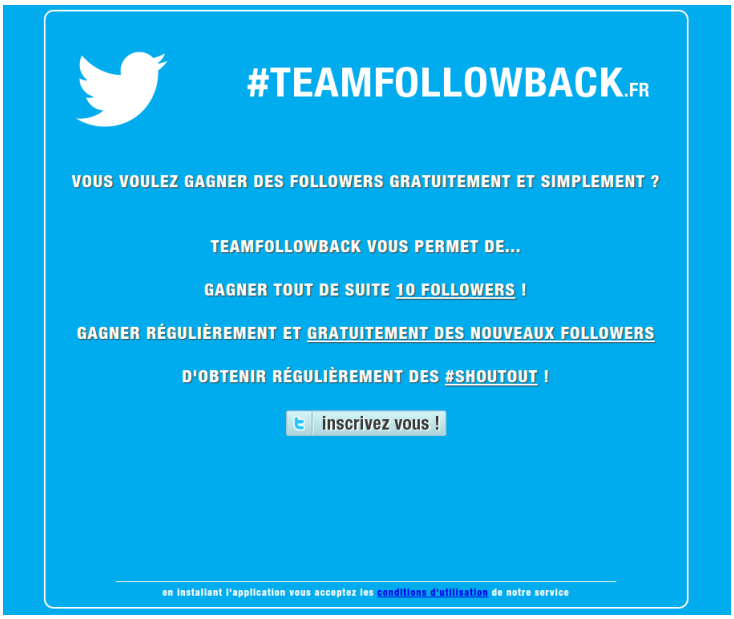

FIGURE 0.9 – Gagner des abonnés gratuitement avec teamfollowback.fr.

Afin d'illustrer l'efficacité de ces méthodes, nous donnons quelques exemples de capitalistes sociaux bien connus identifiés par Ghosh et al. [40] tels que **Barack Obama**, **Britney Spears** ou **JetBlue Airways**. D'autres utilisateurs ont été ajoutés à cette liste en fonction de leur *screen name* (utilisé pour identifier le compte lorsqu'il est précédé d'une arobase) ou parce que le nom du compte est explicitement associé au capitalisme social tel que *TFBJP* or *iFollowBack*. En effet, *TFBJP* sont les initiales de *Team Follow Back Japan* et *iFollowBack* signifie littéralement *Je m'abonne en retour*. On constate que leur nombre d'abonnés est très élevé.

screen name	**name**	**Abonnés**	**Abonnements**
IFOLLOWBACKJP	TFBJP	$1.2 \cdot 10^5$	$1.1 \cdot 10^5$
itsrealchris	iFollowBack	$1.7 \cdot 10^5$	$1.6 \cdot 10^5$
AllFollowMax	TFBJP	$4.2 \cdot 10^4$	$4.3 \cdot 10^4$
BarackObama	Barack Obama	$2.5 \cdot 10^7$	$6.7 \cdot 10^5$
britneyspears	Britney Spears	$2.2 \cdot 10^7$	$4.1 \cdot 10^5$
JetBlue	JetBlue Airways	$1.7 \cdot 10^6$	$1.1 \cdot 10^5$
Starbucks	Starbucks Coffee	$3.2 \cdot 10^6$	$7.9 \cdot 10^4$

TABLE 0.1 – Capitalistes sociaux bien connus. Les nombres d'abonnements et d'abonnés sont arrondis.

Travaux réalisés durant la thèse

Les travaux réalisés durant la thèse concernent principalement l'étude des capitalistes sociaux sur le réseau **Twitter**. Le comportement particulier de ces utilisateurs (qui utilisent des techniques d'abonnements massifs complètement indépendantes du contenu posté) soulève en effet de nombreuses questions. Dans un premier temps, il semble important de se demander si ces utilisateurs peuvent être détectés de manière efficace. Ensuite, il est intéressant d'analyser et de comprendre leur positionnement sur **Twitter** : la visibilité qu'ils acquièrent en utilisant les techniques **IFYFM** et **FMIFY** leur permet-elle de réellement être vus sur le réseau ? Et de manière similaire, les capitalistes sociaux qui arrivent à être visibles sur le réseau via cette méthode sont-ils influents pour autant ?

Ce sont autour de ces problématiques que sera articulé le manuscrit. Nous tenons également à préciser que l'objectif de ces travaux était de pouvoir être applicables en utilisant des ressources *raisonnables*, aussi bien en termes de matériel que de délai d'application. En ce qui concerne le premier point, nous avons réalisé toutes nos expérimentations sur une même machine qui dispose de *64* Go de mémoire vive et de *48 cœurs*. De plus, nous avons essayé de limiter au maximum la récupération d'informations sur les comptes **Twitter** que nous voulions étudier. Ce dernier point a bien entendu été inévitable pour effectuer certaines observations, mais nous avons toujours essayé de récupérer ces informations dans des délais raisonnables. Nous présentons maintenant la structure du manuscrit.

Chapitre 1 - Détection, organisation et évolution des capitalistes sociaux sur Twitter. Ghosh et al. [40] fournissent une liste de 100.000 capitalistes sociaux. Cette liste a été établie en choisissant les 100.000 utilisateurs réels ayant le plus d'abonnements à une liste de spammeurs que les auteurs étudiaient. Cette liste va nous fournir un point d'ancrage pour débuter notre étude des capitalistes sociaux. Nous l'utiliserons notamment dans le prochain chapitre où nous nous intéresserons à la détection de ces capitalistes sociaux. Il nous semble en effet important de disposer d'une méthode plus efficace, plus reproductible que celle de Ghosh et al. [40] afin de pouvoir étudier les capitalistes sociaux. Établir une liste de spammeurs et étudier les comptes d'utilisateurs réels qui y sont abonnés constituent en effet des tâches difficiles. Nous allons donc dans un premier temps définir une méthode de détection naïve mais rapide. Nous étudierons sa précision en utilisant la liste de Ghosh et al. [40] comme vérité de terrain. Nous présenterons par ailleurs une méthode plus efficace dans le Chapitre 3 mais qui nécessite une grande quantité de données différentes. Celle-ci n'est pas applicable à large échelle et ne peut donc être utilisée pour une étude des capitalistes sociaux à un niveau macroscopique du réseau. Nous étudierons ensuite les hashtags dédiés à l'application des méthodes de capitalisme social tels que *#TeamFollowBack*. Nous verrons notamment que les capitalistes sociaux postent -très majoritairement manuellement- des tweets contenant une large variété de hashtags et des sollicitations au retweet et à l'abonnement. Nous montrerons ensuite qu'en utilisant ces hashtags

et les méthodes **IFYFM** et **FMIFY**, il est possible d'acquérir un grand nombre d'abonnés et de retweets comme l'a fait notre compte automatisé *@Rain_bow_ash*. Par ailleurs, nous mettrons en évidence le fait qu'il existe une forme d'*attachement préférentiel* entre capitalistes sociaux. En effet, ces derniers utilisent les mêmes hashtags pour appliquer leurs techniques. Par ailleurs, pour des capitalistes sociaux de type **IFYFM**, les utilisateurs les plus pertinents sont les capitalistes sociaux de type **FMIFY**. Ainsi, nous montrons d'abord que le *coefficient de clustering* des capitalistes sociaux est plus élevé en moyenne que celui des autres utilisateurs et que leurs voisins sont majoritairement des capitalistes sociaux. Finalement, nous étudierons l'état en 2013 de comptes détectés sur un réseau d'abonnés-abonnements en 2009 afin d'étudier l'évolution des capitalistes sociaux.

Chapitre 2 - Rôles communautaires. Nous nous intéressons dans ce Chapitre à la visibilité des capitalistes sociaux. Pour cela, nous étudions leur position dans le réseau en nous plaçant au niveau de la structure de communautés. Nous développons ainsi une méthode capable de catégoriser les nœuds d'un vaste réseau en différentes classes illustrant la position de ces nœuds. Nous appliquons cette méthode aux capitalistes sociaux et montrons que les positions qu'ils occupent prouvent leur visibilité dans le réseau.

Chapitre 3 - Mesures d'influence : le cas des capitalistes sociaux. Le Chapitre 2 démontre la visibilité des capitalistes sociaux sur le réseau **Twitter**. Il s'agit dans ce Chapitre d'observer si cette visibilité se traduit en influence. Nous montrons que les outils de mesure d'influence communément utilisés considèrent la majorité de ces utilisateurs comme influents. Nous mettons donc en place une nouvelle méthode capable de détecter plus efficacement les capitalistes sociaux qui utilise des données décrivant les comptes **Twitter** des capitalistes sociaux et d'utilisateurs réguliers. À partir de cette méthode, nous élaborons une nouvelle mesure de l'influence. Ces travaux sont implémentés dans une application web que nous décrirons également.

Chapitre 4 - Sur l'extraction de sous-graphes denses. Le Chapitre 1 nous montre qu'il existe un attachement préférentiel entre les capitalistes sociaux et que ceux-ci sont donc particulièrement connectés entre eux. Partant de ce constat, nous appliquons plusieurs méthodes de *détection de communautés* et observons que les méthodes existantes ne regroupent pas les capitalistes sociaux en communautés. Ceci nous conduit à reconsidérer notre approche : les capitalistes sociaux ne forment peut-être pas des communautés au sens strict mais des *sous-graphes denses*. Nous présentons donc une étude préliminaire de l'application de la méthode des *coeurs de communautés* à la tâche d'extraction de *sous-graphes denses*.

CHAPITRE

1 Détection, organisation et évolution des capitalistes sociaux sur Twitter

Dans ce Chapitre, nous commençons par introduire une méthode de détection des capitalistes sociaux sur une capture du réseau d'abonnements entre utilisateurs de **Twitter** réalisé en 2009 [20]. Nous nous basons pour cela sur la liste de 100.000 capitalistes sociaux établie par Ghosh et al. [40]. Cette méthode de détection est basée uniquement sur les liens d'abonnements entre utilisateurs. Elle se calcule donc vite sur un réseau complet et ne nécessite aucune donnée supplémentaire. Nous présentons ensuite une étude des hashtags dédiés au capitalisme social. Pour ce faire, nous collectons un grand nombre de tweets tagués avec le hashtag *#TeamFollowBack*. Nous montrons que de nombreux capitalistes sociaux appliquent leurs techniques de façon manuelle alors qu'il est possible d'automatiser ces pratiques. À l'aide d'un *compte Twitter automatisé*, nous montrerons d'ailleurs l'efficacité de ces méthodes lorsqu'elles sont *informatisées*. Par ailleurs, nous nous intéresserons à la connectivité des capitalistes sociaux. Nous étudierons également les relations existant entre capitalistes sociaux, et illustrerons le fait que leur voisinage est essentiellement constitué de capitalistes sociaux. De plus, leur *coefficient de clustering* est supérieur à celui des autres utilisateurs de même degré.

Définition 1.1 (Coefficient de clustering)**.** *Soit $G = (V, E)$ un réseau non orienté, et $v \in V$ un sommet de ce réseau. Le coefficient de clustering cc_v d'un sommet v est défini comme suit :*

$$cc_v = \frac{2 \cdot n_{edges}}{d(v)(d(v)-1)}$$

où n_{edges} représente le nombre d'arêtes entre les voisins de v et d_v le nombre de voisins de v.

Pour conclure, nous étudierons l'évolution des capitalistes sociaux entre 2009 et 2013 et montrerons ainsi qu'un grand nombre de ces utilisateurs a employé les méthodes de capitalisme social avec succès.

1.1 Détection naïve des capitalistes sociaux sur Twitter

Afin d'étudier les capitalistes sociaux, nous souhaitons tout d'abord introduire une méthode capable de les détecter de façon rapide sur le réseau abonné-abonnement de **Twitter** fourni par Cha et al. [20]. Afin de décrire le réseau abonné-abonnement qui nous sert à modéliser **Twitter**, nous introduisons ici quelques notations.

Définition 1.2 (Réseau orienté)**.** *Un réseau orienté est une paire $D = (V, A)$ où V représente l'ensemble des sommets de ce réseau et A les arcs orientés entre les sommets de V.*

Définition 1.3 (Voisinage)**.** *Le voisinage sortant (resp. entrant) d'un noeud $v \in V$ est l'ensemble $N^+(v) = \{u \in V : (v,u) \in A\}$ (resp. $N^-(v) = \{u \in V : (u,v) \in A\}$).*

Définition 1.4 (Degré)**.** *Le degré sortant d'un noeud $v \in V$, noté $d^+(v)$, correspond à la taille $|N^+(v)|$ de son voisinage sortant. De manière similaire, le degré entrant d'un noeud est la valeur $d^-(v) = |N^-(v)|$. Le degré $d(v)$ d'un noeud est égal à la somme de ses degrés entrant et sortant.*

Le réseau orienté recueilli par Cha et al. [20] en 2009 contient environ 50 millions (52.579.682) d'utilisateurs et les 2 milliards (1.963.263.821) d'abonnements qui les connectent. Dans ce réseau abonné-abonnement orienté, les utilisateurs de **Twitter** sont représentés par les sommets du réseau. Les arcs orientés du réseau constituent les liens d'abonnement entre utilisateurs de type u suit v. L'ensemble $N^+(v)$ représente donc les abonnements de l'utilisateur v et $N^-(v)$ ses abonnés. La moyenne des degrés des noeuds du réseau est d'environ 75 et le degré maximal dans le réseau est de 3.691.240. Le taux de réciprocité, c'est à dire le pourcentage d'arcs $(u,v) \in A$ du réseau tels que $(v,u) \in A$ est d'environ 10%.

1.1.1 Mesures de similarités

Nous allons détecter les capitalistes sociaux sur ce réseau, justement en nous intéressant aux liens d'abonnements réciproques. En effet, pour détecter les capitalistes sociaux sur ce réseau, nous utilisons l'intuition que les comportements **FMIFY** et **IFYFM** (voir Définitions 0.1 et 0.2) engendrent une intersection élevée entre l'ensemble des abonnés et des abonnements d'un utilisateur. Pour quantifier à quel point cette intersection est élevée, nous utilisons une première mesure de similarité : l'indice de chevauchement introduit par Simpson [107].

Définition 1.5 (Indice de chevauchement)**.** *Étant donnés deux ensembles A et B, l'*indice de chevauchement *de A et B qui a valeur dans $[0,1]$ est donné par :*

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

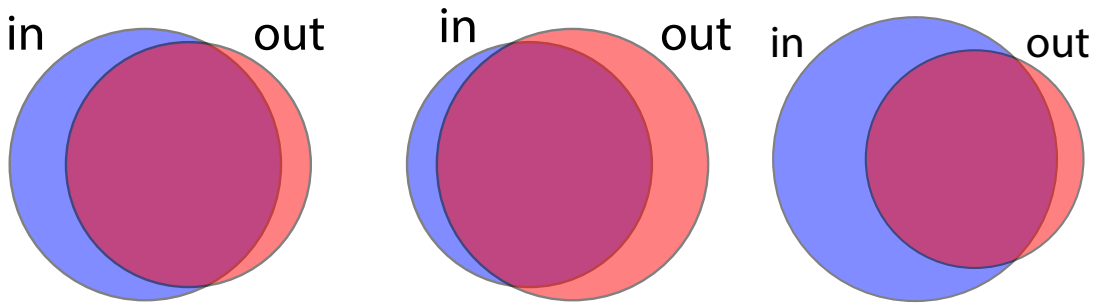

FIGURE 1.1 – Les trois classes de capitalistes sociaux. L'ensemble *in* symbolise les abonnés d'un compte et *out* ses abonnements. A gauche, les utilisateurs appliquant **FMIFY** avec des abonnés qui attendent un abonnement en retour. Au milieu, les utilisateurs qui appliquent **IFYFM** et qui attendent des abonnements de ceux à qui ils sont abonnés. Enfin, à droite les capitalistes sociaux passifs.

Pour chaque sommet v de notre réseau Twitter, nous appliquons l'indice de chevauchement sur les ensembles $A = N^+(v)$ et $B = N^-(v)$. Si l'indice est suffisamment proche de 1, nous considérons alors l'utilisateur comme un capitaliste social. Nous préférons cet indice à la mesure de *Jaccard* [53].

Définition 1.6 (Indice de Jaccard)**.** *Étant donnés deux ensembles A et B, l'*indice de Jaccard *de A et B qui a valeur dans* $[0,1]$ *est donné par :*

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

En effet, nous cherchons ici à détecter les utilisateurs ayant appliqué les principes de capitalisme social **FMIFY** et **IFYFM**. Certains utilisateurs comme **Barack Obama** semblent avoir cessé ces pratiques. Leur nombre d'abonnements reste maintenant stable, quand leur nombre d'abonnés continue d'augmenter grâce à leur notoriété bien réelle. Ainsi, pour ces utilisateurs, l'ensemble des abonnés est bien plus grand que celui des abonnements. Utiliser l'union pour diviseur comme dans la mesure de *Jaccard* engendre des résultats proche de 0 tandis que l'indice de chevauchement est proche de 1. Cet indice permet donc de détecter les capitalistes sociaux que nous appelons *passifs* : ils ont cessé d'appliquer les principes **FMIFY** et **IFYFM**.

Nous utilisons ensuite une seconde mesure de similarité pour classifier ces utilisateurs.

Définition 1.7 (Ratio)**.** *Soit $v \in V$ un sommet d'un réseau orienté $D = (V, A)$. Le* ratio *de v est donné par :*

$$R(v) = \frac{|N^+(v)|}{|N^-(v)|}$$

Le ratio nous permet de distinguer les capitalistes sociaux appliquant le principe **IFYFM** de ceux qui utilisent la méthode **FMIFY**. Intuitivement, les capitalistes sociaux qui suivent le principe **IFYFM** ont un nombre d'abonnements plus élevé que leur nombre d'abonnés, et donc un ratio supérieur à 1. C'est le contraire pour les capitalistes sociaux de type **FMIFY** qui ont donc un ratio entre 0 et 1. De même, le ratio nous aide à détecter les capitalistes sociaux dits *passifs*. Ces utilisateurs dont le nombre d'abonnés continue à croître même s'ils ont cessé l'application des méthodes **IFYFM** ou **FMIFY** se retrouvent avec un ratio bien inférieur à 1 (Figure 1.1).

Nous avons donc présenté deux mesures de similarité qui utilisent la topologie locale du réseau pour détecter et classifier les capitalistes sociaux. Bien que basées sur des intuitions simples, nous allons tout de même formaliser leur utilisation. Nous avons notamment établi qu'un utilisateur peut être considéré comme un capitaliste social si son indice de chevauchement est proche de 1. En utilisant la liste de 100.000 capitalistes sociaux de Ghosh et al. [40], nous définissons dans la suite la notion *proche de 1* en fournissant un seuil de détection.

1.1.2 Un seuil pour l'indice de chevauchement

Pour fournir un seuil de détection, nous utilisons ainsi les 100.000 capitalistes sociaux fournis par Ghosh et al. [40] comme vérité de terrain. Ces utilisateurs ont été détectés par les auteurs sur le réseau abonné-abonnement fourni par Cha et al. [20] et c'est donc celui-ci que nous utilisons. Nous calculons ainsi l'indice de chevauchement des 100.000 capitalistes sociaux fournis par Ghosh et al. [40]. À partir de cela, nous traçons la distribution des valeurs de l'indice de chevauchement de ces utilisateurs considérés comme des capitalistes sociaux (voir Figure 1.2).

Nous observons que très peu d'utilisateurs ont un indice de chevauchement inférieur à $0,6$. Ils ne semblent pas utiliser les principes décrits ci-dessus. Au-dessus de $0,6$, on observe une très forte augmentation du nombre d'utilisateurs dans chaque intervalle. On remarque notamment que 80% des utilisateurs recensés dans cette liste ont un indice de chevauchement supérieur à $0,74$. Dans la mesure où nous souhaitons être capable de détecter des capitalistes sociaux en minimisant le nombre de faux positifs, nous utilisons ce seuil plutôt élevé.

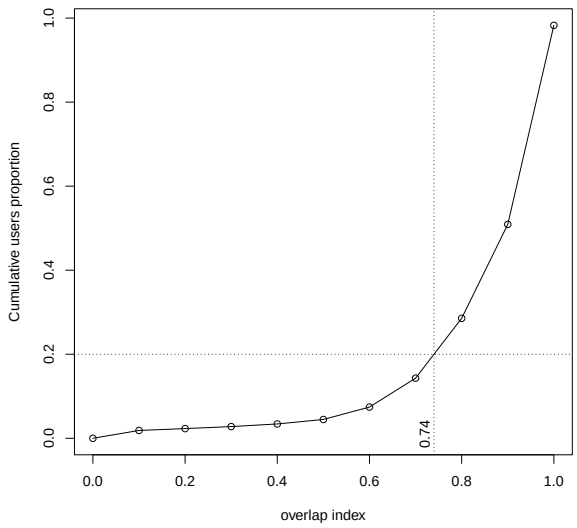

FIGURE 1.2 – Distribution cumulative de l'indice de chevauchement des 100.000 capitalistes sociaux de la liste de Ghosh et al. [40].

Plus tôt dans ce manuscrit, nous introduisions une liste de capitalistes sociaux (voir Table 0.1) identifiés par Ghosh et al. [40] ou évidents. Nous souhaitons étudier leurs *indice de chevauchement* et *ratio*. Pour cela, nous utilisons l'API Twitter [119] afin d'obtenir les informations sur ces utilisateurs, *API* signifiant *Application Programming Interface* (Annexe C). En utilisant nos mesures de similarité et le seuil de 0,74 maintenant établi sur les données récupérées sur ces utilisateurs, nous montrons dans la Table 1.1 que nous détectons effectivement ces utilisateurs.

screen name	**name**	I_c	**ratio**
IFOLLOWBACKJP	TFBJP	0,97	0,92
itsrealchris	iFollowBack	0,81	0,94
AllFollowMax	TFBJP	0,99	1,04
BarackObama	Barack Obama	0,77	0,03
britneyspears	Britney Spears	0,82	0,02
JetBlue	JetBlue Airways	0,74	0,06
Starbucks	Starbucks Coffee	0,77	0,02

TABLE 1.1 – *Indice de chevauchement* (I_c) et *ratio* des capitalistes sociaux de la Table 0.1.

Les deux premiers sont considérés comme appliquant la méthode **FMIFY**, leur ratio étant inférieur à 1. Le troisième est classifié comme utilisant le principe **IFYFM**, son ratio étant

supérieur à 1. Enfin, les quatre derniers ont un ratio très proche de 0, ce qui les range dans la catégorie des capitalistes sociaux **passifs**. Ces comptes **Twitter** de personnalités publiques ou de grandes entreprises obtiennent en effet tout naturellement de plus en plus d'abonnés grâce à leur renommée. Nous verrons Section 1.5 que ces comptes ont pratiqué des méthodes d'abonnements agressives.

Nous avons donc maintenant un seuil qui nous permet de détecter les capitalistes sociaux dans n'importe quel réseau **Twitter**. Nous appliquons notre méthode de détection et de classification sur le réseau d'abonné-abonnement **Twitter** complet fourni par Cha et al [20]. Nous calculons ainsi l'indice de chevauchement et le ratio de chacun des 55 millions de sommets du réseau. Par ailleurs, nous introduisons d'autres contraintes. La première est le nombre d'abonnés. Nous nous intéressons aux capitalistes sociaux qui ont effectivement gagné des abonnés en appliquant les principes décrits dans la Section 1.1. Nous considérons ainsi des utilisateurs avec au moins 500 abonnés. De plus, afin d'éviter de détecter des utilisateurs avec un indice de chevauchement au dessus du seuil de $0,74$ mais avec un très petit nombre d'abonnements, nous plaçons aussi une contrainte sur ce dernier. En effet, des utilisateurs avec très peu d'abonnements et un grand nombre d'abonnés sont susceptibles d'avoir un indice de chevauchement proche de 1. Néanmoins, cela ne dénote pas un comportement de capitaliste social. Ainsi, pour résumer, notre méthode de détection considère les utilisateurs ayant des nombres d'abonnements et d'abonnés supérieurs à 500 ainsi qu'un indice de chevauchement supérieur à $0,74$. Avec ces contraintes, nous nous assurons au maximum de considérer uniquement des capitalistes sociaux, et non simplement des utilisateurs dont les amis, la famille et les collègues se suivent mutuellement.

Nous détectons un peu plus de 160.000 capitalistes sociaux sur ce réseau, ce qui est un peu plus de 55% du nombre d'utilisateurs avec plus de 500 abonnés et plus de 500 abonnements (voir Table 1.2).

Nous classifions ensuite ces utilisateurs selon leur ratio. La plupart des utilisateurs avec un nombre d'abonnés supérieur à 500 ont un ratio supérieur à 1. Si l'on ne considère que les utilisateurs avec un nombre d'abonnés supérieur à 10.000, on observe 9% de capitalistes sociaux dits *passifs*, dont le ratio est en moyenne autour de $0,25$, ce qui signifie qu'ils ont en moyenne quatre fois plus d'abonnés que d'abonnements.

abonnés	**abonnements**	**sommets**	**ratio**	%
> 500	> 500	161.424		
			> 1	68
			$[0,7;1]$	25
> 2.000	> 500	47.221		
			> 1	61
			$[0,7;1]$	31
> 10.000	> 500	5.743		
			> 1	66
			$[0,7;1]$	25
			$< 0,7$	9

TABLE 1.2 – Détection des capitalistes sociaux sur le réseau complet.

Nous avons maintenant décrit les capitalistes sociaux, leurs méthodes **FMIFY** et **IFYFM** ainsi qu'une approche pour les détecter sur le réseau **Twitter**. Nous allons maintenant voir que les capitalistes sociaux utilisent les *hashtags* pour appliquer plus efficacement leurs méthodes. Nous nous intéressons en particulier au hashtag *#TeamFollowBack*.

1.2 Organisation des capitalistes sociaux

Dans cette Section, nous montrons que certains des capitalistes sociaux mettent en application les principes **FMIFY** et **IFYFM** de façon organisée. On pourrait en effet supposer que les capitalistes sociaux de type **IFYFM** choisissent des cibles comme les abonnés de leurs abonnés, ou bien les utilisateurs suggérés par Twitter. Néanmoins, en étudiant l'activité de ces utilisateurs, nous constatons par exemple qu'un certain nombre d'entre eux sont regroupés en équipes telles que *TeamFollowBackJapan*, dont l'utilisateur *IFOLLOWBACKJP* (voir Table 0.1) semble faire partie. Ces équipes utilisent souvent un hashtag dédié pour appliquer leurs méthodes. Les utilisateurs qui tweetent en utilisant ce hashtag font partie de l'équipe, ou en tout cas sont très susceptibles d'être des capitalistes sociaux. De façon plus générale, il existe de nombreux hashtags dédiés au capitalisme social. Et poster des tweets en utilisant ces hashtags permet de trouver de nouveaux abonnés.

Plusieurs méthodes sont employées par les utilisateurs postant des tweets en utilisant ces hashtags :

— tweeter des messages invitant les autres utilisateurs à s'abonner à leur compte ;

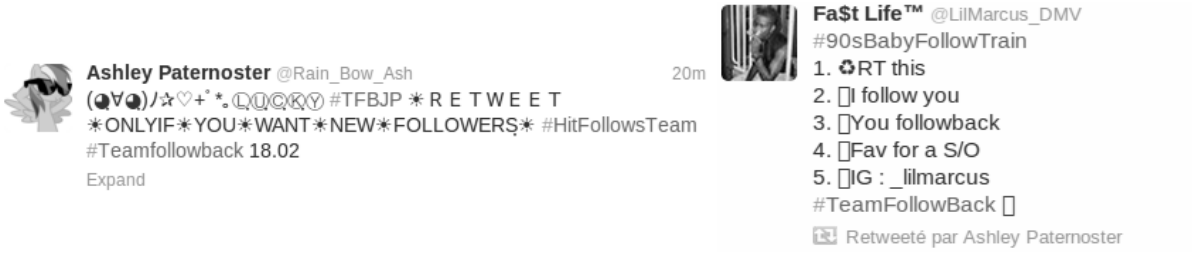

FIGURE 1.3 – Exemples d'un tweet et retweet posté par le bot *@Rain_bow_ash*, créé par nos soins (Section 1.3).

— tweeter des messages invitant les autres utilisateurs souhaitant obtenir des abonnements à les retweeter (voir Figure 1.3) ;

— tweeter des messages mentionnant les utilisateurs dont ils souhaiteraient se voir abonnés ;

— s'abonner aux utilisateurs qui postent ce genre de tweets

Afin d'étudier l'efficacité de ces méthodes, nous avons utilisé (durant le mois de Février 2013) l'API fournie par **Twitter** afin de récupérer les tweets postés sur le hashtag *#TeamFollowBack* (Annexe C). Nous avons ainsi obtenu, en éliminant les tweets redondants, près de 725.000 tweets (Table 1.3). Nous allons maintenant décrire les données obtenues et en tirer des conclusions quant à l'efficacité des méthodes utilisant les hashtags dédiés au capitalisme social.

Type	**Nombre**
Tweets	726.470
Hashtags	4.227.703
Hashtags distincs	25.028
Nombre moyen de hashtags par tweets	5,82
Utilisateurs distincts	124.786
Mentions	719.972
Utilisateurs mentionnés distincts	43.199

TABLE 1.3 – Statistiques des tweets contenant *#TeamFollowBack* et récupérés via l'API **Twitter**.

Hashtags. Comme nous pouvons le voir dans la Table 1.3, les tweets postés sur ce hashtag contiennent en moyenne 6 hashtags. Ce taux très élevé est supérieur à celui observé pour les tweets des spammeurs par Benevuto et al. [7]. L'objectif est sans doute de maximiser la diffusion de ces tweets sur le plus grand nombre de canaux dédiés au capitalisme

social. Par ailleurs, nous avons examiné plus en détail les hashtags utilisés par les capitalistes sociaux dans ces messages. Dans notre jeu de données, nous obtenons 25.028 hashtags distincts (voir Table 1.3). Cependant, seulement certains d'entre eux sont fréquents. En effet, en additionnant le nombre d'occurences des 10 hashtags les plus fréquents, nous obtenons plus de 50% du nombre total d'occurences de hashtags du jeu de données, à savoir 4.227.703. En faisant de même avec les 46 hashtags les plus fréquents (voir Table 1.4), nous obtenons plus de 90% du nombre total d'occurences de hashtags du jeu de données. En résumé, en observant 0,2% des hashtags distincts du jeu de données, nous pouvons observer 90% des occurences du jeu de données.

Hashtag	**Occurences**	**Hashtag**	**Occurences**
TeamFollowBack	766.421	retweet	48.656
TFBJP	339.917	rt2gain	48.008
sougofollow	177.064	teamfollowwack	43.856
500aday	172.655	follow2gain	41.499
OPENFOLLOW	148.304	TFW	40.637
FollowBack	143.174	teamhitfollow	35.405
RT	125.211	Followers	33.546
instantfollowback	107.989	hdyf	28.750
AutoFollow	105.528	thf	27.861
HitFollowsTeam	102.100	INSTANTFOLLOW	27.075
90sBabyFollowTrain	100.742	R_Family	25.293
f4f	100.043	teamfollowwacky	22.495
mustfollow	100.041	Retweets	21.384
FollowNGain	99.967	followmejp	21.340
follow	92.415	Favorites	19.655
tfb	83.820	love	19.162
FF	82.299	tmw	18.087
1000aday	79.657	follow4follow	17.346
teamautofollow	59.670	SiguemeyTeSigo	16.982
autofollowback	53.479	RTRTRT	16.161
IFollowBack	53.032	tfbfollowtrain	15.784
maxvip	51.316	OpenFollowTeam	14.590
followme	49.809	ffback	14.576

TABLE 1.4 – Les 46 hashtags les plus fréquents et leur nombre d'occurences dans le jeu de données.

En observant ces hashtags, on constate que la plupart d'entre eux sont explicitement liés au capitalisme social. Les mots clés *team*, *follow* (souvent abrégé *f* comme dans *TFBJP*), *back*, *retweet* (souvent abrégé *rt* comme dans *rt2gain*) sont présents dans quasiment tous

ces hashtags fréquents du jeu de données. Nous observons également quelques exceptions telles que *love* et *Favorites*, qui sont probablement des hashtags populaires. Benevuto et al. [7] ont observé ce type de comportement chez les spammeurs qui ont tendance à utiliser un grand nombre de hashtags, et certains hashtags populaires comme *#musicmonday* les jours de diffusion de l'émission *Britain's Got Talent*.

Sources. En collectant les tweets contenant le hashtag *#TeamFollowBack*, nous avons également stocké certaines informations supplémentaires fournies par l'API **Twitter**. Ainsi nous avons obtenu ce que **Twitter** appelle les *sources* utilisées par les utilisateurs pour poster leurs tweets. Cette information est très intéressante puisqu'elle peut indiquer si l'utilisateur se sert d'outils pour poster de façon automatique ce genre de tweets dédiés au capitalisme social (voir Figure 1.3).

Parmi les 606 sources différentes utilisées pour poster les tweets du jeu de données, 15 sont particulièrement pertinentes à étudier puisqu'elles ont servi à tweeter 90% des messages comme le montre la Table 1.5.

Sources	**Occurences**
Twitter for BlackBerry	196.911
web	196.747
Twitter for Android	66.473
Twitter for iPhone	49.556
twitterfeed	39.010
Mobile Web (M2)	28.482
BotMaker	14.260
BestFollowers App.2.85	11.189
dlvr.it	10.443
TweetCaster for Android	9.777
Twitter for iPad	9.329
twittbot.net	8.300
UberSocial for BlackBerry	8.037
Write Longer	5.155
twisuke	5.040

TABLE 1.5 – Sources utilisées pour poster 90% des tweets du jeu de données. Les sources qui permettent d'automatiser l'activité d'un compte **Twitter** sont en gras.

Comme nous pouvons le voir sur la Table 1.5 (en gras), certaines de ces sources sont des sites webs ou des applications destinées à automatiser certaines fonctionnalités **Twitter**, et notamment l'action de poster des tweets. Cependant, seulement 12% des tweets recueillis

ont été postés en utilisant ces sources. Ainsi, une grande majorité de ces tweets sont postés en utilisant les applications Twitter pour téléphones ou tablettes qui permettent de gérer un compte **Twitter** de façon plus ergonomique. Cela nous permet donc de conclure que la plupart de ces utilisateurs ne sont ni des robots, ni des comptes automatisés mais bien des utilisateurs réels qui appliquent des techniques de capitalisme social manuellement. Ceci rejoint le constat établi par Ghosh et al. [40] qui observent que les utilisateurs qui suivent le plus les spammeurs et qu'ils considèrent comme des capitalistes sociaux semblent être des utilisateurs réels.

1.2.1 Indice de chevauchement et ratio

Nous considérons que les utilisateurs ayant posté les tweets de notre jeu de données sont des capitalistes sociaux. En effet, ces tweets ont été collectés sur un hashtag dédié à l'utilisation des méthodes de capitalisme social (Figure 1.3). Ainsi, ils nous servent ici à tester la méthode de détection détaillée dans la Section 1.1. Pour tester cette méthode, nous avons récupéré les abonnés et les abonnements de 12% des 124.786 utilisateurs distincts ayant tweeté sur le hashtag *#TeamFollowBack*. Avec ces informations, nous avons calculé l'indice de chevauchement de ces utilisateurs et nous vérifions ainsi qu'ils sont bien détectés par notre méthode.

Type	**Nombre**
Utilisateurs	124.786
Utilisateurs avec N^+ et N^- récupérés	15.226
Utilisateurs avec N^+ et N^- récupérés > 500	8.442
Utilisateurs avec N^+ et N^- récupérés > 500 et $I_c > 0,74$	6.740

TABLE 1.6 – Nombre d'utilisateurs recueillis qui tweetaient avec le hashtag *#TeamFollowBack*. Ici, N^+ (resp. N^-) représente l'ensemble des abonnements (resp. abonnés) et I_c vaut pour *indice de chevauchement*.

Comme nous pouvons le voir dans la Table 1.6, plus de la moitié des utilisateurs ont des nombres d'abonnés et d'abonnements inférieurs à 500. Ceci confirme ce qui a été établi précédemment : un faible nombre de ces utilisateurs automatise leur façon de faire du capitalisme social. Manuellement, ce processus est lent et fastidieux, ce qui explique que beaucoup de ces utilisateurs aient encore peu d'abonnés. Parmi les utilisateurs ayant des nombres d'abonnements et d'abonnés supérieurs à 500, 80% ont un indice de chevauchement supérieur à 0,74. Ce chiffre est donc similaire à celui obtenu lors de l'établissement du seuil de 0,74. En revanche, il existe probablement un grand nombre de capitalistes sociaux ayant moins de 500 abonnés et abonnements et cette méthode ne permet pas de les détecter sans prendre le risque de retourner également un grand nombre de faux positifs, tels que des comptes abonnés mutuellement à leurs amis, collègues et familles. Le faible

rappel de cette méthode montre ainsi ses limites. Néanmoins, cette méthode ne nécessite qu'une faible quantité de données. Par ailleurs, ces données nous sont facilement accessibles puisqu'elles ne dépendent pas de l'API.

1.3 Automatisation du capitalisme social

Rappelons que seulement 12% des tweets de notre jeu de données sont postés de façon automatique. Néanmoins, il semble intéressant de déterminer l'efficacité de ces méthodes. Ainsi, comme le robot créé par Aiello et al. [1] sur le réseau social *aNoobi.com* dédié aux amateurs de livres, nous souhaitons montrer qu'un compte sans informations de profil, qui ne poste aucune information pertinente et dont l'unique but est d'obtenir plus d'abonnés, peut réussir à accroître efficacement son nombre d'abonnés et ainsi sa visibilité sur le réseau. Nous avons donc créé un compte automatisé reproduisant les méthodes décrites dans la Section 1.2. Ce compte est enregistré sous le nom de *@Rain_bow_ash* et il peut être consulté en ligne à l'URL suivante : http://www.twitter.com/Rain_bow_ash.

FIGURE 1.4 – Profil du bot *@Rain_bow_ash* au 31 mars 2015.

Chaque heure, le compte automatisé que nous appelons *bot* poste un tweet que nous choisissons aléatoirement parmi ceux recueillis sur le hashtag *#TeamFollowBack* et qui font partie de notre jeu de données. De plus, le bot retweete chaque heure le dernier message posté sur le hashtag *#90sBabyFollowTrain* (Figure 1.3).

Ensuite, en utilisant l'API Twitter, nous avons créé plusieurs actions simples déclenchées lors des interactions d'autres utilisateurs avec le bot :

— chaque fois que le bot est mentionné, il s'abonne au compte qui l'a mentionné ;

— chaque fois qu'un message du bot est retweeté, il s'abonne au compte qui l'a retweeté ;
— chaque fois qu'un compte s'abonne au bot, il s'abonne au compte en retour : le bot applique le principe **FMIFY**.

Par ailleurs, nous enregistrons chaque action liée au bot : lorsqu'il est mentionné, retweeté, lorsqu'un utilisateur s'y abonne, lorsqu'il s'abonne à un utilisateur, le mentionne ou retweete un de ses messages. Ainsi, nous pouvons suivre et comprendre les interactions entre le bot et les autres utilisateurs.

Nous avons lancé le bot le 30 Mai 2013 et il fut stoppé une semaine pour des raisons matérielles. Depuis Janvier 2014, il n'applique que la dernière fonctionnalité : chaque fois qu'un compte s'abonne au bot, il s'abonne au compte en retour. Par ailleurs, nous ne surveillons plus toutes les actions du bot. Les chiffres qui suivent sont donc les chiffres obtenus le 23 Août 2013. Nous souhaitons mentionner que **Twitter** n'a pas bloqué une seule fois le bot même s'il semble évident que le comportement de ce robot est automatique et que son intérêt pour le réseau social est nul. Le bot existe même toujours, comme le montre la Figure 1.4.

Nous observons sur la Table 1.7 que le bot était abonné le 23 août 2013 à 7.124 utilisateurs et 6.792 utilisateurs y étaient abonnés [1].

Type	**Nombre**
Nombre d'abonnements	7.124
Nombre d'abonnés	6.792
Nombre d'abonnés totaux durant la période	9.319
Nombre d'abonnés suspendus	615
Nombre d'abonnés supprimés	77

TABLE 1.7 – Statistiques à propos des abonnés et abonnements du bot.

Cependant, durant toute la période qu'a duré l'expérimentation, 9.319 utilisateurs différents ont été abonnés au bot. Certains de ces utilisateurs ont cessé de suivre le bot, mais un grand nombre d'utilisateurs qui ne sont plus abonnés au bot ont été supprimés (77) ou sont des comptes suspendus par **Twitter** (615). Plus surprenant, 9.753 abonnements au bot ont été enregistrés. Ceci signifie que certains utilisateurs ont suspendu leur abonnement, puis s'y sont réabonnés. Nous observons ainsi que 324 utilisateurs ont eu ce comportement, l'un d'entre eux s'étant même abonné à notre bot 12 fois. Nous supposons qu'il s'agit d'utilisateurs qui trichent avec les principes **FMIFY** et **IFYFM**. Ces utilisateurs s'abonnent au bot

1. Comme le montre la figure 1.4, ce nombre a aujourd'hui quasiment doublé.

puis s'y désabonnent de façon à conserver un faible ratio (voir Définition 1.7). Conserver un faible ratio offre deux avantages. Premièrement, cela permet de respecter les restrictions de **Twitter** sur le ratio. En effet, quand un compte atteint les 2.000 abonnés, il existe des restrictions sur son ratio qui ne sont pas explicitées clairement par **Twitter** [118], mais si celui-ci est trop élevé, l'utilisateur ne peut plus s'abonner à d'autres utilisateurs. Ainsi, se désabonner de certains utilisateurs peut permettre à ces capitalistes sociaux de contourner les restrictions imposées par **Twitter** et continuer à appliquer le principe **IFYFM**. Deuxièmement, conserver un ratio faible donne une bonne image du compte. Avoir un plus grand nombre d'abonnés que d'abonnements renvoie l'image d'un utilisateur influent.

1.3.1 Comparaison des différentes stratégies utilisées

Être retweeté et mentionné. Comme le montrent les Tables 1.8 et 1.9, les différentes stratégies implémentées par notre bot n'ont pas la même efficacité. Certaines d'entre elles permettent d'obtenir un plus grand nombre d'abonnés plus rapidement.

	utilisateurs	**utilisateurs distincts**	**abonnements**	**pourcentage**
mention	977	646	146	23
retweeet	20.526	5.728	2.893	50

TABLE 1.8 – Proportion des utilisateurs qui s'abonnent au bot après l'avoir retweeté ou mentionné.

Par exemple, les messages du bot sont très retweetés, et lorsque le bot est retweeté par un utilisateur, dans 50% des cas, il s'abonne au bot par la suite. Au contraire, le bot est plutôt mentionné par des utilisateurs qui y sont déjà abonnés. Il semble donc que mentionner un utilisateur sur ces hashtags ne soit pas un moyen largement utilisé pour demander un nouvel abonnement mutuel : ce n'est le cas que dans 23% des mentions.

	utilisateurs	**utilisateurs distincts**	**abonnements**	**pourcentage**
mention	877	234	26	11
retweeet	3.527	1.331	470	35

TABLE 1.9 – Proportion des utilisateurs qui s'abonnent au bot après avoir été mentionnés ou retweetés.

Retweeter et mentionner. Retweeter et surtout mentionner d'autres utilisateurs est beaucoup moins efficace que les méthodes précédentes. Seulement 11% des utilisateurs mentionnés par le bot s'y abonnent par la suite. Cependant, il faut nuancer ce chiffre par le fait que les messages contenant des mentions que le bot tweete sont issus du jeu de données. Certains ne sont donc plus forcément à jour : les utilisateurs peuvent ne plus exister ou encore ne plus prendre part à ce genre de pratiques.

Tweeter. Finalement, il semble que la plus simple des stratégies soit la plus efficace. En effet, simplement tweeter sur le hashtag a permis au bot de gagner 5.784 abonnés, ce qui est un plus grand nombre que celui qui regroupe toutes les autres stratégies (3.535). Nous pouvons affirmer que c'est la visibilité des tweets du bot sur ces hashtags qui a conduit le plus souvent à des gains d'abonnés pour le bot. Pour s'en assurer, nous avons fait en sorte que le bot cesse de tweeter 4 jours et avons gagné beaucoup moins d'abonnés. Ceci confirme que les utilisateurs qui s'abonnent au bot sans qu'il n'y ait de relations avec un retweet ou une mention, s'y abonnent parce qu'ils ont vu ses tweets relativement à l'un des hashtags. Cela exclut la possibilité que ces utilisateurs s'y abonnent par exemple parce que l'un de leurs abonnés est abonné à notre bot ou que **Twitter** leur suggère de s'y abonner.

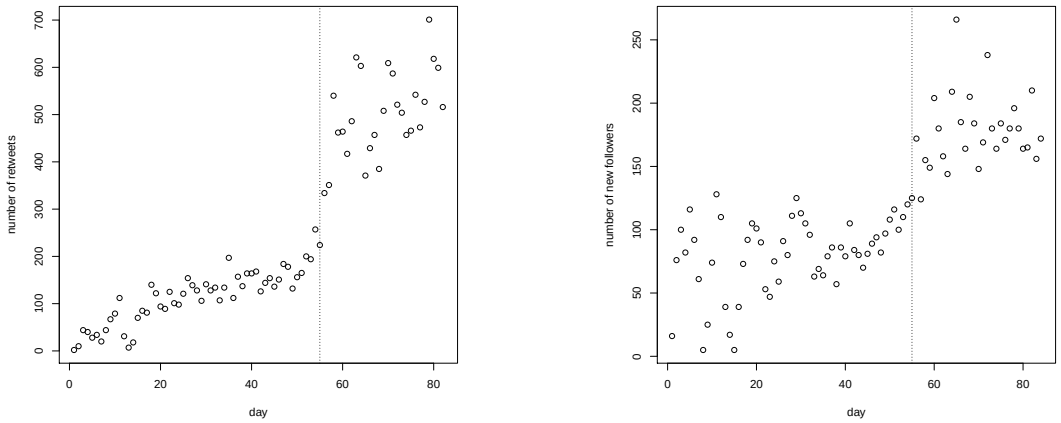

FIGURE 1.5 – Nombre de retweets (à gauche) et nombre de nouveaux abonnés (à droite) quotidiens durant l'expérimentation.

Après 55 jours d'expérimentation, nous observons une augmentation très forte du nombre de retweets de messages du bot. Nous constatons une augmentation similaire du nombre quotidien de nouveaux abonnés au bot comme le montre la Figure 1.5. Nous supposons que, son nombre d'abonnés ayant augmenté, le bot a vu ses tweets et retweets plus visibles sur les pages dédiées aux hashtags contenus dans ses tweets. En effet, le fil d'actualité d'un hashtag ne contient pas de façon exhaustive les tweets des utilisateurs, **Twitter** en renvoyant une sélection des plus *populaires* (Figure 0.3). Notre hypothèse est donc que

ses tweets sont devenus plus populaires, que sa visibilité a augmenté, et ainsi que les interactions des autres utilisateurs avec lui ont fait de même. Cette hypothèse semble cohérente avec le fait que lorsque nous avons stoppé le bot, son nombre d'abonnés a très peu augmenté.

Nous pouvons conclure que tweeter sur ces hashtags permet d'être retweeté et mentionné ce qui entraîne le gain de nouveaux abonnés. Cependant, le simple fait d'être visible sur ces hashtags est particulièrement efficace, les autres utilisateurs suivant le principe **IFYFM** et s'abonnant ainsi directement aux utilisateurs qui postent des tweets contenant ces hashtags.

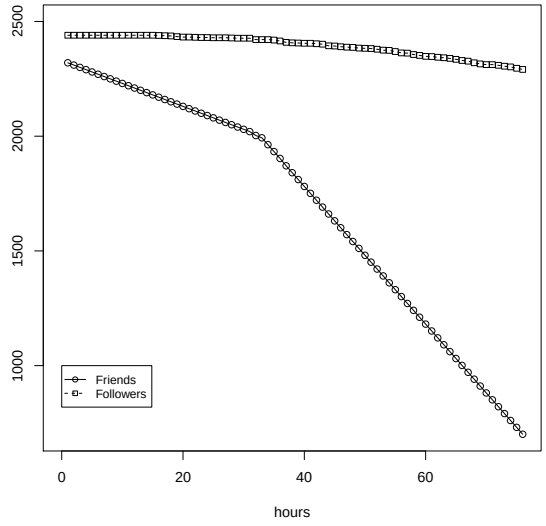

FIGURE 1.6 – Évolution du nombre d'abonnés et d'abonnements d'un utilisateur qui se désabonne de 10 utilisateurs par heure puis de 30 utilisateurs par heure.

Désabonnement massif. Afin de valider l'efficacité du désabonnement massif pour conserver un faible ratio tel que décrit au début de la Section 1.3, nous avons implémenté un programme qui utilise l'API **Twitter** et applique cette méthode sur un compte avec presque autant d'abonnements (2.322) que d'abonnés (2.440). Ce compte est un autre bot que nous avons créé spécialement pour étudier l'impact du désabonnement massif. Les abonnements de ce compte sont donc des comptes identifiés comme capitalistes sociaux par notre méthode. Le programme se désabonne tout d'abord de 10 comptes par heure pendant 30 heures. Puis, il se désabonne de 30 comptes par heure pendant 46 heures. Nous avons fait cela de façon progressive de peur que **Twitter** ne suspende le compte. Comme

nous pouvons le voir sur la Figure 1.6, la très grande majorité des utilisateurs desquels notre bot se désabonne ne s'y désabonnent pas réciproquement. Le bot s'est en effet désabonné de 1.680 comptes en un peu plus de trois jours, et moins de 150 utilisateurs s'en sont désabonnés. Cela confirme à nouveau que les comptes de ces capitalistes sociaux sont administrés manuellement et non de façon automatique. Ainsi, il est tout à fait possible d'appliquer la méthode **IFYFM** tout en conservant un faible ratio.

1.4 Attachement préférentiel

Nous avons ainsi vu que les capitalistes sociaux utilisent les principes **FMIFY** et **IFYFM** pour augmenter leur nombre d'abonnés et ce, sur des hashtags dédiés. Intuitivement, ces principes sont particulièrement efficaces lorsque les utilisateurs ciblés sont également des capitalistes sociaux. Par ailleurs, la pratique de ce genre de méthodes sur des hashtags dédiés regroupant les capitalistes sociaux nous laisse penser qu'il existe une forme d'attachement préférentiel des capitalistes sociaux à d'autres capitalistes sociaux. Nous étudions cette hypothèse en étudiant le voisinage des capitalistes sociaux ainsi que leur *coefficient de clustering*.

1.4.1 Un voisinage de capitalistes sociaux

Nous commençons donc par examiner le voisinage des capitalistes sociaux détectés avec la méthode que nous présentons au Chapitre 1. Comme nous pouvons le voir sur la Figure 1.7, un peu moins de 70% des capitalistes sociaux ont plus de 50% de capitalistes sociaux parmi leurs abonnés. Ceci montre tout d'abord que pour un grand nombre de capitalistes sociaux, la visibilité qu'ils peuvent obtenir sur le réseau en augmentant leur nombre d'abonnés est dûe à la présence d'autres utilisateurs appliquant ces techniques. Cette visibilité est donc complètement artificielle, indépendante du contenu de leurs tweets. Par ailleurs, cette observation soutient l'hypothèse d'un attachement préférentiel entre capitalistes sociaux.

Nous étudions maintenant la connectivité entre les voisins des capitalistes sociaux.

1.4.2 Un coefficient de clustering élevé

Nous savons que les voisins des capitalistes sociaux sont majoritairement des capitalistes sociaux. Intuitivement, on peut penser que ces voisins sont densément connectés entre eux, puisque ce sont eux-mêmes des capitalistes sociaux. Afin de vérifier cette hypothèse expérimentalement, nous utilisons la notion de *coefficient de clustering* local telle qu'elle est présentée par Watts et al. [124]. Pour le calculer, nous ne considérons pas l'orientation des liens. Ainsi, si on étudie un réseau orienté $D = (V, A)$ et que $(u, v) \vee (v, u) \in A$ alors

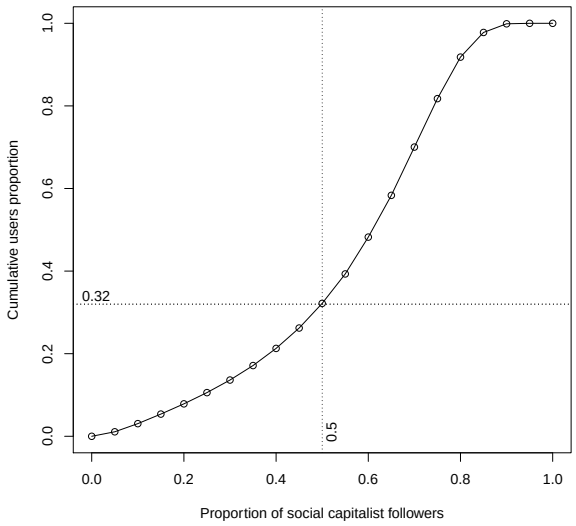

FIGURE 1.7 – Proportion des capitalistes sociaux parmi les abonnés des capitalistes sociaux.

$(u, v) \in E$ dans le réseau non-orienté $G = (V, E)$ sur lequel on calcule le *coefficient de clustering* de chacun des sommets $v \in V$ comme dans la Définition 1.1.

Pour mettre en évidence que le *coefficient de clustering* des capitalistes sociaux est élevé, nous le comparons aux valeurs obtenues pour des sommets dits *réguliers* du réseau. Puisque notre méthode de détection des capitalistes sociaux ne considère que des sommets de degrés entrant et sortant supérieur à 500, nous ne tenons compte que des valeurs obtenues pour des sommets réguliers de degrés entrant et sortant supérieur à 500. Nous observons que 128.212 sommets de notre réseau correspondent à ces contraintes. Nous comparons donc la valeur du *coefficient de clustering* des 161.424 détectés par notre méthode à celle de ces utilisateurs dans la Figure 1.8. En moyenne, le *coefficient de clustering* des capitalistes sociaux (0,084) est presque deux fois plus élévé que celui des utilisateurs considérés comme réguliers (0,044).

A l'issue de ces expérimentations, nous savons que les capitalistes sociaux détectés avec notre méthode ont un voisinage majoritairement constitué de capitalistes sociaux. Par ailleurs, ces voisins sont plus connectés entre eux en moyenne que d'autres utilisateurs aux degrés similaires. Ces observations montrent que les capitalistes sociaux forment un ou des ensembles de sommets plus densément connectés que la moyenne.

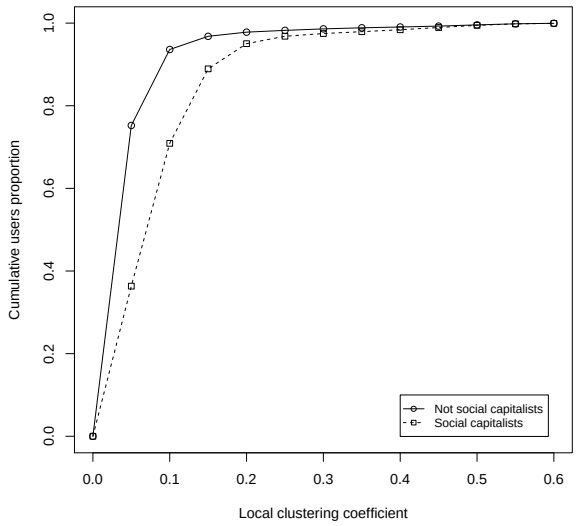

FIGURE 1.8 – Comparaison du coefficient de clustering local des capitalistes sociaux et des autres utilisateurs.

1.5 Évolution des capitalistes sociaux entre 2009 et 2013

Nous décrivons maintenant l'évolution des capitalistes sociaux entre Juillet 2009 et Juillet 2013. Afin de réaliser cette étude, nous utilisons un jeu de données **Twitter** non anonymisé recueilli par Kwak et al. [63]. En effet, le jeu de données de Cha et al. [20] étant anonymisé, il est impossible d'utiliser les identifiants de compte **Twitter** de ce réseau afin d'étudier l'état de ces comptes quatre ans après.

Méthodologie. Afin d'étudier l'évolution des capitalistes sociaux entre 2009 et 2013, il s'agit tout d'abord de détecter les capitalistes sociaux présents sur le réseau de Kwak et al. [63] en 2009. Nous utilisons la méthode de détection décrite Section 1.1 et basée sur l'*indice de chevauchement* et le *ratio*. Sur les 41 millions d'utilisateurs du réseau de Kwak et al. [63], nous obtenons une liste de 145.000 capitalistes sociaux dont les identifiants sont réels et utilisables. Nous utilisons ainsi ces identifiants et l'API **Twitter** [119] pour obtenir les abonnés et les abonnements de ces utilisateurs ainsi que leur *screen name*, afin de vérifier que celui-ci est bien identique au *screen name* fourni par Kwak et al. [63]. Nous avons ainsi récupéré les abonnés et abonnements de 110.000 des 145.000 capitalistes sociaux, ce qui nous fournit un échantillon fiable pour étudier l'évolution de ces utilisateurs. Par la suite nous utilisons la notation *Twitter*'09 pour nous référer au réseau de Kwak et al. [63]. La notation *Twitter*'13 fait référence au réseau formé par les informations obtenues en utilisant l'API **Twitter** en 2013.

Observations générales. Nous commençons par comparer l'évolution du nombre d'abonnés et abonnements ainsi que l'indice de chevauchement et le ratio des 110.000 capitalistes sociaux de *Twitter*'13 (Figures 1.9 et 1.10).

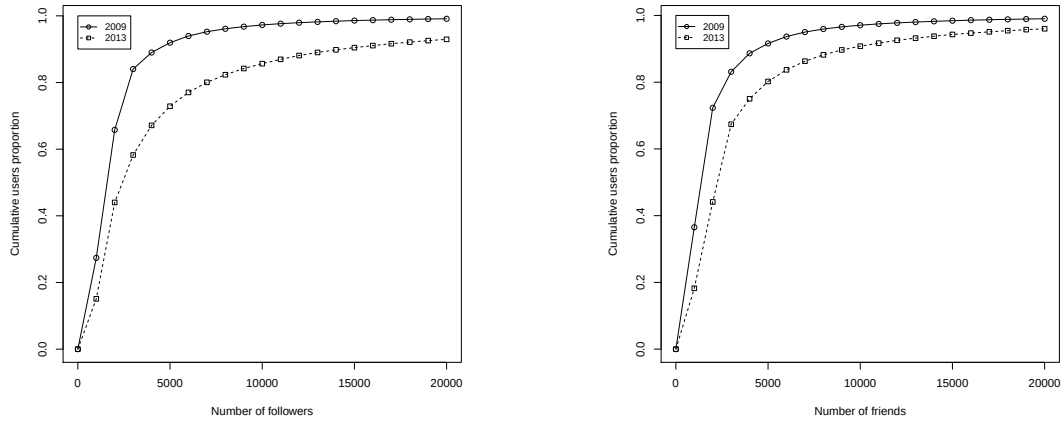

FIGURE 1.9 – Différence entre la distribution cumulative du nombre d'abonnés et d'abonnements entre *Twitter*'09 et *Twitter*'13, respectivement à gauche et à droite.

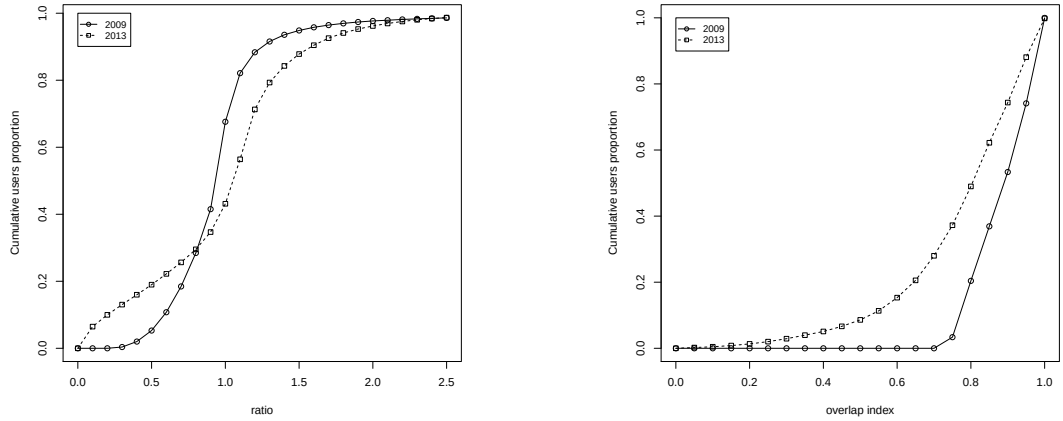

FIGURE 1.10 – Différence entre la distribution cumulative du ratio et de l'indice de chevauchement entre *Twitter*'09 et *Twitter*'13, respectivement à gauche et à droite.

La très grande majorité des utilisateurs a vu son nombre d'abonnés et d'abonnements augmenter entre 2009 et 2013. Quatre ans ont passé, ce n'est donc pas étonnant. De plus, on note que leur nombre d'abonnés a plus augmenté que leur nombre d'abonnements. Par ailleurs, on observe qu'un grand nombre d'utilisateurs de *Twitter*'13 ont un ratio très

proche de 0 alors que de tels utilisateurs ne sont quasiment pas présents dans *Twitter*'09. Par exemple, 75% des 2.500 capitalistes sociaux de *Twitter*'09 qui ont un ratio supérieur à 2 ont un ratio proche de 0 dans *Twitter*'13. Rappelons qu'un ratio proche de 0 indique un nombre d'abonnés très largement supérieur au nombre d'abonnements. Ces utilisateurs ont donc appliqué des méthodes d'abonnements puis de désabonnements massifs avec succès. De telles pratiques sont actuellement plus difficiles qu'en 2009 à cause des restrictions sur le ratio instaurées par **Twitter** [118]. Il n'est ainsi plus possible d'avoir un ratio supérieur à 2 afin de limiter des techniques d'abonnements trop massifs. De tels utilisateurs sont donc passés à travers les mailles du filet. Enfin, plus de 50% des utilisateurs ont toujours un indice de chevauchement supérieur à $0,74$ et seulement 10% des utilisateurs ont un indice de chevauchement inférieur à $0,5$. Le désabonnement massif après l'abonnement peut conduire à la diminution de l'indice de chevauchement. Cela montre à nouveau les limites de notre méthode de détection en terme de rappel.

Utilisateurs de *Twitter*'13 avec un indice de chevauchement $\geq 0,74$. Nous présentons maintenant plus précisément l'évolution du ratio des capitalistes sociaux détectés en 2009 qui conservent un indice de chevauchement supérieur à $0,74$ en 2013. Les résultats présentés dans la Table 1.10 montrent la distribution du ratio de ces utilisateurs en 2013 (colonnes) en fonction de leur ratio en 2009 (lignes).

	Nombre	$I_c^{2013} \geqslant 0.74$	$\mathbf{r}^{2013} > 1$	$\mathbf{r}^{2013} \in [0.7;1]$	$\mathbf{r}^{2013} < 0.7$
$\mathbf{r}^{2009} > 1$	35.058	19.892	40%	25%	35%
$\mathbf{r}^{2009} \in [0.7;1]$	53.986	37.789	80%	14%	6%
$\mathbf{r}^{2009} < 0.7$	20.026	10.435	89%	7%	4%

TABLE 1.10 – Ratio des capitalistes sociaux avec un indice de chevauchement supérieur à $0,74$ en 2013. Les notations r^i and I_c^i dénotent respectivement le ratio et l'indice de chevauchement pour les indices $i \in \{2009, 2013\}$. Les pourcentages sont relatifs aux chiffres de la seconde colonne.

Ces résultats mettent en exergue les observations faites précédemment. En particulier, cela montre que la majorité des utilisateurs de *Twitter*'09 qui avaient un ratio supérieur à 1 ne voient pas leur ratio changer. Ceci indique qu'ils continuent d'appliquer le principe **IFYFM**. On observe que 68% de tels utilisateurs ont un nombre d'abonnés inférieur dans *Twitter*'13 par rapport à celui de *Twitter*'09. Ceci peut indiquer que ces utilisateurs ont été suspendus temporairement par **Twitter** ou que certains de leurs abonnés ont également été suspendus. Les limitations de **Twitter** sur le ratio ont probablement aussi ralenti leur progression. Enfin, certains de leurs abonnés ont probablement pratiqué une méthode de

désabonnement massif afin de diminuer leur ratio. Cependant, pour 35% de ces utilisateurs, la méthode **IFYFM** a été appliquée avec succès en aboutissant à un ratio inférieur à 0,7.

Utilisateurs de *Twitter*'13 avec un indice de chevauchement $< 0,74$. Nous nous concentrons maintenant sur les utilisateurs qui avaient un indice de chevauchement supérieur à 0,74 en 2009 mais dont l'indice de chevauchement est inférieur à 0,74 en 2013. On observe 38.000 de ces utilisateurs qui représentent ainsi 35% de l'échantillon que nous étudions. Comme précédemment, nous analysons l'évolution du ratio de ces utilisateurs dans la Table 1.11.

	Nombre	$I_c^{2013} < 0.74$	$\mathbf{r}^{2013} > 1$	$\mathbf{r}^{2013} \in [0.7;1]$	$\mathbf{r}^{2013} < 0.7$
$\mathbf{r}^{2009} > 1$	35.058	14.103	16%	25%	59%
$\mathbf{r}^{2009} \in [0.7;1]$	53.986	15.377	36%	23%	40%
$\mathbf{r}^{2009} < 0.7$	20.026	8.894	61%	14%	25%

TABLE 1.11 – Ratio des capitalistes sociaux avec un indice de chevauchement inférieur à 0,74 en 2013. Les notations r^i and I_c^i dénotent respectivement le ratio et l'indice de chevauchement pour les indices $i \in \{2009, 2013\}$. Les pourcentages sont relatifs aux chiffres de la seconde colonne.

Un grand nombre d'utilisateurs dont le ratio était supérieur à 1 dans *Twitter*'09 ont un ratio inférieur à 0,7 dans *Twitter*'13, ce qui indique qu'ils ont appliqué les méthodes de capitalisme social efficacement, notamment l'abonnement avec **IFYFM** et le désabonnement massif. Rappelons qu'au moment où la capture du réseau a été réalisée, **Twitter** ne semblait pas imposer de restriction sur le ratio Abonnements/Abonnés. Cette absence de limitation a ainsi été utilisée par de nombreux utilisateurs afin d'augmenter considérablement leur nombre d'abonnés. Comme nous pouvons le voir sur la Table 1.12, certains de ces utilisateurs maintiennent un indice de chevauchement proche de 0,74. L'indice de chevauchement moyen de ces utilisateurs est 0,56. Ceci signifie qu'ils conservent la plupart des liens réciproques qu'ils ont créé en appliquant les principes de capitalismes social.

Utilisateurs de *Twitter*'13 avec un indice de chevauchement inférieur à 0,25. Il existe 2.500 utilisateurs comme ceci. Nous observons que la moitié d'entre eux avaient un ratio proche de 1 dans *Twitter*'09 et ont maintenant un ratio inférieur à 0,23. Puisque leur indice de chevauchement est très faible, ces utilisateurs sont ceux qui ont appliqué les méthodes de capitalisme social efficacement puis se sont désabonnés massivement. Par ailleurs, si

Nom	**abonnements**	**abonnés**	**ratio**	I_c
Coldplay	151.7067	2.633	576,174	0,886
UberSoc	4.090	1.213	3,372	0,839
SHAQ	1.843.561	563	3274,531	0,838
ESPN	138.982	109.377	1,271	0,843
funnyordie	593.981	1.691	351,260	0,782
noaheverett	3.635	588	6,182	0,767
MLB	133.953	8.509	15,743	0,781
TFL	71.027	1.506	47,163	0,764
addthis	14.093	13.363	1,055	0,927
TechCruch	1.041.057	691	1506,595	0,839

Nom	**abonnements**	**abonnés**	**ratio**	I_c
Coldplay	2.391	105.70.550	0,000	0,701
UberSoc	2.418	10.495.065	0,000	0,570
SHAQ	1.110	7.225.068	0,000	0,686
ESPN	360	6.441.694	0,000	0,606
funnyordie	4.248	6.200.508	0,001	0,546
noaheverett	1.227	4.497.421	0,000	0,737
MLB	2.198	2.964.578	0,001	0,602
TFL	1.730	2.882.159	0,001	0,602
addthis	22.762	2.750.610	0,008	0,736
TechCruch	863	2.728.905	0,000	0,718

TABLE 1.12 – Les 10 premiers utilisateurs avec un indice de chevauchement inférieur à $0,74$ ordonnés par leur nombre d'abonnés en 2013. Les nombres pour *Twitter*'09 sont en haut, ceux pour *Twitter*'13 en bas.

nous considérons les utilisateurs avec un ratio inférieur à $0,7$ *Twitter*'09, nous observons que les différences entre le nombre d'abonnés et d'abonnements dans les deux réseaux sont relativement faibles. Ainsi, ces utilisateurs ont probablement cessé d'appliquer les techniques de capitalisme social.

Des comptes de célébrités. Afin de conclure cette Section, nous nous concentrons sur plusieurs comptes qui ont réussi à appliquer efficacement les techniques de capitalisme social. Plus précisément, nous considérons des comptes qui ont un indice de chevauchement supérieur à 0,74 et un ratio supérieur à 1 dans *Twitter*'09, et qui conservent un indice de chevauchement supérieur à 0,74 dans *Twitter*'13 mais avec un ratio très faible. Parmi cette classe d'utilisateurs, ceux qui ont gagné le plus d'abonnés sont **Lady Gaga**, **Barack Obama**, et **Britney Spears**, qui sont des capitalistes sociaux qui avaient été mis en évidence par Ghosh et al. [40]. Nous présentons des résultats similaires dans la Table 1.13, où certains comptes affichent une diminution impressionnante de leur ratio (e.g. **paulpierce34**, qui est passé de 1.500 à un chiffre proche de 0). Ces utilisateurs ont bénéficié de l'absence de restrictions sur le ratio à l'époque.

Nom	**abonnements**	**abonnés**	**ratio**	I_c
ladygaga	636.929	73274	8,69	0,97
BarackObama	1.882.889	770.155	2,44	0,91
BritneySpears	2.674.874	406.238	6,58	0,95
paulocoelho	75.423	48.446	1,56	0,98
Anahi(Magia)	15.337	1.765	8,69	0,96
stephenfry	693.512	55.044	12,60	0,90
hootsuite	80.936	61.828	1,31	0,94
TheOnion	1.380.160	369.569	3,73	0,87
showdavida	598.444	1005	595,47	0,90
yokoono	81.765	71.585	1,14	0.95
DwightHoward	727.413	2.564	283,70	0,82
Starbucks	271.215	138.045	1,96	0,96
NatGeo	14.339	11.755	1,22	0,99
WholeFoods	1.112.628	498.700	2,23	0,89
jimmycarr	183.925	1.344	136,85	0,89
wossy	386.479	3.985	96,98	0,96
wyclef	643.237	3.412	188,52	0,93
paulpierce34	815.197	524	1555,72	0,95
zappos	1.075.935	407.705	2,64	0,88

Nom	**abonnements**	**abonnés**	**ratio**	I_c
ladygaga	136.386	37.485.540	0,00	0,82
BarackObama	680.428	30.836.226	0,02	0,77
BritneySpears	412.703	27.763.836	0,01	0,81
paulocoelho	98	7.721.670	0,00	0,86
Anahi(Magia)	455	6.492.119	0,00	0,90
stephenfry	54.122	5.823.567	0,01	0,81
hootsuite	1.274.698	5.013.919	0,25	0,75
TheOnion	8	4.931.732	0,00	0,87
showdavida	36	4.651.921	0,00	0,75
yokoono	1.003.791	4.311.890	0,23	0,79
DwightHoward	8.037	4.134.991	0,00	0,84
Starbucks	86.594	3.723.806	0,02	0,77
NatGeo	23.792	3.668.447	0,01	0,88
WholeFoods	562.219	3.400.523	0,16	0,76
jimmycarr	228	3.302.043	0,00	0,74
wossy	5.966	3.249.585	0,00	0,87
wyclef	8.649	3.246.278	0,00	0,80
paulpierce34	85	2.804.060	0,00	0,78
zappos	380.335	2.759.301	0,14	0,75

TABLE 1.13 – Les nombres pour *Twitter*'09 sont en haut et ceux pour *Twitter*'13 en bas.

CHAPITRE

2 Rôles communautaires

Nous avons présenté les capitalistes sociaux sur **Twitter**, les mécanismes qu'ils utilisent pour essayer d'augmenter leur visibilité et ainsi tenter d'acquérir de l'influence sur le média social. Dans ce Chapitre, nous étudions la visibilité des capitalistes sociaux sur le réseau. Nous cherchons à vérifier que les techniques mises en place par ces utilisateurs leur permettent de tisser des liens virtuels à travers une large frange du réseau. Nous utilisons pour cela la structure de communautés du réseau. Nous nous servons de cette notion pour étudier la position des capitalistes sociaux à un niveau d'observation intermédiaire du réseau, le voisinage étant considéré comme le niveau local et le réseau entier comme le niveau global.

Pour ce faire, nous évaluons la qualité d'une adaptation de l'algorithme de Louvain pour optimiser la modularité orientée. Puis nous présentons en détails la notion de rôle communautaire, qui nous sert à étudier la position d'un utilisateur au sein de la structure de communautés du réseau. Nous présentons notamment la méthode de Guimerà et Amaral [49] pour détecter les rôles communautaires puis en décrivons les limites. Nous établissons ensuite une nouvelle méthode de détection de rôles communautaires pour pallier ces limites. Cette dernière repose notamment sur l'introduction de nouvelles mesures destinées à mieux caractériser le rôle communautaire d'un noeud. En utilisant cette méthode, nous montrons que les rôles occupés par les capitalistes sociaux au sein du réseau de**Twitter** mettent en évidence leur grande visibilité.

2.1 Détecter les communautés d'un grand réseau orienté

2.1.1 Structure de communautés

En 2002, Girvan et Newman [42] montrent qu'une variété de réseaux sociaux et biologiques possèdent une structure de communautés. Pour les auteurs, une structure de communautés est une *partition* des nœuds du réseau telle que les nœuds à l'intérieur de chaque partie sont densément connectés, tandis que les connexions entre parties sont moins denses. C'est cette notion qui nous intéresse : les capitalistes sociaux semblent former un ensemble de noeuds densément connectés.

Girvan et Newman [42] présentent d'abord dans cet article les méthodes qu'ils appellent "traditionnelles" de partitionnement hiérarchique du réseau. Tout d'abord, pour chaque paire (i,j) de sommets de V, on calcule un poids W_{ij} qui représente la densité de connexion entre ces deux noeuds. Ce poids peut par exemple être calculé à partir du nombre de chemins sommets-indépendants entre ces sommets : deux chemins sont dits sommets-indépendants si l'intersection des sommets par lesquels ils passent se restreint aux sommets initial et final. Une fois les poids calculés, un nouveau réseau est créé avec les sommets de V, sans arêtes. Les paires $(i, j) \in V$ sont triées par ordre décroissant selon le poids W_{ij} calculé. Une à une, dans cet ordre, des arêtes de poids W_{ij} sont ajoutées au réseau jusqu'à ce qu'il soit connexe. On obtient ainsi une hiérarchie de partitions emboitées.

Les résultats peu satisfaisants de ces méthodes poussent Girvan et Newman [42] à développer une approche basée sur la centralité des arêtes du réseau. Ils adaptent la mesure de centralité d'intermédiarité (*betweenness*) des nœuds du réseau proposée par Freeman [39] afin de pouvoir calculer la centralité des arêtes du réseau. La centralité d'une arête (u, v) est ainsi fonction du nombre de plus court chemins entre paires de sommets (i, j) du réseau qui passent par (u, v). Afin de partitionner le réseau, l'algorithme suivant est itéré :

— Calculer la centralité des arêtes du réseau ;
— Supprimer l'arête de centralité la plus élevée.

Grâce à cet algorithme, les auteurs mettent en évidence et déplient la structure de communautés de plusieurs réseaux du réel. Détecter ces communautés est un axe de recherche important. Dans les réseaux du web, les communautés peuvent correspondre à une même thématique [41], dans les réseaux sociaux à des groupes d'amis, de collègues ou à des groupes d'intérêt [128], dans des réseaux d'associations de mots à des associations d'idées courantes [19], etc. Par ailleurs, déplier de façon hiérarchique des communautés peut également fournir un moyen de visualiser à différents niveaux de précision un réseau [33].
Ainsi, de nombreux travaux autour de la caractérisation de la structure de communautés ont vu le jour. Certains proposent d'autres définitions de la structure de communautés,

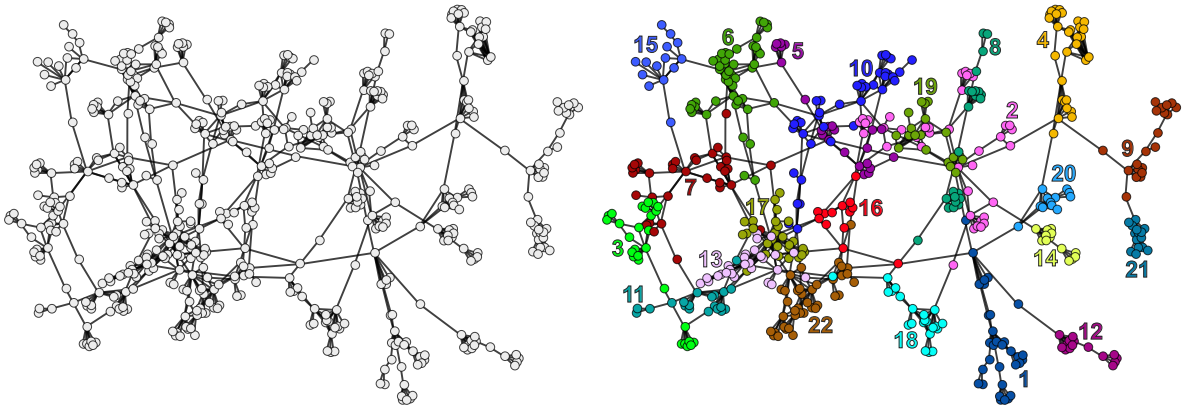

FIGURE 2.1 – A droite, on observe la structure de communautés établie sur le réseau de gauche. Chaque couleur représente une communauté.

par exemple adaptées aux réseaux orientés [73], d'autres mettent en lumière le fait que les communautés semblent se chevaucher dans les réseaux du réel [92]. Des méthodes d'évaluation des communautés détectées basées sur des mesures comme la conductance ont également émergé.

Définition 2.1 (Conductance)**.** *Soit m_i le nombre d'arêtes dans la communauté C_i et b_i le nombre d'arêtes (u, v) telles que $u \in C_i$ et $v \notin C_i$, alors la conductance est définie comme suit :*

$$\frac{b_i}{2 \cdot m_i \cdot b_i}$$

La conductance mesure la proportion des arêtes d'une communauté dont l'autre extrémité est située à l'extérieur de la communauté [127]. Leskovec et al. montrent que la conductance est une mesure particulièrement fiable [127]. Par ailleurs, de nombreux autres algorithmes de détection de communautés ont vu le jour : des méthodes à base de percolation de cliques ou quasi-cliques [96], utilisant la propagation de label [98], la théorie de l'information et les marches aléatoires [101], l'algèbre linéaire et la théorie spectrale des réseaux, les approches égo-centrées [25] ou encore les approches par optimisation de fonction de qualité [90]. Nous nous intéressons à ces dernières dans la prochaine Section, et en particulier à l'algorithme de Louvain introduit par Blondel et al. [10] .

2.1.2 Algorithme de Louvain

La modularité est introduite par Newman [90] pour mesurer la qualité d'une partition des noeuds du réseau (Définition 2.2).

Définition 2.2 (Modularité)**.** *La modularité Q d'un réseau est donnée par la formule suivante :*

$$Q = \frac{1}{2m}\sum_{i,j}(A_{ij} - \frac{d(i)d(j)}{2m})\delta(c_i, c_j)$$

avec $d(i)$ le degré du noeud i, m le nombre d'arêtes du réseau, A_{ij} le poids de l'arête entre i et j ou 0 *s'il n'y en a pas, c_i la communauté du noeud i et $\delta(c_i, c_j)$ égal à* 1 *si i et j sont dans la même communauté,* 0 *sinon.*

Lorsque l'on cherche à partitionner le réseau en communautés, on ne connaît pas le nombre de communautés recherché. Le problème est donc d'estimer le nombre de communautés et leur topologie simultanément. L'optimisation de la modularité a pour avantage de fournir un moyen de mesurer la qualité d'une partition du réseau de façon *objective*. En effet, elle compare le nombre d'arêtes dans les communautés trouvées au nombre attendu dans un réseau aléatoire similaire, de même taille et à distribution de degré identique. La structure communautaire détectée n'est ainsi pas issue d'une organisation aléatoire, les ensembles densément connectés obtenus sont dûs à une organisation specifique du réseau. Optimiser la modularité conduit donc à optimiser la partition du réseau.

Cette optimisation est cependant très difficile. Pour en expliquer la difficulté, nous introduisons brièvement quelques notions de complexité des algorithmes que nous réutiliserons par la suite :

1. Un problème appartient à la classe des problèmes **P** s'il est possible de le *résoudre* en temps polynomial par rapport à la taille de l'entrée ;
2. Un problème est dans la classe **NP** s'il est possible de *vérifier* en temps polynomial une solution proposée ;
3. Un problème est NP-**difficile** si tout problème de la classe **NP** peut se réduire vers ce problème en temps polynomial ;
4. Un problème est NP-**complet** s'il vérifie la condition précédente et qu'il appartient à **NP**.

La recherche d'une partition optimale, i.e. de modularité maximale, est un problème NP-**difficile** d'après Brandes et al. [16]. La rechercher se fait ainsi en temps *exponentiel* par rapport à la taille du réseau. Il s'agit donc de trouver des heuristiques pour obtenir une *bonne* partition du réseau en un temps raisonnable.

La méthode de Louvain [10] est une méthode d'optimisation *gloutonne* de la modularité (voir Définition 2.2). À notre connaissance, elle est la seule à passer à l'échelle des centaines de millions de nœuds sur une machine seule. Par ailleurs, la méthode produit des

partitions de bonne qualité d'après les tests effectués par Fortunato et Lanchichinetti sur le benchmark LFR [38]. Ceci explique pourquoi nous nous concentrons sur cette méthode.

L'algorithme de Louvain [10] procède comme suit. Tout d'abord, on initialise l'algorithme en considérant chaque nœud comme une communauté du réseau. On itère ensuite deux phases. Dans la première, on choisit tour à tour chacun des nœuds du réseau (l'ordre est aléatoire), et on considère leur voisinage. Pour chacun des voisins v du nœud u choisi, on calcule le gain de modularité obtenu si l'on déplace u dans la communauté de v. On choisit le plus grand gain de modularité (s'il y en a un) et on effectue le déplacement. Si aucun gain de modularité n'est possible, alors le nœud ne change pas de communauté. Une fois que plus aucun gain de modularité n'est possible, on applique la seconde phase de l'algorithme qui consiste en la création d'un nouveau réseau contracté. Les nœuds d'une même communauté sont rassemblés en un seul nœud qui représente la communauté. Les liens entre ces nouveaux nœuds ont pour poids la somme des liens entre communautés dans le réseau précédent. On forme ainsi un nouveau méta-réseau contracté des communautés détectées à la première phase sur lequel on applique de nouveau la première phase de l'algorithme et ainsi de suite. La rapidité de la méthode de Louvain provient notamment du fait qu'il est très rapide de calculer le gain de modularité.

2.1.3 Algorithme de Louvain orienté

Puisque notre réseau est orienté, nous adaptons aux réseaux orientés le code de l'algorithme de Louvain fourni par les auteurs de la méthode de Blondel et al. [10]. Pour cela, nous utilisons la définition de la modularité orientée de Leicht et Newman [73] :

Définition 2.3 (Modularité orientée)**.** *La modularité orientée d'un réseau est calculée comme suit :*

$$Q_o = \frac{1}{2m}\sum_{i,j}(A_{ij} - \frac{d^-(i)d^+(j)}{2m})\delta(c_i,c_j)$$

avec $d^-(i)$ le degré entrant du noeud i, $d^+(j)$ le degré sortant du noeud j, m le nombre d'arêtes du réseau, A_{ij} le poids de l'arête entre i et j ou 0 *s'il n'y en a pas, c_i la communauté du noeud i et $\delta(c_i,c_j)$ égal à* 1 *si i et j sont dans la même communauté,* 0 *sinon.*

Ainsi, la nouvelle méthode du *Louvain orienté* optimise non plus la modularité mais la modularité orientée. Pour évaluer l'efficacité du *Louvain orienté*, nous utilisons l'outil de génération de réseaux artificiels introduit par Lancichinetti et Fortunato [65] que nous appelons *LFR*.

LFR benchmark. L'outil permet de faire varier des paramètres importants et ainsi de générer des réseaux artificiels réalistes avec une structure de communautés. La structure

Algorithm 1: Pseudo-code de l'algorithme de Louvain.

```
// Initialisation
G = (V, E)
i = 0
for v ∈ V do
    v.com = i
    i++
modularityGain=true
while modularityGain do
    // Phase 1, calcul des communautés
    modularityGain2=true
    modularityGain=false
    while modularityGain2 do
        modularityGain2=false
        for v ∈ V do
            gain=0
            bestCom=-1
            for u ∈ N(V) do
                newGain = calculerGain(v, u.com)
                if newGain > gain then
                    gain = newGain
                    bestCom=u.com
            if gain > 0 then
                v.com = bestCom
                modularityGain=true
                modularityGain2=true
    // Phase 2, création du métagraphe
    V_new = {}
    for v ∈ V do
        if v.com ∉ V_new then
            V_new.add(v.com)
```

de communautés existant par construction sur ces réseaux, il est ainsi possible de disposer d'une vérité à laquelle comparer les sorties de notre algorithme.

Les paramètres à faire varier lors de l'utilisation du LFR sont :

— La loi de puissance t_1 suivie par les degrés des noeuds du réseau ;
— La loi de puissance t_2 suivie par la taille des communautés à détecter ;

— Le degré moyen des noeuds $\bar{d}$;
— Le degré maximum des noeuds $max(d)$;
— Les tailles minimum $minc$ et maxium $maxc$ des communautés ;
— Le paramètre de mixage μ qui sert à définir la netteté des communautés : quand μ est petit (resp. grand), les communautés sont faciles (resp. difficiles) à détecter.

Mesures de qualité. Pour comparer les résultats retournés par un algorithme de détection de communautés à une partition de référence, nous utilisons dans ce manuscrit trois mesures d'évaluation. La première mesure, appellée *V-Mesure* [100] est basée sur deux critères : *l'homogénéité* et la *complétude*. Une partition maximise l'homogénéité si, pour chaque partie, on trouve seulement des éléments de la même communauté. De manière symétrique, la complétude est maximisée quand pour chaque partie, tous les éléments sensés être réunis le sont. En calculant la moyenne harmonique de ces mesures, on obtient la *V-Mesure*.

La seconde mesure est la NMI [109] pour *Normalized Mutual Information*. Cette mesure est basée sur des concepts de la théorie de l'information. Nous utilisons la normalisation introduite par Strehl et Ghosh [109].

La dernière mesure que nous utilisons est la *Pureté* [129]. Pour la calculer, il s'agit d'assigner à chaque communauté détectée, la partie de la structure de communauté de référence dont le plus de noeuds y sont représentés. Ensuite, en sommant tous les noeuds correctement classifiés pour chaque partie et en divisant par le nombre de sommets, on obtient la *Pureté*.

Résultats. Afin d'évaluer la qualité des partitions retournées par le *Louvain orienté*, nous générons un benchmark d'évaluation en utilisant le LFR. À l'identique de Fortunato [36], deux jeux de paramètres sont utilisés dans lesquels $\bar{d}$ et $max(d)$ sont respectivement fixés à 20 et 50. Les lois de puissance t_1 et t_2 sont également respectivement fixées à 2 et 1. Dans le premier jeu de paramètres, $n = 1000$ et $minc$ et $maxc$ ont respectivement pour valeur 10 et 50. Dans le second jeu de paramètres, $n = 5000$ et $minc$ et $maxc$ ont respectivement pour valeur 20 et 100. Enfin, nous faisons varier le paramètre μ entre $0,1$ et $0,6$.

Nous comparons tout d'abord les résultats de *Louvain orienté* à la méthode de Louvain [10] sur ces réseaux artificiels. Pour exécuter Louvain sur un réseau orienté, nous ne tenons simplement pas compte de l'orientation des liens. Nous montrons dans la Table 2.1 que la méthode de Louvain adaptée aux réseaux orientés fournit de meilleurs résultats que l'originale sur les benchmarks orientés du LFR [65]. L'algorithme de Louvain qui maximise la modularité orientée est meilleur dans 75% des cas. Nous le comparons également à une

n	μ	NMI	V-mesure	Homogénéité	Complétude	Pureté
1000	0,1	0,987	0,987	1,000	0,975	1,000
1000	0,6	0,965	0,964	0,999	0,932	0,999
5000	0,1	0,966	0,965	1,000	0,934	1,000
5000	0,6	0,909	0,905	0,999	0,828	0,999

n	μ	NMI	V-mesure	Homogénéité	Complétude	Pureté
1000	0,1	**0,995**	**0,995**	1,000	**0,990**	1,000
1000	0,6	**0,978**	**0,978**	**1,000**	**0,958**	**1,000**
5000	0,1	**0,978**	**0,978**	**1,000**	**0,957**	1,000
5000	0,6	**0,920**	**0,917**	0,999	**0,848**	0,999

TABLE 2.1 – Résultats obtenus sur les réseaux du LFR avec l'algorithme de Louvain qui optimise d'abord la modularité classique, puis celle orientée dans le tableau du dessous. Chaque mesure indique la moyenne obtenue sur 100 graphes.

approche statistique, OSLOM [67]. OSLOM recherche des partitions significatives statistiquement : une communauté est significative si la probabilité de la trouver dans un réseau aléatoire est faible [67]. OSLOM est adaptée à tous types de réseaux, qu'ils soient orientés ou non. Par ailleurs, cette méthode montre en général de meilleures performances que celle de Louvain : ses résultats sont plus proches de ceux attendus dans le cas de tests effectués sur les réseaux artificiels du LFR [65] comme le montre la Table 2.2. En revanche, contrairement à la méthode du Louvain, OSLOM est incapable de passer à l'échelle de réseaux contenant des centaines de millions de liens. Son temps d'exécution est supérieur à 10 heures sur des réseaux d'un peu plus de 300.00 liens.

n	μ	NMI	V-mesure	Homogénéité	Complétude	Pureté
1000	0,1	0,999	0,999	0,999	0,999	0,999
1000	0,6	0,999	0,999	0,999	0,999	0,999
5000	0,1	0,999	0,999	0,999	0,999	0,999
5000	0,6	0,999	0,999	0,999	0,999	0,999

TABLE 2.2 – Resultats obtenus en utilisant OSLOM sur les réseaux du LFR. Chaque valeur est une moyenne obtenue sur 100 graphes.

Les meilleures performances de la méthode du *Louvain orienté* par rapport au *Louvain* en justifient l'utilisation dans la suite de ce Chapitre. En effet, même si les performances

d'OSLOM sont encore meilleures, cet algorithme n'est pas une option envisageable à cause de sa complexité. OSLOM ne passe pas à l'échelle d'un réseau de la taille de celui de Twitter. Nous précisons que des résultats plus complets à propos du *Louvain orienté* sont présentés Annexe B.

Nous avons donc une méthode de détection de communautés capable de passer à l'échelle du réseau Twitter et de tenir compte de l'orientation de ses liens. Il s'agit maintenant de décrire la position des capitalistes sociaux relativement à la structure de communautés détectée. Pour cela, nous introduisons dans la prochaine Section la notion de rôles communautaires.

2.2 Rôles communautaires : approche originale

2.2.1 Degré intra-module et Coefficient de participation

La notion de *rôles* en sciences des réseaux est apparue dans les années 1970. Deux noeuds sont considérés comme ayant le même rôle s'ils sont structurellement équivalents d'après Lorrain et al. [78], i.e. s'ils partagent les mêmes voisinages dans le graphe selon Burt [17]. Cette notion apparaît aussi dans les modèles par blocs, où les réseaux sont partitionnés en groupes partageant les mêmes motifs de connexion (Holland et al. [51]). Dans les deux cas, le concept de rôle est défini globalement, i.e. relativement à tout le réseau.

Considérons maintenant différents niveaux d'observations du réseau. Le niveau local est constitué de l'étude du voisinage d'un noeud. Le niveau global constitue lui l'étude d'un nœud à travers sa position dans tout le réseau. La structure de communautés permet d'observer le réseau à un niveau intermédiaire. Cela permet de recueillir plus d'informations que l'étude au niveau local, mais aussi d'éviter des coûts calculatoires trop importants lorsque l'on prend pour échelle la totalité du réseau. La notion de rôle communautaire est un bon exemple des avantages fournis par l'étude du réseau à l'échelle de sa structure communautaire. Il s'agit d'étudier la position du nœud au sein de sa communauté, d'en déterminer son ancrage et de qualifier ses connexions avec l'extérieur. Cette notion est introduite par Guimerà et Amaral [49] pour l'étude de réseaux biologiques. A l'aide de la méthode développée, ils montrent notamment que les rôles dans un réseau métabolique ont un sens biologique.

Pour caractériser les rôles des nœuds, Guimerà et Amaral [49] définissent d'abord deux mesures complémentaires. La première mesure permet de caractériser la façon dont un nœud est connecté à sa communauté, i.e. sa *connectivité interne*. La seconde mesure encapsule les connexions d'un nœud avec toutes les communautés auxquelles il est lié, i.e. sa *connectivité externe*. Ces deux mesures permettent aux auteurs de placer chaque nœud dans un

espace bidimensionnel. Ils proposent ensuite plusieurs seuils pour discrétiser cet espace, chaque zone ainsi définie correspondant à un rôle particulier. Nous décrivons d'abord les mesures, puis la méthode qu'ils utilisent pour identifier les rôles. Nous rappelons que nous notons $d^-(v)$ (resp. $d^+(v)$) le degré entrant (resp. sortant) d'un nœud v, i.e. le nombre de liens entrants (resp. sortants) connectés à ce nœud. À partir de cette notation, nous définissons le *degré entrant interne* (resp. *externe*) d'un nœud v, noté $d^-_{int}(v)$ (resp. $d^-_{ext}(v)$)et représentant le nombre de liens entrants que le nœud possède à l'intérieur (resp. extérieur) de sa communauté. On définit de la même façon les degrés sortants interne et externe en remplaçant − par + dans les précédentes notations. Nous définissons $d^-_i(v)$ comme le *degré communautaire entrant*, à savoir le nombre de liens entrants qu'un nœud v a avec les nœuds de la communauté c_i. En remplaçant − par +, on obtient le *degré communautaire sortant.*

Degré intra-module. La première mesure, nommée *degré intra-module* traite de la connectivité interne du nœud. Elle est basée sur la notion de z-score. Nous définissons tout d'abord le z-score de façon générique, cette notion sera réutilisée plus tard.

Définition 2.4 (Z-Score)**.** *Pour une fonction nodale quelconque $f(u)$, permettant d'associer une valeur numérique à un nœud u, le z-score $Z_f(u)$ par rapport à la communauté de u est :*

$$Z_f(u) = \frac{f(u) - \mu_i(f)}{\sigma_i(f)}$$

avec $u \in c_i$ où c_i représente une communauté, et $\mu_i(f)$ et $\sigma_i(f)$ dénotent respectivement la moyenne et l'écart-type de f sur les nœuds appartenant à la communauté c_i.

À partir de la définition de z-score, on obtient le degré intra-module de Guimerà et Amaral [49] en substituant le degré interne $d_{int} = d^+_{int} + d^-_{int}$ à f dans l'équation (2.4).

Définition 2.5 (Degré intra-module)**.** *Le degré intra-module de Guimerà et Amaral , noté $z(u)$, correspond au z-score du degré interne, calculé pour la communauté du nœud considéré.*

Coefficient de participation. La seconde mesure, appelée *coefficient de participation*, traite de la connectivité externe du nœud, i.e. relative à toutes les communautés auxquelles il est lié. Elle est définie de la manière suivante :

Définition 2.6 (Coefficient de participation)**.** *On note la participation externe comme suit :*

$$P(u) = 1 - \sum_i \left(\frac{d_i(u)}{d(u)} \right)^2$$

où $d_i(u) = d^+_i(u) + d^-_i(u)$ représente le nombre de liens que u possède vers des nœuds de la communauté c_i. Notons que dans le cas où c_i est la communauté de u, alors on a $d_i(u) = d_{int}(u)$.

Le coefficient de participation représente la diversité des connexions d'un nœud en terme de communautés externes. Une valeur proche de 1 signifie que le nœud est connecté de façon uniforme à un grand nombre de communautés différentes. Au contraire, une valeur de 0 ne peut être atteinte que si le nœud n'est connecté qu'à une seule communauté (vraisemblablement la sienne).

Guimerà et Amaral [49] proposent de caractériser le rôle d'un nœud dans un réseau en se basant sur ces deux mesures. Pour ce faire, ils définissent **sept** rôles différents en discrétisant l'espace à deux dimensions formé par z et P (Figure 2.2).

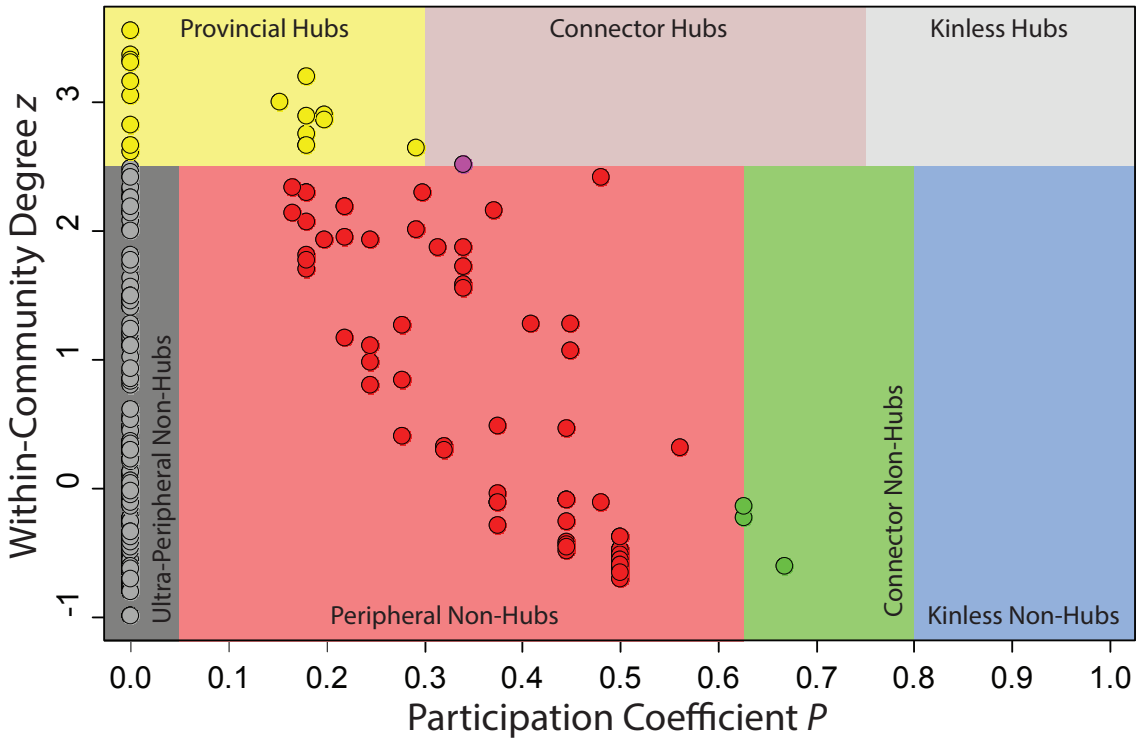

FIGURE 2.2 – Représentation des noeuds d'un réseau en 2 dimensions selon leur connectivité externe en abscisse et leur connectivité interne en ordonnée. Chaque zone de couleur représente un rôle selon les seuils définis par Guimerà et Amaral (voir Tableau 2.3). Figure issue de Guimerà et Amaral [49].

Un premier seuil défini sur le degré intra-module z permet de distinguer ce que les auteurs appellent les *hubs communautaires* ($z \geq 2,5$) des autres nœuds ($z < 2,5$). Ces *hubs* sont considérés comme fortement intégrés à leur communauté, par rapport au reste des nœuds de cette même communauté. Ces deux catégories (hub et non-hub) sont subdivisées au moyen d'une série de seuils définis sur le coefficient de participation P. En considérant les nœuds par participation croissante, Guimerà et Amaral [49] les qualifient de *provinciaux*

ou *(ultra-)périphériques*, *connecteurs* et *orphelins*. Les deux premiers rôles sont essentiellement connectés à leur communauté, les troisièmes, bien qu'eux aussi potentiellement bien connectés à leur propre communauté, sont également largement liés à d'autres communautés, et les derniers sont connectés à un grand nombre de communautés.

Rôle communautaire				**Connectivité Externe**
Degré intra-communautaire		**Participation Coefficient**		
Hub	$z \geq 2,5$	Provincial	$P \leq 0,30$	Faible
		Connecteur	$P \in]0,30;0,75]$	Forte
		Orphelin	$P > 0,75$	Très forte
Non-Hub	$z < 2,5$	Ultra-périphérique	$P \leq 0,05$	Très basse
		Périphérique	$P \in]0,05;0,62]$	Basse
		Connecteur	$P \in]0,62;0,80]$	Forte
		Orphelin	$P > 0,80$	Très forte

TABLE 2.3 – Rôles communautaires en fonction des connectivités interne et externe.

Nous proposons maintenant d'adapter les mesures proposées par Guimerà et Amaral [49] aux réseaux orientés puisque le réseau d'abonnements Twitter sur lequel nous souhaitons appliquer la méthode est un réseau orienté.

2.2.2 Orientation des liens

Il est souvent assez simple de généraliser des mesures définies sur des graphes non-orientés vers des graphes orientés. En effet, le schéma classique consiste à distinguer les liens *entrants* des liens *sortants*. Dans notre cas, cela consiste à utiliser 4 mesures au lieu de 2 : degrés intra-module entrant et sortant, ainsi que coefficients de participation entrant et sortant.

En calculant le z-score du *degré entrant interne* $d^-_{int}(v)$ d'un nœud v, nous obtenons ainsi le *degré intra-module entrant*, noté z^-. De manière similaire, en remplaçant $d(v)$ par $d^-(v)$ et $d_i(v)$ par $d^-_i(v)$ dans la Définition 2.6, nous définissons le *coefficient de participation entrant*, noté P^-. Le *degré intra-module sortant* z^+ et le *coefficient de participation sortant* P^+ sont obtenus de façon symétrique, en utilisant les contreparties sortantes des degrés entrants : d^+, d^+_{int} et d^+_i .

Cette adaptation aux réseaux orientés des mesures proposées par Guimerà et Amaral [49] ne constitue pas une réelle innovation. En revanche, cela nous permet d'ores et déjà de nous questionner sur la méthode de définition des rôles. Celle-ci est en effet basée sur des seuils définis empiriquement. Ainsi, pour passer d'un réseau non-orienté à un réseau

orienté, il faut réappliquer cette méthode pour définir de nouveaux seuils et partitionner un espace de dimension 4. Cette observation est valide pour toute mise-à-jour des mesures. Ceci nous amène donc à considérer les limites de l'approche originale.

2.2.3 Limites

Seuils. Pour définir leurs seuils sur les mesures de connectivité interne z et de connectivité externe P, Guimerà et Amaral [47] ont calculé les valeurs de P et z pour les noeuds de plusieurs réseaux :

- Quatre réseaux biologiques ;
- Un réseau de transport ;
- Deux réseaux de collaboration ;
- Un réseau de l'internet ;
- 8 réseaux artificiels obtenus avec le modèle d'attachement préférentiel de Barabási-Albert [5] et avec le modèle de réseaux aléatoires défini par Erdös-Rényi [34].

Ils se sont ensuite servis des valeurs de P et z de chaque noeud pour les placer dans l'espace de dimension 2 formé par ces deux mesures. A partir de cela, Guimerà et Amaral [49] ont créé une *carte de chaleur* décrivant la densité en points pour chaque zone de ce plan qui leur a permis de définir les seuils pour chacune de ces deux mesures (Figure 2.2).

Tout d'abord, il est surprenant dans une telle analyse de prendre en compte des modèles de graphes aléatoires. Ceux-ci ne comportent ni structure de communautés ni distribution de degrés en loi de puissance. De même, le modèle d'attachement préférentiel ne possède pas de structure communautaire. Le sens des valeurs obtenues est donc difficilement interprétable relativement aux réseaux du réel. Par ailleurs, ces seuils sont considérés universels. Pourtant, seulement 8 réseaux du réel sont étudiés. De plus, une seule méthode de détection de communautés est utilisée. Une autre méthode peut conduire à une structure de communautés aux caractéristiques différentes, et ainsi à d'autres seuils. Par ailleurs, z n'est pas normalisée, il n'existe donc pas de limites à cette mesure. Aucune garantie ne peut donc être donnée quant à la validité du seuil fixé par Guimerà et Amaral [49] sur d'autres réseaux. D'ailleurs, nos expérimentations (voir Section 2.4) montrent des valeurs de z bien plus élevées que celles observées par Guimerà et Amaral [49]. De plus, nous observons que la proportion de hubs tels que définis par Guimerà et Amaral [49] (i.e. $z \geq 2,5$) est bien plus faible dans le réseau que nous étudions (0,35%) que dans les réseaux traités par Guimerà et Amaral [49] (2%). Ces seuils semblent donc sensibles aux changements dans la taille des données, dans la structure du réseau ou au changement de méthode de détection de communautés. Pour les actualiser, il est donc nécessaire de réappliquer toute la méthode de définition de ces seuils de façon empirique, ce qui ne semble pas adapté et fastidieux.

Connectivité externe. Nous considérons maintenant la mesure utilisée pour calculer la connectivité externe des noeuds. Le *coefficient de participation* encapsule plusieurs aspects de la connectivité externe d'un nœud mais se concentre principalement sur l'*hétérogénéité* de la distribution de ses liens, relativement aux communautés auxquelles il est connecté. Pourtant, il est possible de caractériser cette connectivité de deux autres manières, toutes deux importantes. Premièrement, on peut considérer sa *diversité*, c'est à dire le nombre de *communautés* concernées. Deuxièmement, il est possible de s'intéresser à son *intensité*, i.e. au nombre de *liens* concernés. Comme le montre la Figure 2.3, ces deux aspects ne sont pas pris en compte dans P. En effet, la connectivité externe du nœud central est différente sur chacun des trois graphes mais le coefficient de participation reste le même.

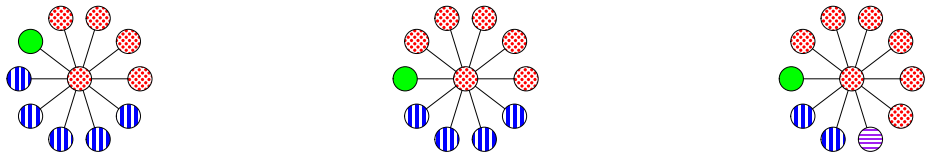

FIGURE 2.3 – Chaque forme représente une communauté. Dans chaque cas, le coefficient de participation du nœud central est 0,58.

Pour pallier cette limitation, nous proposons dans la prochaine Section de nouvelles mesures qui prennent en considération tous les aspects de la connectivité externe d'un noeud. Ceci nous ramène donc au problème des seuils estimés par Guimerà et Amaral [49]. Puisque nous ajoutons de nouvelles mesures, il est nécessaire d'établir de nouveaux seuils.

2.3 Nouvelle approche

Dans cette Section, nous décrivons les deux modifications que nous proposons pour résoudre les limitations de l'approche de Guimerà & Amaral [49]. Nous proposons tout d'abord des mesures supplémentaires permettant de mieux évaluer la connectivité externe des nœuds, et par ailleurs une méthode non-supervisée pour déterminer les rôles nodaux.

2.3.1 Aspects de la connectivité externe

Pour pallier les limitations de la participation externe, nous proposons deux nouvelles mesures permettant de quantifier la diversité et l'intensité. De plus, afin d'obtenir un ensemble cohérent de mesures, nous révisons également P. Puisque la taille des communautés varie -la distribution de leur taille suit généralement une loi de puissance [66]- et que l'on souhaite obtenir le rôle tenu par un noeud dans sa communauté, il semble cohérent de normaliser toutes ces mesures relativement à celles obtenues pour les autres

noeuds de leur communauté. Nous exprimons ainsi toute ces mesures sous forme de z-score. Par ailleurs, pour chacune des 4 mesures présentées, nous utilisons deux variantes, l'une considérant les liens entrants, l'autre les liens sortants. Nous obtenons donc 8 mesures, 2 pour caractériser la connectivité interne et 6 pour la connectivité externe.

Diversité. Notre mesure de *diversité*, notée $D(u)$, évalue le nombre de communautés différentes auxquelles le nœud u est connecté, indépendamment de la densité de ces connexions. Soit $\epsilon(u)$ le nombre de communautés, autres que la sienne, auxquelles le nœud u est connecté. Alors la diversité est définie comme le z-score de ϵ relativement à la communauté de u. C'est à dire qu'on l'obtient en substituant ϵ à f dans la Définition 2.4.

Intensité externe. L'*intensité externe* $I_{ext}(u)$ mesure la force de la connexion de u à des communautés externes, en terme de nombre de liens, et relativement aux autres nœuds de sa communauté. Soit $d_{ext}(u)$ le degré externe de u, correspondant au nombre de liens que u possède avec des nœuds n'appartenant pas à sa communauté. Nous définissons l'intensité externe comme le z-score du degré externe, c'est à dire qu'on l'obtient en substituant d_{ext} à f dans la Définition 2.4.

Hétérogénéité. L'*hétérogénéité* $H(u)$ quantifie la variation du nombre de connexions externes du nœud u d'une communauté à l'autre. Nous utilisons pour cela l'écart-type du nombre de liens externes que le nœud possède par communauté, que nous notons $\lambda(u)$. L'hétérogénéité est alors le z-score de λ, relativement à la communauté de u, et on l'obtient donc en substituant λ à f dans la Définition 2.4. Cette mesure a une signification très proche de celle du coefficient de participation P de Guimerà et Amaral [49]. Elle diffère en ce qu'elle est exprimée relativement à la communauté de u, et que les liens internes à cette même communauté sont exclus du calcul.

Intensité interne. Pour représenter la connectivité interne du nœud, nous conservons la mesure z de Guimerà et Amaral [49]. En effet, celle-ci est construite sur la base du z-score, et est donc cohérente avec les autres mesures définies pour décrire la connectivité externe. De plus, il n'est pas nécessaire de lui adjoindre d'autres mesures, car les notions d'hétérogénéité et de diversité n'ont pas de sens ici (puisqu'on considère seulement une seule communauté). Cependant, en raison de sa symétrie avec notre intensité externe, nous désignons z sous le nom d'*intensité interne*, et la notons $I_{int}(u)$.

Tout comme dans la Section 2.2.2, on constate que les seuils définis par Guimerà et Amaral [49] sont inutilisables étant données nos nouvelles mesures. Le problème des seuils et de la pertinence de la méthode empirique de Guimerà et Amaral [49] pour les définir se pose donc à nouveau. Nous proposons notre solution dans la prochaine Section.

2.3.2 Identification non-supervisée des rôles

Nous expliquons dans la Section 2.2.3 que les seuils définis de façon empirique sont considérés universels. Pourtant, une seule méthode de détection de communautés est utilisée par Guimerà et Amaral [49]. Si l'on observe la distribution de nos mesures de connectivité externe sur les réseaux du LFR à 5.000 noeuds (décrits dans la Section 2.1.3), on constate que celle-ci varie en fonction des algorithmes utilisés (Figure 2.4). Le paramètre μ du LFR qui définit la *netteté* des communautés a également un impact important. Puisque celles-ci sont basées sur le $z-$score comme le degré intra-module de Guimerà et Amaral [49], il nous semble peu réaliste dans ces conditions de considérer que les seuils observés en utilisant un seul algorithme sont universels. Par ailleurs, les distributions de nos mesures de connectivité externe ne sont pas semblables dans leurs versions entrante et sortante (Figure 2.5). Il semble donc que des seuils différents doivent être décidés pour chaque variante de nos mesures. Ceci confirme l'importance de tenir compte de l'orientation du réseau.

Nous considérons donc qu'il est nécessaire de changer la manière dont les rôles sont définis. En effet, ces expérimentations mettent en doute la fiabilité de la méthode proposée par Guimerà et Amaral [49] pour nos données. Nous proposons ainsi d'appliquer une méthode automatique de classification non supervisée. L'objectif d'une méthode de classification non supervisée est de regrouper automatiquement ensemble des observations qui se ressemblent en fonction de leurs attributs. Ici, les noeuds sont nos observations et les valeurs des mesures, leurs attributs. Dans un premier temps, nous calculons donc l'ensemble des mesures sur les données considérées. Ensuite, nous appliquons une analyse de regroupement. Chaque groupe ainsi identifié correspond a un rôle communautaire. Cette méthode présente l'avantage de ne pas être affectée par le nombre de mesures utilisé, et revient à ajuster les seuils pour le système considéré. Pour passer à l'échelle, l'analyse de regroupement a été menée au moyen d'une implémentation libre et distribuée de l'algorithme des k-moyennes [76]. L'algorithme des k-moyennes procède simplement :

1. Nous initialisons k observations comme centres des groupes ;
2. Chacune des observations est affectée au groupe dont le centre est le plus proche parmi les k ;
3. Nous recalculons chacun des centres de groupe : chaque centre prend comme valeur le barycentre des observations affectées à son groupe. Puis on réitère à partir de l'étape 2 jusqu'à convergence.

L'algorithme *converge* lorsque l'étape 2 ne change l'affectation d'aucune des observations. Les méthodes non-distribuées des k-moyennes, basées sur le calcul d'une unique matrice de distance, se sont révélées impossible à appliquer en raison de la quantité de mémoire nécessaire à la représentation de la matrice.

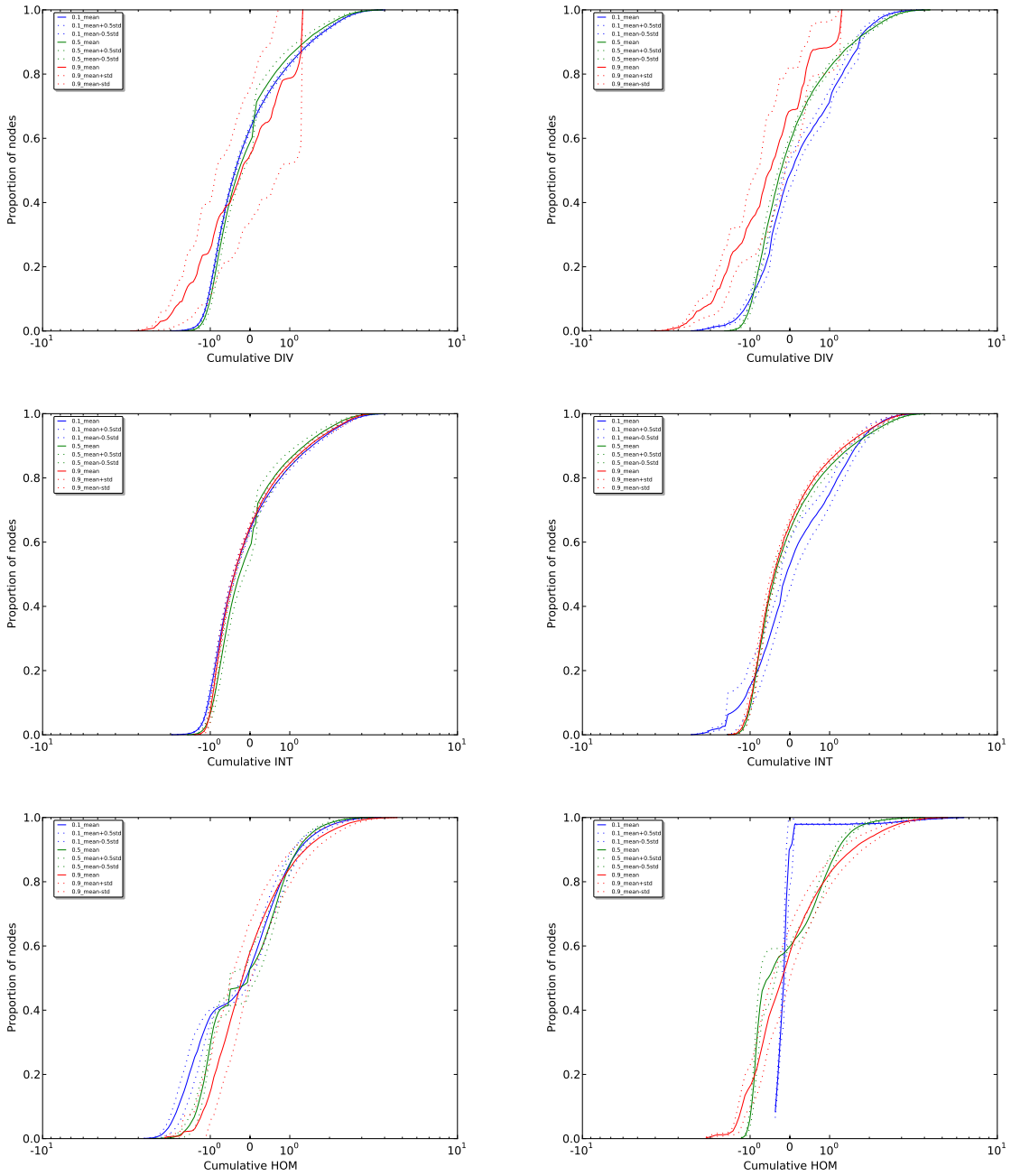

FIGURE 2.4 – À gauche, les distributions cumulatives obtenues avec notre algorithme de *Louvain orienté*. À droite, celles obtenues avec *OSLOM*. De haut en bas, les distributions sont celles de la diversité interne, l'intensité interne, l'homogénéité interne. En bleu, $\mu = 0,1$, en vert, $\mu = 0,5$ et en rouge, $\mu = 0,9$.

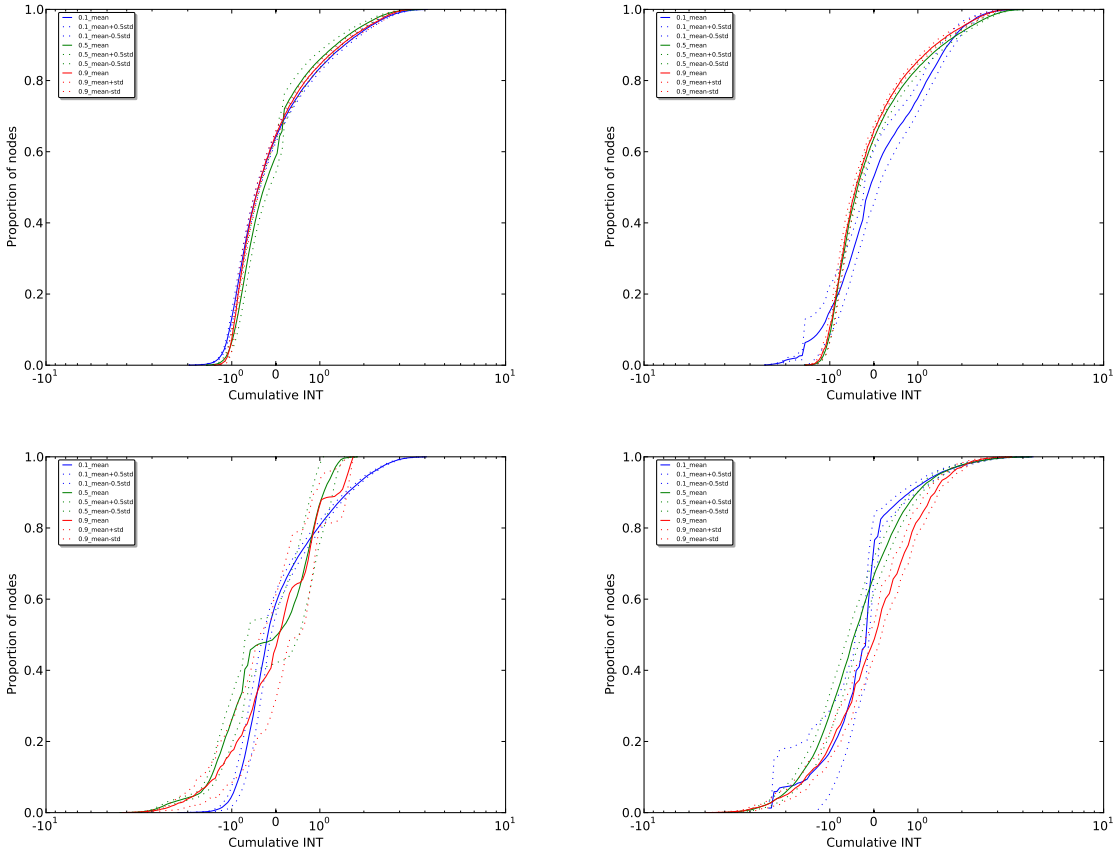

FIGURE 2.5 – À gauche, les distributions cumulatives obtenues avec notre algorithme de *Louvain orienté*. À droite, celles obtenues avec *OSLOM*. De haut en bas, les distributions sont celles de l'intensité interne puis de l'intensité externe. En bleu, $\mu = 0,1$, en vert, $\mu = 0,5$ et en rouge, $\mu = 0,9$.

Nous avons donc appliqué la version distribuée de cet algorithme pour des valeurs de k allant de 2 à 15, et avons sélectionné la meilleure partition d'après l'indice de Davies-Bouldin [29]. Cet indice est un ratio entre les distances des individus au sein du groupe et la séparation intra-groupes qui permet d'évaluer la qualité d'une partition.

Nous utilisons donc la méthode des k-moyennes pour partitionner notre ensemble de noeuds en plusieurs groupes auxquels nous affectons un rôle. Nous avons conscience de l'instabilité de notre méthode. En effet, les algorithmes de Louvain et des k-moyennes ont certes la capacité de passer à l'échelle de notre réseau, mais ils sont également non-déterministes. Cependant, nous n'avons pas pour objectif d'obtenir les meilleurs groupes en terme de séparation. Nous souhaitons simplement obtenir des groupes interprétables

au sens de la terminologie de Guimerà et Amaral [49] : distinguer les noeuds bien connectés dans leurs communautés de ceux qui ne le sont pas et distinguer les différents niveaux de connexions des noeuds avec les communautés externes. Nous obtenons de tels groupes dans la Section suivante. Une fois ces groupes obtenus, nous pouvons ainsi regarder où se placent les capitalistes sociaux et étudier leur connectivité dans le réseau.

2.4 Étude du réseau Twitter et rôles des capitalistes sociaux

Le réseau sur lequel nous avons travaillé est celui récolté par Cha et al.[20] décrit dans le Chapitre 1. Nous étudions la position des capitalistes sociaux détectés sur ce réseau dans le Chapitre 1 au sein des rôles établis. Pour faciliter l'interprétation des résultats, nous distinguons différentes catégories de capitalistes sociaux en fonction de deux de leurs caractéristiques topologiques : le *ratio* (voir Définition 1.7) et le degré entrant. Nous séparons ceux de faible degré (entre 500 et 10.000) et ceux de degré élevé (supérieur à 10.000).

2.4.1 Interprétation des rôles

Rôles attendus. Afin d'estimer la pertinence de notre analyse, nous présentons tout d'abord les rôles qui peuvent être attendus pour les capitalistes sociaux relativement aux mesures précédemment décrites. Ceci ne constitue pas une vérité de terrain mais certaines réalités concernant les capitalistes sociaux sont à prendre en compte dans l'analyse.

Nous nous attendons notamment à ce que le degré élevé des capitalistes sociaux joue un rôle important au regard de leur position dans le réseau (Chapitre 1). On suppose ainsi que ces derniers sont fortement connectés à leur communauté ou aux communautés externes, ou aux deux. Si les noeuds qui les représentent dans le réseau sont bien connectés aux communautés externes alors ces utilisateurs ont réussi à obtenir une grande visibilité sur une large frange du réseau, et pas seulement dans leur communauté. De plus, puisque nous tenons compte de l'orientation des liens dans le calcul de nos mesures, nous nous attendons à ce que les capitalistes sociaux soient discriminés en fonction de leur ratio (Définition 1.7). Nous pensons notamment que les capitalistes sociaux de degré élevé et avec un très faible ratio (capitalistes sociaux *passifs*) sont fortement connectés à leurs communautés mais aussi au reste du réseau. En effet, ces utilisateurs ont majoritairement un degré entrant très élevé.

En ce qui concerne les capitalistes sociaux de faible degré, nous n'avons en revanche aucune intuition quant à leurs rôles communautaires. L'étude de ces derniers et de la visibilité qu'ils leur accordent dans le réseau sera donc particulièrement intéressante.

Approche originale orientée. Par souci de complétude, nous utilisons d'abord l'approche originale proposée par Guimerà & Amaral [49] adaptée pour un réseau orienté (Section 2.2.2). Comme expliqué Section 2.3.2, les seuils définis par Guimerà & Amaral [49] ne semblent pas adaptés à des réseaux orientés. Nous adoptons donc l'approche par analyse de regroupement que nous détaillons Section 2.3.2. Une étude de corrélation montre que z^+ et z^- sont légèrement corrélés (avec un coefficient de corrélation $\rho < 0.3$) tandis que la corrélation est nulle pour toutes les autres paires de mesures. Ceci nous confirme donc l'importance d'utiliser l'orientation des liens du réseau dans ce genre d'études. Nous obtenons les groupes les mieux séparés avec $k = 6$. Par ailleurs, une ANOVA -*Analyse de la variance*- puis des *t-tests* montrent que toutes les paires de groupes sont significativement différentes statistiquement.

Une analyse de la distribution des capitalistes sociaux de degré élevé dans ces groupes nous montre que très peu d'entre eux sont placés dans le rôle de hub connecteur (Table 2.3). Cela reste cohérent avec notre analyse des rôles attendus de la Section 2.4.1. Cependant, la plupart de ces capitalistes sociaux sont considérés comme non-hubs périphériques ou ultra-périphériques. Plus de 60% des utilisateurs avec un faible ratio -capitalistes sociaux *passifs*- sont placés dans le groupe correspondant aux noeuds ultra-périphériques. Étant donné leur degré particulièrement élevé, cela semble très étonnant : ils devraient au moins être soit hubs, soit connecteurs. Une analyse qualitative montre que le *coefficient de participation* de ces noeuds est étonnamment bas. Cette mesure difficilement interprétable ne reflète donc pas bien la connectivité des noeuds avec leur communauté externe comme nous l'avons déjà souligné. Ceci aboutit donc à des incohérences dans la classification des noeuds en rôles.

Puisque l'approche originale ne peut être utilisée pour caractériser la participation externe de façon réaliste avec des noeuds, nous passons à la présentation des résultats avec notre approche.

Nouvelle approche. Considérons tout d'abord les mesures obtenues sur l'ensemble des données traitées. On observe des corrélations positives pour l'ensemble des paires de mesures, allant de valeurs proches de 0 à 0,9. Les deux variantes d'une même mesure (liens entrants contre liens sortants) sont peu corrélées, ce qui confirme une nouvelle fois l'intérêt de tenir compte de l'orientation dans notre étude. Trois mesures sont fortement corrélées : les intensités interne et externe et l'hétérogénéité (ρ allant de 0,78 à 0,92). Le lien entre les intensités interne et externe semble indiquer que les variations dans le degré total d'un nœud ont globalement le même effet sur ses degrés interne et externe. Autrement dit, la proportion entre ces deux types de liens ne dépend pas du degré du nœud. Le lien très fort observé entre hétérogénéité et intensité indique que seuls les nœuds de faible intensité sont connectés de façon homogène à des communautés externes, tandis

que les nœuds possédant de nombreux liens sont connectés de façon hétérogène. Cette observation est très intéressante : cela montre qu'il est quasiment impossible qu'un noeud de degré très élevé ait un *coefficient de participation* élevé. En effet, le score sera fortement pénalisé par la dissémination très hétérogène des liens.

En ce qui concerne l'analyse de regroupement, nous obtenons la meilleure séparation pour $k = 6$ groupes, dont la Table 2.4 donne les tailles. Nous avons caractérisé les groupes relativement à nos huit mesures, afin d'en identifier les rôles et de les comparer à ceux définis par Guimerà et Amaral [49]. La Table 2.5 contient les valeurs moyennes obtenues pour chaque mesure dans chaque groupe. Les ANOVA que nous avons réalisées ont révélé des différences significatives pour toutes les mesures. Les t-test ont montré que ces différences existaient entre tous les groupes, pour toutes les mesures.

Groupe	**Taille**	**Proportion**	**Rôle**
1	24.543.667	46,68%	Non-hub ultra-périphérique
2	304	< 0,01%	Hub orphelin
3	303.674	0,58%	Hub connecteur
4	11.929.722	22,69%	Non-hub périphérique (entrant)
5	10.828.599	20,59%	Non-hub périphérique (sortant)
6	4.973.717	9,46%	Non-hub connecteur

TABLE 2.4 – Tailles de groupes détectés, et rôles correspondants dans la terminologie de Guimerà et Amaral [49].

Dans le **Groupe** 1, toutes les mesures sont négatives mais proches de 0, à l'exception des deux variantes de la diversité, en particulier l'entrante, qui est proche de -1. Il ne peut pas s'agir de hub (nœud largement connecté à sa communauté), puisque l'intensité interne est négative. De même, les mesures externes sont très faibles ce qui montre qu'il ne s'agit pas non plus de nœuds qualifiés de connecteurs (ayant une connexion privilégiée avec d'autres communautés que la leur). On peut donc considérer que ce groupe correspond aux non-hubs ultra-périphériques. Ce groupe est le plus grand (il contient à lui seul 47% des nœuds), ce qui confirme la correspondance avec ce rôle, dont les nœuds constituent généralement la masse du réseau. Relativement au système modélisé, ces nœuds sont caractérisés par le fait qu'ils sont particulièrement peu suivis par les autres communautés.

Le **Groupe** 4 est extrêmement similaire au **Groupe** 1, à la différence que sa diversité entrante est de 0,69. Ces nœuds restent donc périphériques, car l'intensité externe est toujours négative, mais ils reçoivent néanmoins des liens provenant d'un nombre relativement élevé de communautés. Autrement dit, ils sont suivis par peu d'utilisateurs externes, mais ceux-ci sont situés dans un grand nombre de communautés distinctes.

G	I_{int}		D		I_{ext}		H	
1	−0,12	−0,03	−0,55	−0,80	−0,09	−0,04	−0,12	−0,06
2	94,22	311,27	7,18	88,40	113,87	283,79	112,79	285,57
3	5,52	1,40	5,60	3,10	5,28	1,43	6,76	2,34
4	−0,04	0,00	−0,37	0,69	−0,07	0,00	−0,10	−0,01
5	−0,03	−0,01	0,60	0,19	−0,03	−0,02	−0,04	−0,02
6	0,48	0,12	1,96	1,70	0,35	0,12	0,53	0,19

TABLE 2.5 – Mesures moyennes obtenues pour les 6 groupes. Pour chaque mesure, deux valeurs sont indiquées, correspondant respectivement aux deux variantes : liens sortants et entrants.

Autrement dit, ils sont suivis par peu d'utilisateurs externes, mais ceux-ci sont situés dans un grand nombre de communautés distinctes.

Le **Groupe** 5 est lui aussi très proche du Groupe 1, mais la différence est cette fois que les deux variantes de la diversité sont positives, avec une diversité sortante de 0,60. À l'inverse du **Groupe** 4, on peut donc dire ici que les utilisateurs concernés suivent (avec une faible intensité) des utilisateurs situés dans un grand nombre de communautés différentes. Les **Groupes** 4 **et** 5 sont respectivement le deuxième (23%) et troisième (21%) plus grands groupes en terme de taille, ce qui porte le total des nœuds périphériques à 91%.

Toutes les mesures sont positives dans le **Groupe** 6. L'intensité interne reste proche de 0, donc on ne peut toujours pas parler de hub, même si ces nœuds sont mieux connectés à leur communauté que ceux des groupes précédents. L'intensité externe est elle aussi faible, mais le fait qu'elle soit positive, à l'instar des autres mesures externes, semble suffisant pour considérer ces nœuds comme des connecteurs (relativement bien reliés à d'autres communautés). La diversité est relativement élevée, aussi bien pour les liens entrants que sortants ($D > 1,7$). Ces nœuds sont donc plus fortement connectés à leur communauté mais aussi à l'extérieur, et avec une plus grande diversité. Il s'agit du quatrième plus gros groupe, représentant 9,5% des nœuds.

Toutes les mesures du **Groupe** 3 sont largement positives : supérieures à 1,4 pour celles basées sur les liens entrants, et supérieures à 5,2 pour les liens sortants. L'intensité interne élevée permet d'associer ce groupe au rôle de hub. Les valeurs externes montrent en plus que ces nœuds sont connectés à de nombreux nœuds présents dans de nombreuses autres communautés. Toutefois, les liens sortants sont plus nombreux, ces nœuds correspondent donc à des utilisateurs plus suiveurs que suivis. Ce groupe ne représente que 0,6% des nœuds, il s'agit donc d'un rôle bien plus rare que ceux associés aux groupes précédents. Cette observation est encore plus caractéristique du **Groupe** 2, qui représente bien moins

de 1% des nœuds. Toutes les mesures y sont particulièrement élevées, la plupart dépassant 100. Pour une mesure donnée, la variante concernant les liens entrants est toujours largement supérieure, ce qui signifie que les utilisateurs représentés par ces nœuds sont particulièrement suivis, et donc visibles. Nous associons ce groupe au rôle de hub orphelin.

Nous avons affecté à tous les groupes des rôles et les avons présentés dans la Table 2.4. Avant de nous intéresser à la position des capitalistes sociaux au sein de ces groupes, nous étudions la relation de hiérarchie qui existe entre ces groupes.

Hiérarchie de groupes. La Figure 2.6 est une représentation simplifiée des connexions qui existent entre les noeuds de chacun des groupes qui représentent un rôle communautaire distinct.

Pour plus de 70% des liens sortants des noeuds considérés comme périphériques ou ultra périphériques - Groupes 1,4 et 5, on observe à l'autre extrémité des hubs orphelins ou connecteurs -Groupes 2 et 3. Si l'on interprète cette observation, cela signifie que les utilisateurs les plus isolés sur Twitter s'abonnent principalement aux utilisateurs ayant déjà un grand nombre d'abonnés. Ceux-ci sont probablement des comptes de célébrités, de médias ou en tout cas, des comptes qui disposent d'une grande visibilité.

Les noeuds connecteurs -Groupes 3 et 6- sont principalement connectés entre eux. Ce sont les groupes qui ont les connexions les plus fortes puisqu'elles représentent près de 43% des connexions du réseau. Pourtant, ces groupes ne sont pas les plus grands. Ils sont aussi largement connectés aux autres groupes et forment en quelque sorte le squelette du réseau.

Les hubs orphelins -Groupe 2- sont massivement suivis par des non-hubs, qui représentent 38% (Groupe 1), 43% (Groupe 4), 19% (Groupe 5) et 8% (Groupe 6) des liens sortants de ces groupes. Nous remarquons également que les liens sortants des hubs orphelins ciblent ces mêmes groupes : 9% vont vers le Groupe 1, 20% vers le Groupe 4, 22% vers le Groupe 5 et 41% vers le Groupe 6. Ceci signifie que les utilisateurs les plus visibles et les plus suivis sur Twitter s'abonnent à des utilisateurs qui semblent bien moins populaires. On aurait pu s'attendre à ce qu'une hiérarchie se forme, les ultra-périphériques se connectant aux périphériques, eux mêmes connectés aux hubs connecteurs etc. Ce n'est pas le cas. Les noeuds (ultra-)périphériques ne sont que très marginalement connectés à des noeuds de même rôle. Ils sont massivement connectés aux noeuds connecteurs et aux hubs. Enfin, les hubs connecteurs et orphelins ne s'abonnent pas entre eux.

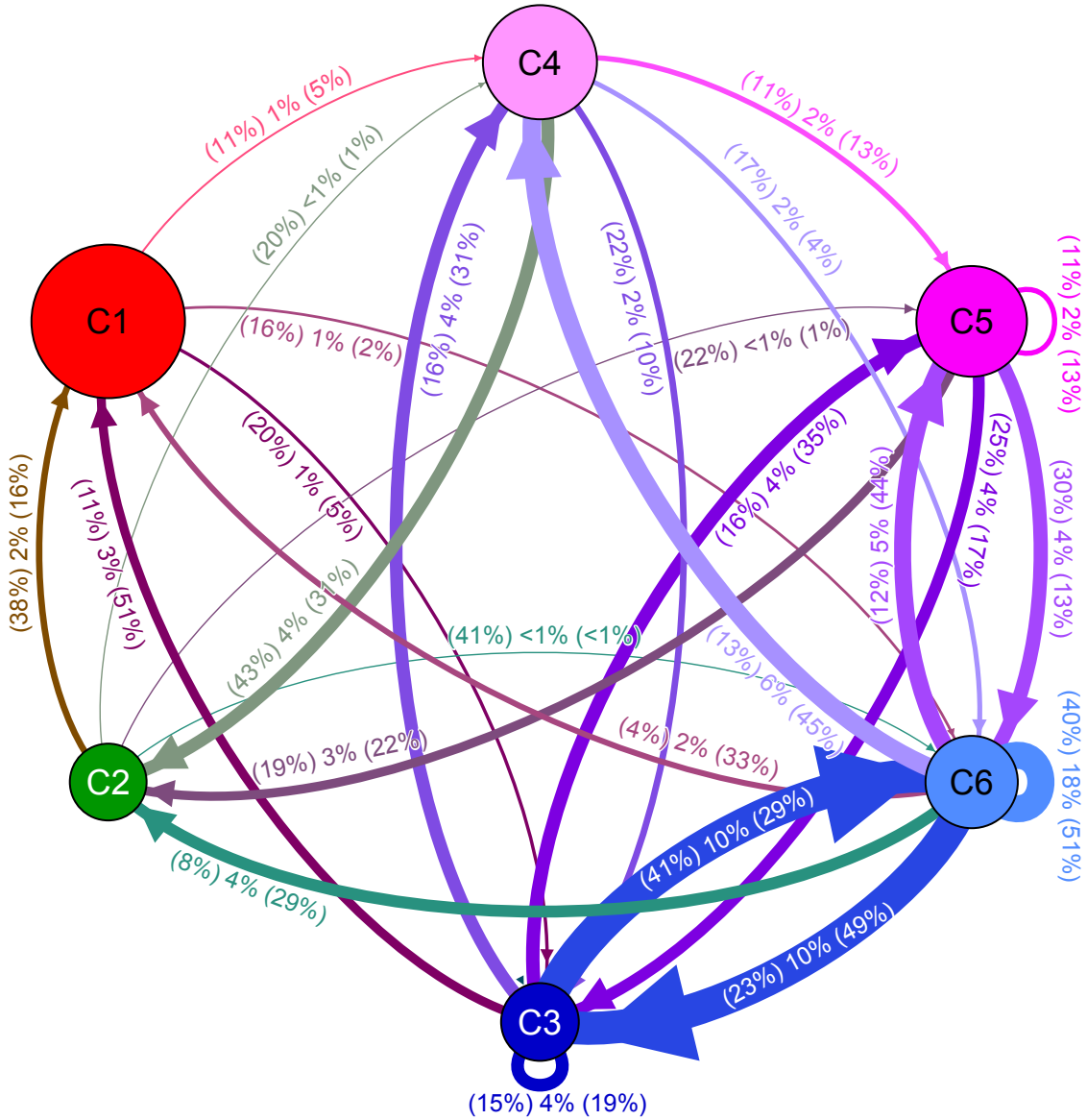

FIGURE 2.6 – Connexions entre groupe. Un sommet C_i correspond au Groupe i de la Table 2.4. Un arc (i, j) représente l'ensemble de liens qui connectent les noeuds du Groupe i aux noeuds du Groupe j. Ces arcs possèdent 3 labels. Chacun d'entre eux décrit quelle proportion de liens l'arc représente relativement aux liens sortants du Groupe i, à tous les liens du réseau et aux liens entrants du Groupe j. L'épaisseur de l'arc est proportionnelle à la seconde valeur. La taille du sommet correspond à la taille du Groupe. Pour une meilleure lisibilité, les arcs qui représentent moins de 1% de ceux du réseau et de 10% du Groupe ne sont pas visibles sur la Figure.

2.4.2 Positionnement des capitalistes sociaux

Avec la méthode définie dans le Chapitre 1, nous détectons près de 160.000 capitalistes sociaux. Nous étudions ici leur positionnement dans les 6 groupes identifiés par la méthode des k-moyennes. De plus, nous affinons notre analyse en structurant les capitalistes sociaux en différents groupes. Tout d'abord via le ratio, qui nous permet de mettre en évidence les comportements **FMIFY** et **IFYFM**. Ensuite, en utilisant le degré de ces utilisateurs. En effet, les capitalistes sociaux ayant accru le plus efficacement leur nombre d'abonnés sont susceptibles d'avoir un placement ou un rôle différent au sein des communautés.

Chaque tableau présente ainsi sur la première ligne la proportion de capitalistes sociaux du réseau qui sont contenus dans chaque groupe, et sur la deuxième la proportion de nœuds du groupe qui sont des capitalistes sociaux.

Capitalistes sociaux de faible degré entrant (> 500 **et** < 10.000**).** Ces capitalistes sociaux se regroupent dans trois Groupes : 3, 5 et 6. Les nœuds du **Groupe** 3 sont des hubs connecteurs qui ont en particulier tendance à suivre plus d'utilisateurs du réseau que la normale. Même si le degré entrant des capitalistes sociaux est considéré comme faible ici, il reste élevé relativement au degré moyen du reste du réseau.

Ratio	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**
< 1	$0,01\%$	$0,00\%$	$\mathbf{23,10\%}$	$3,42\%$	$\mathbf{18,28\%}$	$\mathbf{55,19\%}$
	$< 0,01\%$	$0,00\%$	$3,71\%$	$0,14\%$	$0,08\%$	$0,54\%$
> 1	$0,03\%$	$0,00\%$	$\mathbf{18,78\%}$	$0,48\%$	$\mathbf{14,31\%}$	$\mathbf{66,40\%}$
	$< 0,01\%$	$0,00\%$	$\mathbf{6,61\%}$	$< 0,01\%$	$0,14\%$	$1,43\%$

TABLE 2.6 – Répartition des capitalistes sociaux de faible degré dans les différents groupes.

Cela semble donc cohérent de voir qu'un grand nombre de capitalistes sociaux est plus connecté à la fois à leur communauté mais également aux autres communautés. Il semble également cohérent d'observer que les capitalistes sociaux de type **IFYFM** (dont le degré sortant est supérieur au degré entrant) sont près de deux fois plus présents dans ce groupe que les autres. La diversité sortante élevée du **Groupe** 3 nous apprend également que ces capitalistes sociaux ont tendance à ne pas cibler uniquement leur communauté même s'ils y sont bien connectés, mais à appliquer leurs méthodes à travers de nombreuses communautés du réseau.

On observe que la large majorité des capitalistes sociaux de faible degré se place au sein du Groupe 6 (non-hub connecteur). Ces nœuds, qui sont légèrement plus connectés au sein de leur communauté et avec l'extérieur que la moyenne, ont en revanche une diversité

bien plus élevée. Les capitalistes sociaux qui s'y situent semblent ainsi avoir débuté l'application de leurs méthodes, en créant des liens avec de nombreuses autres communautés.

Enfin, on retrouve une petite proportion de capitalistes sociaux de faible degré dans le **Groupe** 5, groupe de nœuds non-hubs périphériques. Un certain nombre de capitalistes sociaux sont ainsi isolés au sein de leur communauté et de l'extérieur.

Capitalistes sociaux de degré entrant élevé (> 10.000)**.** Les capitalistes sociaux de degré élevé se placent presque exclusivement dans les **Groupes** 2 et 3. Ces groupes contiennent des nœuds hubs connecteurs et orphelins.

Ratio	**G1**	**G2**	**G3**	**G4**	**G5**	**G6**
$< 0,7$	0,00%	**12,14%**	**87,29%**	0,00%	0,00%	0,57%
	0,00%	**21,05%**	0,15%	0,00%	0,00%	$< 0,01\%$
$> 0,7$ et < 1	0,00%	1,55%	**95,64%**	0,00%	0,00%	2,81%
	0,00%	**7,24%**	0,45%	0,00%	0,00%	$< 0,01\%$
> 1	0,00%	0,03%	**97,99%**	0,00%	0,00%	1,98
	0,00%	0,33%	1,22%	0,00%	0,00%	$< 0,01\%$

TABLE 2.7 – Répartition des capitalistes sociaux de degré élevé dans les différents groupes.

Cela semble cohérent avec les degrés élevés de ces nœuds. Ceux-ci sont naturellement plus connectés avec leurs communautés et avec l'extérieur que les autres nœuds. On constate que les nœuds classés dans le **Groupe** 2 sont ceux de ratio inférieur à 1 et particulièrement ceux de ratio inférieur à $0,7$ ayant beaucoup plus d'abonnés que d'abonnements. Cela correspond bien à la définition du rôle donné par nos mesures qui montre que ce groupe de nœuds est suivi par un grand nombre de nœuds provenant d'une large variété de communautés.

Résumé. On observe ainsi que notre approche permet d'établir une nette séparation entre capitalistes sociaux de faible degré, majoritairement connecteurs et non-hubs et ceux de degré élevé, classé comme hubs. Par ailleurs, les rôles obtenus permettent également de discriminer les utilisateurs de ratios différents. Les capitalistes sociaux de degré élevé et de ratio inférieur à 1 sont par exemple les seuls à appartenir au groupe des hubs orphelins. Ce n'était pas le cas avec l'approche originale adaptée aux graphes orientés. Enfin, notre approche permet de mieux décrire les différents rôles obtenus grâce aux trois mesures utilisées pour caractériser la connectivité du nœud aux communautés auxquelles il n'appartient pas. Pour conclure, la grande majorité des capitalistes sociaux de degré élevé sont bien connectés à leur communauté mais également au reste du réseau. Ceci montre l'efficacité de leurs méthodes qui leur permettent d'accéder à une grande visibilité. Par ailleurs, les capitalistes sociaux de faible degré s'abonnent à un grand nombre

d'utilisateurs en dehors de leur communauté. Seule une faible proportion d'entre eux est périphérique. Pour la majeure partie d'entre eux, ils parviennent donc à être visibles d'un grand nombre de communautés.

Limites de notre approche. Comme nous l'avons déjà mentionné, les algorithmes de Louvain et des k-moyennes ont certes la capacité de passer à l'échelle de notre réseau mais ils sont également non-déterministes. Ceci conduit à rendre très instable notre méthode. Cependant, nous avons fait ces choix pour être capable d'étudier la visibilité des capitalistes sociaux dans tout le réseau **Twitter** et obtenir les résultats précédemment résumés. Dans la Conclusion, nous discutons des perspectives de notre travail pour stabiliser cette méthode et l'appliquer à des réseaux de plus petite taille.

CHAPITRE

3 Mesures d'influence : le cas des capitalistes sociaux

Le **Larousse** définit ainsi l'influence :

> "Pouvoir social et politique de quelqu'un, d'un groupe, qui leur permet d'agir sur le cours des événements, des décisions prises".

À partir de cette définition, il semble évident que quantifier l'influence d'un utilisateur **Twitter** au sein du réseau est un problème difficile. La création de challenges comme celui de l'*author ranking* à **RepLab2014** en témoigne. Néanmoins, certaines entreprises telles que **Klout** ou **KonaRed Corporation** proposent des solutions sous forme de score d'influence : les scores **Kred** [62] et **Klout** [56]. Ces solutions sont utilisables en *boîte-noire*, leur fonctionnement n'est pas connu. Les deux sociétés expliquent néanmoins utiliser fortement les interactions entre utilisateurs pour quantifier leur influence. Ainsi, être souvent retweeté ou mentionné conduit à un score d'influence plus élevé.

Dans ce Chapitre, nous montrons que les capitalistes sociaux parviennent à obtenir des scores d'influence élevés grâce aux techniques qu'ils emploient. Ces techniques incitent en effet les capitalistes sociaux à s'abonner entre eux, et à se retweeter et se mentionner les uns les autres. Nous mettons ensuite en place une méthode basée sur de l'*apprentissage supervisé* pour discriminer efficacement les *capitalistes sociaux* des autres utilisateurs dits *réguliers* à partir d'un grand nombre d'informations -et plus uniquement topologiques. Nous en étudionslsa robustesse, puis nous produisons à partir de celle-ci une mesure d'influence qui pondère le score Klout. Enfin, nous décrivons une application qui implémente une version en ligne de cette méthode.

3.1 Le problème de l'influence

L'augmentation du nombre d'abonnés, l'explosion du nombre de tweets par jour et l'importance de **Twitter** dans l'actualité ont naturellement amené les entreprises et les académiques à s'intéresser à la notion d'influence sur le réseau **Twitter** [2, 20, 103, 112, 125]. De nombreux paramètres peuvent être pris en compte pour mesurer l'influence sur **Twitter**. La plupart considèrent 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 forte [20]. Ces paramètres ne sont en général pas jugés comme de même importance. Par ailleurs, certains paramètres dérivés sont considérés, tels que les ratios *Abonnements/Abonnés* (Définition 1.7), *Retweet et Mention* et *Interactions* [2]. Différents outils ont ainsi été proposés par 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** [56], **Kred** [62], **Tweet Grader** [114] et **Twitalyzer** [115]. Dans tous les cas, l'algorithme utilisé pour calculer le score d'un utilisateur est gardé secret même si **Klout** et **Kred** fournissent quelques informations non détaillées (voir Anger & Kittl [2]). 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, la portée). Selon **Kred**, l'influence augmente *lorsque quelqu'un vous retweete, vous mentionne ou vous répond.*

Récemment, Messias et al. [84] ont étudié cette problématique de l'influence sur **Twitter**. Ils ont ainsi créé un *bot* qui tweetait automatiquement à propos de sujets populaires. Ce dernier obtint 500 abonnés et des scores **Twitalyzer** et **Klout** élevés. Même s'il semble évident que ce compte ne dispose pas d'une réelle influence, il est intéressant de constater que les outils tels que **Twitalyzer** et **Klout** affirment le contraire. De plus, ce compte peut facilement être détecté comme automatique en utilisant les sources utilisées pour poster les tweets ou le rythme régulier des envois de messages [23].

Dans ces travaux, le problème de l'influence des capitalistes sociaux n'est pas traité. Pourtant, nous constatons que ces utilisateurs parviennent à gagner un grand nombre d'abonnés et à être largement retweetés. Nous nous intéressons donc à cette problématique de l'influence des capitalistes sociaux. Nous montrons tout d'abord que les outils actuels ne sont pas capables de distinguer les capitalistes sociaux des utilisateurs réguliers et qu'ils leurs accordent ainsi à tort des scores d'influence potentiellement élevés. Pour ce faire, nous décrivons tout d'abord les principaux outils de mesure d'influence. Nous étudierons ensuite un jeu de données **Twitter** obtenu sur des hashtags dédiés au capitalisme social tels que *#TeamFollowBack* (Section 3.3). En étudiant ces utilisateurs certifiés comme capitalistes sociaux, nous observons que certains d'entre eux sont considérés comme très influents par les outils de mesure tels que **Klout** et **Kred** (Section 3.2.2).

3.2 Mesures d'influence sur Twitter

3.2.1 Klout, Kred and Twitalyzer

Puisque la notion d'influence et son calcul restent assez flous, de nombreux paramètres peuvent être utilisés pour mesurer cette dernière : nombre d'abonnés, retweets, mentions, favoris par exemple. Afin de réaliser des scores plus élaborés, il est également possible de combiner certains ratios. Le plus simple et le plus intuitif est le ratio du *nombre d'abonnements sur le nombre d'abonnés* (Définition 1.7). 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 il est donc 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 : *Retweet et Mention* et *Interaction* [2]. Le premier compte le nombre de tweets postés par un utilisateur qui sont retweetés ou mènent à une conversation divisé 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 [2] définissent ainsi le *Social Networking Potential* d'un utilisateur **Twitter** comme la moyenne de ces deux ratios.

Dans cette Section, nous étudions les outils disponibles en ligne et largement utilisés par les utilisateurs **Twitter** et entreprises pour mesurer (et éventuellement) améliorer leur influence dans le réseau. Ceci est en particulier le cas de **Klout** [56], **Kred** [62] et **Twittalyzer** [115], bien que ce dernier ait stoppé son activité commerciale en septembre 2013 (l'outil en ligne reste cependant disponible).

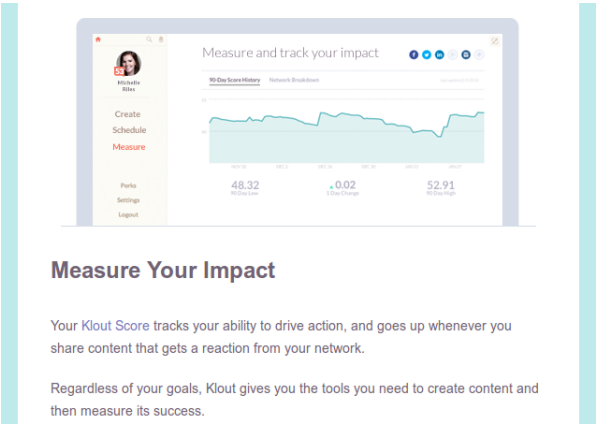

FIGURE 3.1 – Mesurez votre impact avec **Klout**.

Remarquons que **Klout** a été racheté par Lithium Technologies pour 200 millions de dollars en Mars 2014 [106]. Puisque par la suite, nous nous concentrons particulièrement sur **Klout**, nous présentons l'outil plus en détails.

Klout. Lors de l'inscription, **Klout** se présente comme un outil qui mesure la capacité d'un utilisateur à inciter à l'action (voir Figure 3.1). Ainsi **Klout** confirme ce qui a été dit : le score Klout augmente lorsque le réseau d'un utilisateur réagit aux contenus qu'il partage. Une fois connecté à l'outil, l'utilisateur est mené à un tableau de bord (Figure 3.2). Sur ce tableau de bord, l'utilisateur peut observer son score d'influence ainsi que son évolution, les différents réseaux auxquels il a connecté son compte **Klout** et le score d'impact de ses différents posts sur ces réseaux.

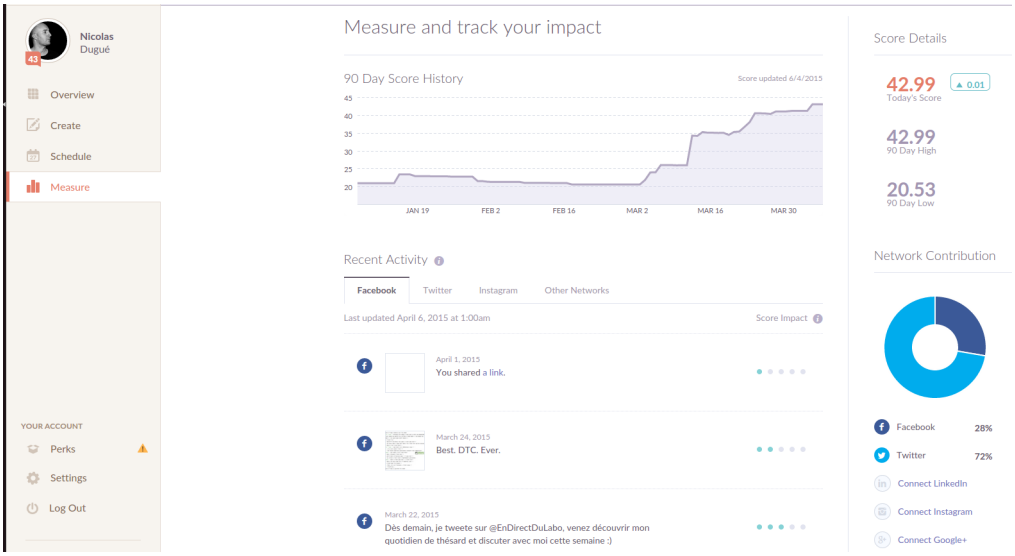

FIGURE 3.2 – Le tableau de bord du compte **Klout** qui nous sert d'exemple. Le score d'influence de 42,99 est en haut à droite, son évolution au centre en haut, les réseaux connectés au compte **Klout** en bas à droite et le score d'impact des contenus partagés au centre.

Nous observons ainsi en bas à droite de la Figure 3.2 que ce compte **Klout** est connecté à **Twitter** et **Facebook**. Par ailleurs, nous remarquons que le statut **Facebook** situé en bas de la Figure 3.2 a un score d'impact plus élevé. Ceci s'explique par le fait qu'il a plus été *liké* que les autres. Enfin, nous pouvons observer l'augmentation du score Klout de ce compte à partir de mi-Mars. Cette augmentation n'est pas dûe à l'accroissement du charisme de l'utilisateur, ni de l'intérêt de ses messages, il coïncide avec deux évènements : l'augmentation de son nombre d'interactions, notamment avec le compte **Twitter** *@EnDirectDuLabo* et la connexion de **Klout** à son compte **Facebook**.

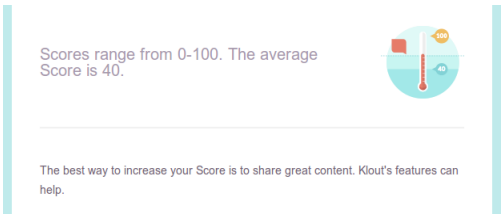

FIGURE 3.3 – Score **Klout** moyen d'après **Klout**.

Klout mesure donc l'influence d'un utilisateur en se basant sur les principaux réseaux sociaux (par exemple **Facebook**, **Linkedin** et **Instagram**) et pas uniquement sur **Twitter**. L'outil utilise également la page **Wikipedia** d'un utilisateur. 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. Par ailleurs, **Klout** fournit un score moyen auquel se comparer. Comme l'indique la Figure 3.3, celui-ci est de 40.

Au cas où un utilisateur jugerait son score trop bas, l'outil **Klout** peut l'aider à partager des contenus susceptibles de faire augmenter son score. En effet, comme le montre la Figure 3.4, l'outil en ligne propose également un outil de recommandation, notamment d'utilisateurs auxquels s'abonner et de contenus à poster.

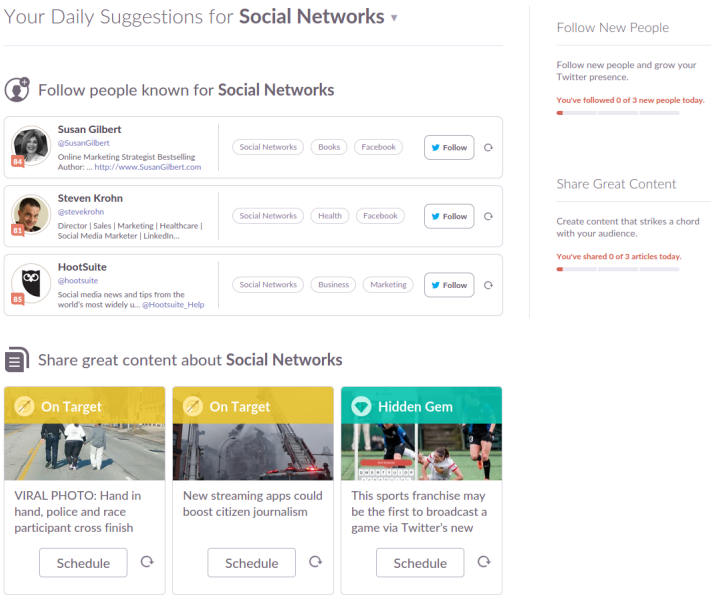

FIGURE 3.4 – Recommandations d'utilisateurs auxquels s'abonner et de contenus à poster dans le domaine *Social networks*.

On peut ainsi se questionner sur la légitimité d'un outil de mesure d'influence qui propose le contenu à tweeter pour devenir influent. Si être influent revient à tweeter du contenu populaire, alors n'importe quel compte automatisé utilisant l'outil de recommandation peut le devenir. Pourtant, **Klout** continue à être une référence dans le monde du marketing ou du *community management*, comme en attestent des articles tels que celui du journaliste John Boitnott publié en Février 2015 [58] ou encore ce guide réalisé par la société *Simply Measured* sur l'utilisation de **Klout** pour détecter les influenceurs en Juin 2014 [57]. Notons que *Simply Measured* a notamment pour clients **Adidas**, **Pepsi**, **American Express** et **Samsung** [35].

Nous allons voir dans la prochaine Section que **Klout**, **Kred** et **Twitalyzer** accordent par ailleurs des scores d'influence élevés aux capitalistes sociaux. Nous expliquerons pourquoi et donnerons des exemples illustratifs.

3.2.2 L'impact du capitalisme social

Avec la promesse de s'abonner en retour aux utilisateurs qui les suivent, les capitalistes sociaux parviennent à accroître leur nombre d'abonnés efficacement. Cependant, comme mentionné précédemment, ceci ne conduit pas nécessairement à une augmentation de leur score d'influence puisque ce paramètre n'est pas considéré comme important par les 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 elles considérées comme des indicateurs d'influence par **Klout** et **Kred**. 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 Figure 3.5).

FIGURE 3.5 – Timelines des utilisateurs *@1000sFollowrs60* et *@TeamFollowBack*.

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**, **Kred** et **Twitalyzer** pour des capitalistes sociaux certifiés, soit extraits du jeu de données

collecté en utilisant les hashtags décrits dans le Chapitre 1, soit à cause de la biographie ou du *screen name* explicites de ces utilisateurs. Rappelons que les utilisateurs avec un indice de chevauchement supérieur à 0,74 sont considérés comme capitalistes sociaux (Chapitre 1). Cette condition est vérifiée pour tous les utilisateurs de la Table 3.1. Ces capitalistes sociaux sont considérés comme influents par les trois mesures, malgré des comportements, des biographies ou même des noms parfois très explicites.

Identifiant	**Abonnements**	**Abonnés**	$\mathbf{I}_c$	**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

TABLE 3.1 – Scores d'influence des capitalistes sociaux extraits de nos jeux de données : **Klout**, **Kred** et **Twitalyzer**. La colonne I_c contient l'Indice de chevauchement.

La Table 3.2 présente des résultats similaires pour des capitalistes sociaux détectés par la méthode du Chapitre 1. 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 la Table 3.1.

Identifiant	**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

TABLE 3.2 – Scores d'influence de capitalistes sociaux détectés par note méthode : **Klout**, **Kred** et **Twitalyzer**.

Néanmoins, ils restent de loin bien supérieurs à la moyenne (qui est annoncée à 40 par **Klout**). De plus, leurs scores **Kred** et **Twitalyzer** sont également relativement élevés.

Pour compléter les observations précédentes, nous comparons dans la Figure 3.6 les 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. [84]).

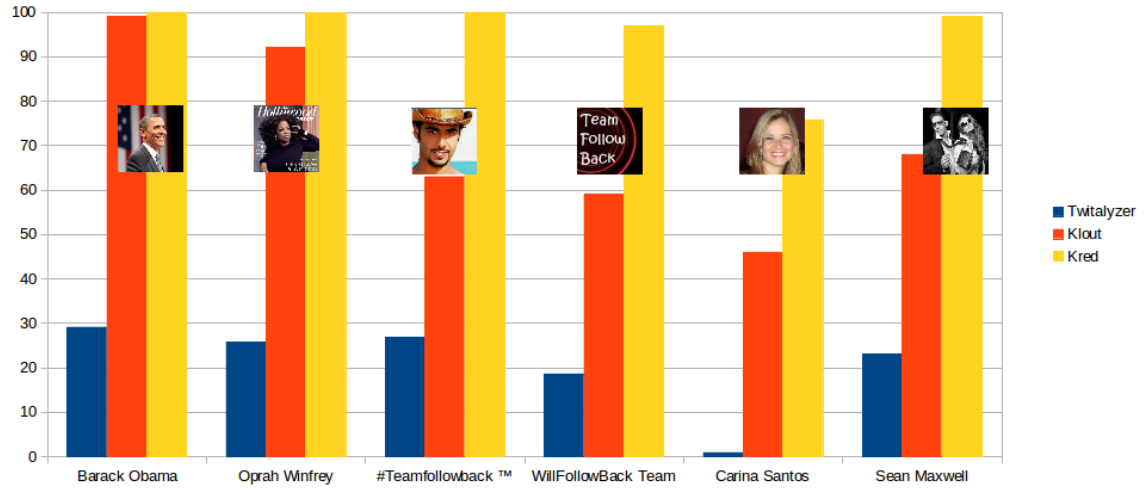

FIGURE 3.6 – Comparaison des trois mesures pour les comptes de **Barack Obama** et **Oprah Winfrey** et ceux de capitalistes sociaux avérés.

Les capitalistes sociaux ont des scores **Kred** et **Twitalyzer** similaires à ceux des comptes de référence quand 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, mais également celle de **Wikipedia**. **Barack Obama** et **Oprah Winfrey**, qui ont leur page **Wikipedia** bien documentée et consultée, accèdent donc naturellement 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 noms et biographies 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 donc 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

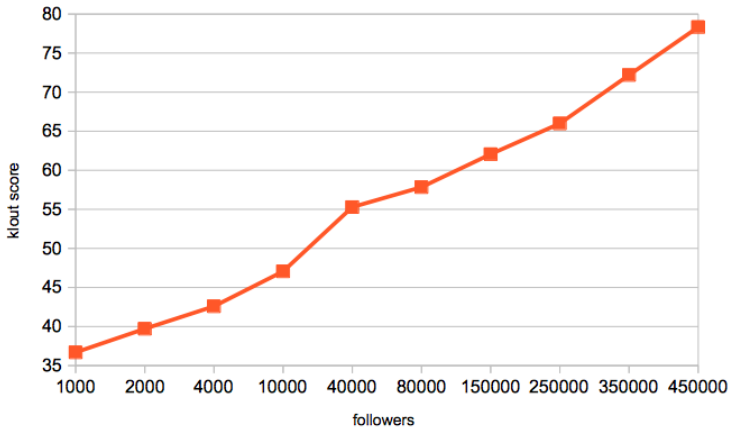

FIGURE 3.7 – Score **Klout** moyen des capitalistes sociaux ayant au moins un certain nombre d'abonnés (indiqué en abscisse).

mentions comme plus important que le nombre d'abonnés. Cependant, comme le montre la Figure 3.7, 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 favorise la course aux abonnés sur **Twitter**. Si l'on s'en tient au score **Klout** tel quel, les contenus partagés par les capitalistes sociaux sur **Twitter** sont à mettre en avant sur les moteurs de recherche ou pages dédiés aux hashtags dans les sections *Top*. Il nous semble important d'être capable de différencier 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**. Nous posons les premières briques d'un tel système dans la Section suivante en collectant et analysant un jeu de données consitué d'utilisateurs *réguliers* et de capitalistes sociaux.

3.3 Jeu de données

Exemples positifs. Dans le but de constituer un jeu de données de capitalistes sociaux certifiés, nous avons récolté en Février 2014 des tweets postés sur les hashtags *#TeamFollowBack*, *#instantfollowback* et *#teamautofollow*. Nous avons identifié les utilisateurs ayant posté au moins trois tweets avec ces hashtags en quelques jours. Nous avons ainsi obtenu un échantillon d'environ 25.000 capitalistes sociaux.

Exemples négatifs. La première étape pour obtenir des exemples négatifs consiste à échantillonner aléatoirement **Twitter**. 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 a 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 nos travaux [32], les capitalistes sociaux représentent 0,2% du réseau. Ainsi, même si choisir les abonnements d'utilisateurs aléatoires favorise l'obtention 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).

Représentation formelle des données. Soit un jeu de données de n exemples à p attributs. On note l'ensemble des exemples $X = \{X_1, \ldots, X_n\}$ et l'attribut d'un exemple X_i, $i \in \{1, \ldots, n\}$ est noté $X_{i,j}$ avec $j \in \{1, \ldots, p\}$. On représente ainsi ce jeu de données par la matrice

$$M_{n,p} = \begin{bmatrix} X_{1,1} & \cdots & X_{1,p} \\ \vdots & \ddots & \vdots \\ X_{n,1} & \cdots & X_{n,p} \end{bmatrix}$$

Attributs. Nous avons obtenu pour chacun de nos exemples un ensemble d'informations utilisé pour les caractériser. Ces informations sont classées en différentes catégories (Table 3.3). Nous souhaitons mentionner que toutes ces données sont obtenues par l'intermédiaire de l'API REST fournie par **Twitter** [119]. 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 par ailleurs que les restrictions imposées par **Twitter** quant à l'utilisation de leur API nous poussent à considérer uniquement les 200 derniers tweets des utilisateurs (Annexe C).
Certaines de ces informations caractéristiques telles que les sources utilisées pour poster les tweets et le nombre d'URLs moyen par tweet sont efficaces pour séparer des comptes humains de comptes automatisés d'après Chu et al. [23]. Cependant, nous avons montré (Section 1.2) 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.

CATÉGORIE	CARACTÉRISTIQUES
Activité	Nombre de : 1. Tweets 2. Listes **Twitter** qui contiennent l'utilisateur 3. Tweets favoris
Topologie locale	Nombre de : 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 tweet 9. URLs par tweet 10. Mentions par tweet
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 14. Nombre moyen de seconde entre deux tweets
Sources	Proportion d'utilisation de : 15. Application **Twitter** officielle 16. Outil de gestion de compte 17. Outil d'abonnement ou de désabonnement automatique 18. Outil d'envoi de tweet automatique 19. Autres applications (**Vine**, **Wiki**, **Soundcloud**, ...) 20. Applications smartphones ou tablettes

TABLE 3.3 – Description des différentes informations de compte étudiées.

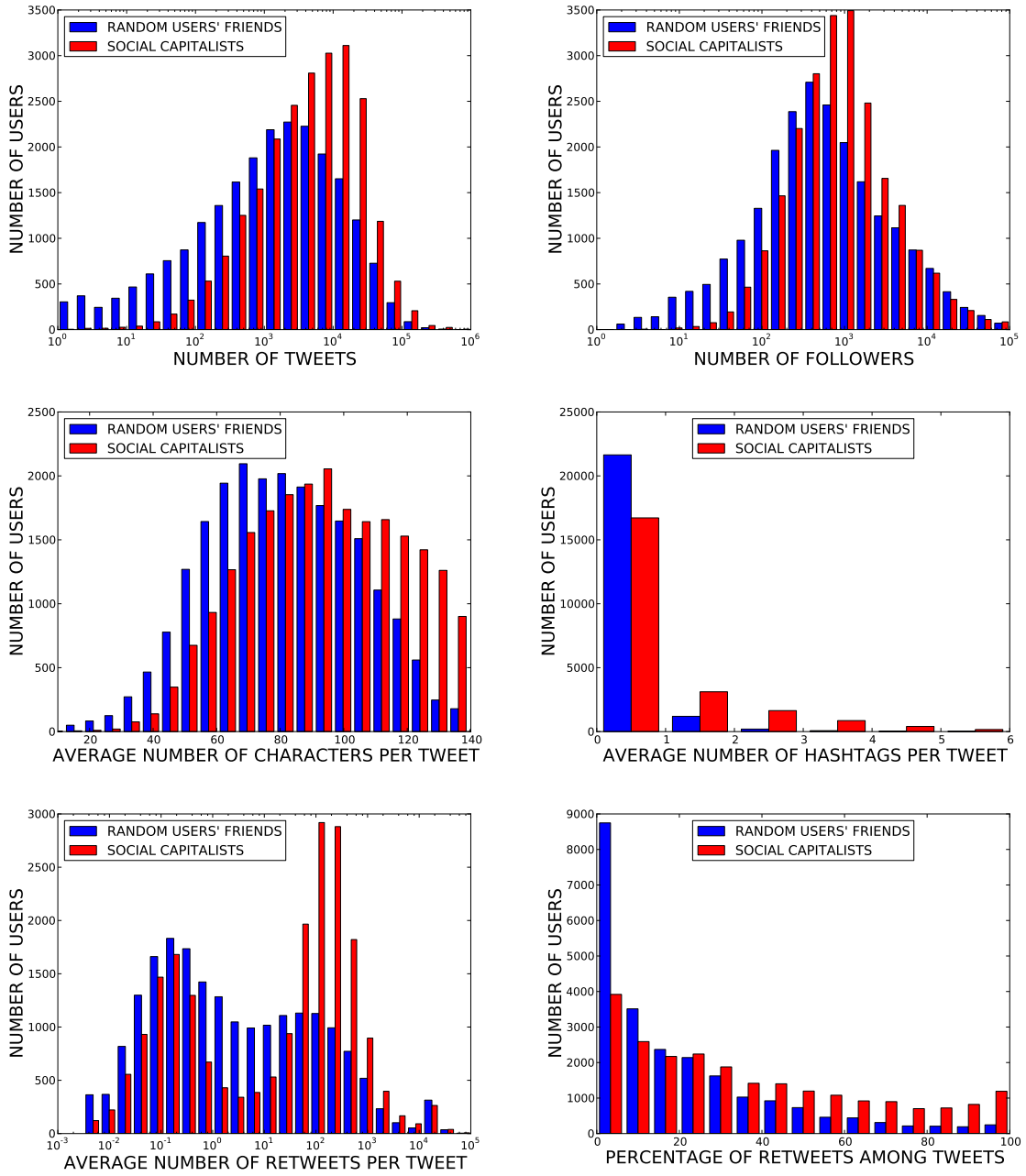

FIGURE 3.8 – Histogrammes de la distribution de 6 attributs pour les deux classes d'utilisateur : capitalistes sociaux et utilisateurs réguliers.

On observe visuellement sur la Figure 3.8 que les attributs choisis semblent pertinents pour discriminer les capitalistes sociaux des utilisateurs *réguliers*. En effet, on peut voir que la distribution de ces attributs est différente pour ces deux types d'utilisateur. Il est par exemple intéressant de constater que les capitalistes sociaux sont plus retweetés que les utilisateurs réguliers, le nombre de retweets étant au cœur des scores **Klout** et **Kred**.

Par ailleurs, afin de mieux visualiser nos données, nous réalisons une *analyse en composantes principales (ACP)* [94]. En effet, nos données sont pour le moment représentées en dimension $p = 20$, ce qui est trop grand pour être visualisé. L'ACP consiste à rechercher une représentation de nos données dans un espace de dimension inférieure à p avec un minimum de perte d'informations. Nous recherchons en fait des combinaisons linéaires de nos variables initiales telles que ces combinaisons nous permettent de *bien* représenter nos données en plus faible dimension : nous choisissons les combinaisons linéaires de nos variables initiales qui ont la variance la plus élevée. Pour construire cette représentation, nous calculons d'abord la matrice des variances-covariances à partir de notre matrice initiale $M_{n,p}$ et nous en cherchons les valeurs propres et vecteurs propres associés. Les vecteurs propres sont les combinaisons linéaires et les valeurs propres la variance des nouvelles variables représentées par ces combinaisons linéaires. Ici, nous choisissons de présenter les résultats de l'*ACP* en nous intéressant aux trois premiers vecteurs propres. Nous projetons donc nos données sur notre nouvel espace de représentation des données que l'on appelle *plan factoriel*. Les trois premières valeurs propres expliquent 46% de la variance des variables initiales. Avec ces visualisations, nous constatons une séparation entre les utilisateurs réguliers en rouge (exemples négatifs) et les capitalistes sociaux en bleu (exemples positifs). Il semble donc que nos variables soient pertinentes pour discriminer ces deux types d'utilisateurs.

Nous nous appuyons donc sur ce jeu de données pour mettre en place une méthode d'*apprentissage supervisé* dans le but d'exploiter ces données efficacement.

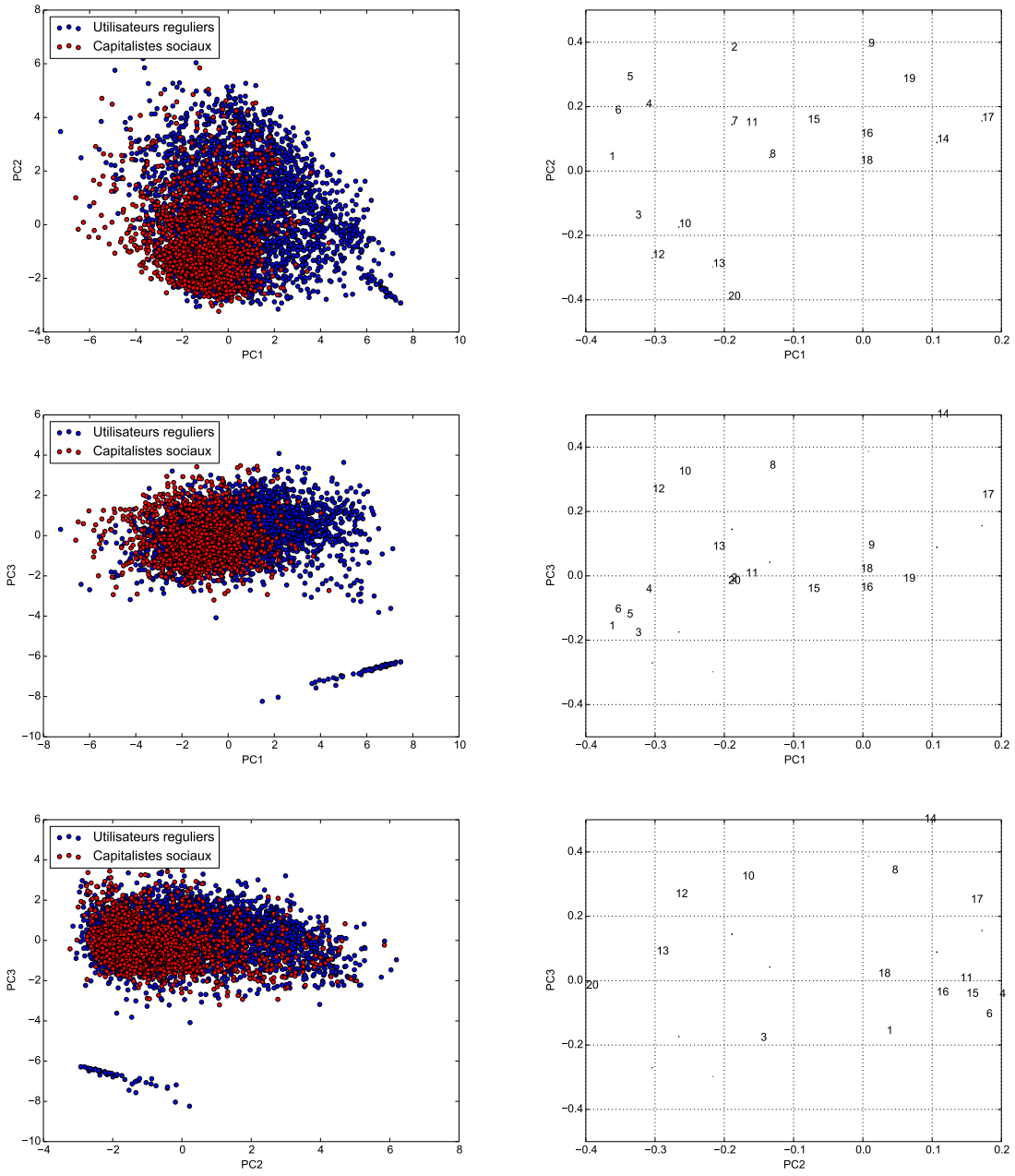

FIGURE 3.9 – Projection d'un sous-ensemble du jeu de données sur les trois premiers plans factoriels. Interprétation de chacune des composantes principales à droite. Les chiffres font référence à la Table 3.3.

3.4 Vers une nouvelle mesure de l'influence

3.4.1 Classification supervisée : détecter les capitalistes sociaux

Dans le Chapitre 1, nous avons discuté une méthode de détection basée uniquement sur des mesures de similarité des *voisinages* des capitalistes sociaux. Cette méthode était particulièrement utile pour une détection à l'échelle du réseau entier de **Twitter**. Par ailleurs, nous n'avions pas d'informations supplémentaires sur les utilisateurs. Pour ces travaux, nous souhaitons établir une méthode plus efficace à partir d'une plus grande variété de données. Nous utilisons donc le jeu de données décrit dans la Section 3.3 pour construire un modèle capable de discriminer les capitalistes sociaux des utilisateurs réguliers. Ce jeu de données contient 77.102 utilisateurs (22.845 capitalistes sociaux et 54.257 utilisateurs réguliers) décrits par les attributs de la Table 3.3.

Pour établir un modèle discriminant les capitalistes sociaux des utilisateurs réguliers, nous utilisons des méthodes de classification supervisée binaire. L'objectif est d'apprendre un classifieur à partir des p attributs de nos exemples étiquetés positifs et négatifs. Une fois ce classifieur appris, il est possible de l'utiliser sur de nouvelles observations non étiquetées pour prédire leur étiquette. Dans ces travaux, nous utilisons des algorithmes classiques destinés à l'apprentissage de classifieurs :

— k plus proches voisins (KNN)
— Machines à vecteurs de support (SVM)
— Forêts aléatoires (RF)
— Régression logistique (LR)
— Naïve Bayes Gaussien (GNB)

Nous avons implémenté ces différents algorithmes en utilisant la librairie Python **Scikit-learn** [94]. Nous décrivons maintenant ces algorithmes de classification supervisée.

k plus proches voisins. L'algorithme procède simplement : pour une entrée donnée que l'on souhaite étiqueter, l'algorithme cherche les k voisins les plus proches dans l'espace décrit par les attributs. La classe la plus représentée parmi ces k voisins est attribuée à l'entrée. Nous avons effectué nos tests en utilisant la distance de Minkowski avec p allant de à 1 à 5 et k variant de 4 à 6. La distance de Minkowski d'ordre p entre deux vecteurs $A = (a_1, a_2, \ldots, a_n)$ et $B = (b_1, b_2, \ldots, b_n)$ de dimension n est définie par $\left(\sum_{i=1}^{n} |a_i - b_i|^p\right)^{1/p}$. Les résultats présentés sont ceux obtenus avec la distance euclidienne ($p = 2$) mais ils sont représentatifs de ce que nous observons avec les autres paramètres.

Machines à vecteurs de support. Les machines à vecteurs de support linéaires cherchent un hyperplan séparateur qui maximise la distance entre celui-ci et les exemples du jeu de données les plus proches appelés *vecteurs supports*. Les noyaux sont utilisés lorsque les données ne sont pas séparables linéairement. Il s'agit d'appliquer une fonction non-linéaire aux observations afin de les représenter dans un nouvel espace dans lequel on peut rechercher une séparation linéaire. Nous avons effectué nos expérimentations avec les SVM linéaires et les noyaux RBF, sigmoïde et polynomial. Les résultats présentés sont ceux obtenus avec les SVM linéaires (*SVM-LIN*) et à noyau RBF (*SVM-RBF*) qui produisent les meilleurs résultats.

Forêts aléatoires. Le principe des *forêts aléatoires* mélange les deux concepts d'arbre de décision et de méthode ensembliste. Un ensemble d'arbres de décisions est appris sur des sous-ensembles d'attributs aléatoires. Enfin, la décision est prise par vote des arbres appris.

Régression logistique. Comme le mot *régression* l'indique, il s'agit d'établir la dépendance entre la classe des exemples et leurs attributs. Cette régression est dite logistique puisqu'elle est basée sur l'expression *logit* qui peut s'écrire comme suit :

$$p(1|X) = \frac{e^{b_0+b_1X_1+\ldots+b_pX_p}}{1+e^{b_0+b_1X_1+\ldots+b_pX_p}}$$

avec X l'exemple à prédire et $\{X_1,\cdots,X_p\}$ les p attributs. Il s'agit donc d'apprendre les coefficients $b_i, i \in \{1,\cdots,p\}$.

Naïve Bayes Gaussien. Ce modèle est un modèle probabiliste qui suppose une distribution gaussienne des variables. Le modèle se base sur cette supposition pour calculer la probabilité qu'un exemple appartienne à une classe, et ainsi de décider en choisissant la classe avec la probabilité sachant les attributs de l'observation maximale.

Afin d'améliorer les résultats de la régression logistique [8], nous effectuons deux modifications aux données lors de l'utilisation de cet algorithme :

— Nous ajoutons des attributs supplémentaires qui sont en fait le produit de chaque paire d'attributs ;
— La plupart des attributs étant plus pertinents à l'échelle logarithmique, nous appliquons donc le logarithme à chacun des attributs avant normalisation.

Pour commencer, nous séparons notre jeu de données en un ensemble d'apprentissage et un ensemble de test équilibrés. L'ensemble d'apprentissage contient ainsi 70% des observations et l'ensemble de test les 30% restants. Nous utilisons ces deux ensembles pour

tester nos méthodes de classification supervisée. Nous évaluons la qualité des différents modèles appris avec plusieurs indicateurs décrits ci-après.

Définition 3.1 (Vrai/faux positif/négatif)**.** *L'ensemble des vrais positifs (resp. négatifs), noté TP (resp. TN), contient les observations classifiées comme positives (resp. négatives) par le classifieur qui le sont réellement. De manière similaire, FP (resp. FN) dénote l'ensemble des faux positifs (resp. négatifs).*

Définition 3.2 (Sensibilité)**.** *La sensibilité calcule la proportion des observations positives classifiées comme telles. Ainsi :*

$$Se = \frac{|TP|}{p}$$

avec $p = |TP| + |FN|$.

Notons que la sensibilité est parfois dénommée *rappel.*

Définition 3.3 (Spécificité)**.** *La spécificité calcule la proportion des observations négatives classifiées comme telles. On a donc*

$$Sp = \frac{|TN|}{n}$$

avec $n = |TN| + |FP|$.

Définition 3.4 (Accuracy)**.** *L'accuracy calcule la proportion d'exemples correctement classifiés selon la formule suivante :*

$$Acc = \frac{|TP| + |TN|}{p + n}$$

où $p = |TP| + |FN|$ et $n = |TN| + |FP|$.

Définition 3.5 (F-score)**.** *Le F-score pénalise un classifieur qui n'optimiserait que la sensibilité ou que la spécificité.*

$$F_s = \frac{2 \cdot Se \cdot Sp}{Se + Sp}$$

Nous observons Table 3.4 que les meilleurs résultats sont obtenus avec les algorithmes de régression logistique et de forêts aléatoires. La précision obtenue avec les classifieurs qu'ils apprennent est supérieure de près de 10% à celle du SVM linéaire et d'un peu moins de 10% à celles des KNN. Le F-Score est supérieur de plus de 30% à celui du SVM et à près de 20% de ceux des KNN. Ces résultats sont similaires à ceux obtenus par McCord et Chuah [81] pour la détection de spammeurs.

Algorithme	Acc	Se	Sp	F_s
RL	90,10	**93,41**	88,70	**91,00**
RF	**92,74**	84,48	**96,21**	89,96
SVM-LIN	79,66	42,22	95,39	58,53
SVM-RBF	81.47	48.23	95.44	64.08
KNN (k=4)	83,53	58,35	94,11	72,03
KNN (k=5)	83,85	67,61	90,67	77,46
KNN (k=6)	83,88	60,42	93,74	73,48
GNB	76.51	82.37	74.04	77.98

TABLE 3.4 – Résultats en pourcentage des différents classifieurs sur l'ensemble de test. Les meilleurs résultats sont en gras.

La régression logistique est spécifiquement conçue pour retourner la probabilité d'une observation d'être positive (Figure 3.10).

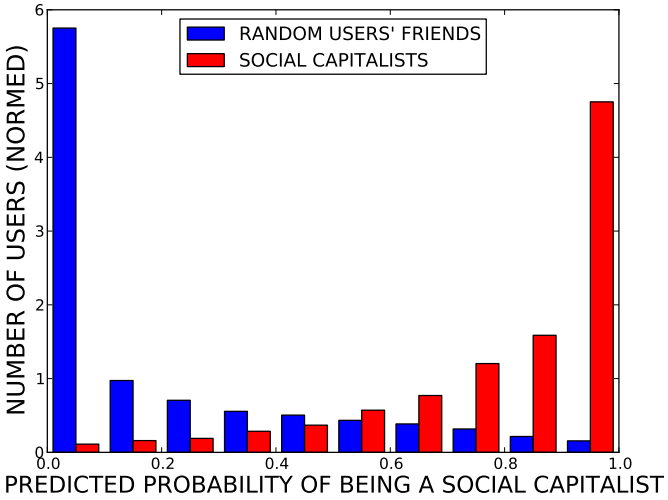

FIGURE 3.10 – Histogramme des probabilités prédites par la régression logistique. En bleu, les probabilités que les utilisateurs soient *réguliers*, en rouge, qu'ils soient des capitalistes sociaux.

Par ailleurs, une fois la régression terminée, il suffit de stocker les coefficients appris pour chaque attribut afin d'appliquer le modèle de prédiction à de nouveaux exemples. Ce sont des caractéristiques particulièrement intéressantes pour la mise en place d'une application en ligne (Section 3.5). De plus, le modèle est facilement instanciable et interprétable, notamment grâce aux *rapport des chances* ou *odds ratio*. Afin de mieux caractériser la

contribution de chaque attribut, nous utilisons deux méthodes. Tout d'abord, nous utilisons les *odds ratio* générés par la régression logistique. L'*odd ratio* décrit la force et le signe de la corrélation entre un attribut et une classe. Il est compris entre 0 et $+\infty$ et s'interprète comme suit :

— Lorsqu'il vaut 1, l'attribut n'aide pas à discriminer les deux classes ;
— S'il est proche de $+\infty$ alors on a une forte corrélation positive entre la variable et la classe positive ;
— S'il est proche de 0 alors on a une forte corrélation négative entre la variable et la classe positive.

Nous remarquons sur la Table 3.5 que les odds ratio les plus élevés sont ceux des attributs 6, 10 et 12 qui représentent respectivement le nombre d'abonnés qui sont également des abonnements, le nombre de mentions par tweets et le nombre de retweets des retweets.

Attribut	1	2	3	4	5	6	7	8	9	10
odd ratio	4,37	7,00	0,54	2,32	0,31	492,86	9,10	7,97	2,65	34,46
Attribut	11	12	13	14	15	16	17	18	19	20
odd ratio	0,52	65,67	1,61	0,03	6,98	4,07	0,76	1,38	9,80	6,93

TABLE 3.5 – Odd ratio de chacun des attributs à l'issue de la régression logistique. Le numéro d'attribut correspond à celui de la Table 3.3.

Le premier attribut fait écho à l'indice de chevauchement utilisé dans le Chapitre 1. Quant aux deux autres attributs, il n'est pas étonnant de les retrouver ici étant données les pratiques des capitalistes sociaux qui incitent aux retweets et aux mentions (Figure 3.5). Par ailleurs, parmi les *odds ratio* proches de 0, nous observons les attributs 3, 5, 11, 14 qui représentent respectivement le nombre de tweets en favoris, le nombre d'abonnés, le nombre de retweets moyen des tweets et le nombre de secondes moyen entre chaque tweet. Les capitalistes sociaux semblent donc tweeter plus souvent, être moins retweetés lorsqu'il s'agit de leurs tweets propres et moins utiliser le système de favoris. Par contre, il est surprenant de retrouver le nombre d'abonnés ici puisque l'objectif des capitalistes sociaux est d'obtenir plus d'abonnés. Cela confirme l'observation faite dans la Section 1.2 où nous remarquions que seulement un peu plus 50% des capitalistes sociaux collectés avaient un nombre d'abonnements supérieur à 500.

Nous évaluons également la pertinence des groupes d'attributs indépendamment et en exprimons les résultats dans la Table 3.6.
Le nombre de secondes entre chaque tweet, le nombre moyen de hashtags par tweet et le nombre moyen de mentions par tweets donnent les meilleurs résultats. Ces observations

Attribut	Acc	Se	Sp	F_s
1	57,2%	74,2%	50,1%	59,8%
2	50,5%	86,0%	35,4%	50,2%
3	57,9%	65,4%	54,7%	59,6%
4	60,2%	42,4%	67,8%	52,2%
5	52,4%	75,8%	42,5%	54,4%
6	65,2%	46,5%	73,2%	56,9%
7	58,9%	62,7%	57,3%	59,9%
8	79,4%	73,1%	82,1%	**77,3%**
9	53,1%	73,1%	44,6%	55,4%
10	65,3%	71,6%	62,7%	**66,8%**
11	45,6%	76,3%	32,4%	45,5%
12	62,4%	64,4%	61,5%	63,0%
13	58,1%	83,0%	47,4%	60,3%
14	72,3%	81,1%	68,55%	**74,3%**
15	38,7%	94,8%	14,9%	25,8%
16	73,0%	23,8%	93,9%	38,0%
17	60,1%	74,2%	54,1%	62,6%
18	70,1%	0,27%	99,8%	0,54%
19	51,1%	85,7%	36,4%	51,1%
20	66,2%	58,5%	69,5%	63,5%

TABLE 3.6 – Résultats obtenus en utilisant un seul attribut pour l'apprentissage et la prédiction. Le numéro d'attribut correspond à celui de la Table 3.3.

sont cohérentes avec les *odds ratio* et notre étude des techniques de capitalisme social qui montrent que ces utilisateurs tweetent en utilisant un nombre élevé de hashtags et de mentions (Figure 3.5).

Dans la Table 3.7, nous évaluons par ailleurs la pertinence des cinq groupes d'attributs de la Table 3.3. Le groupe le plus pertinent est celui qui concerne l'activité des utilisateurs. En revanche, un seul groupe d'attributs ne suffit pas à effectuer de bonnes prédictions.

Groupe	**Acc**	**Se**	**Spe**	$\mathbf{F}_s$
Activité	71,7%	82,8%	67,0%	74,1%
Topologie locale	64,9%	73,4%	61,2%	66,8%
Contenu des tweets	69,4%	63,9%	71,7%	67,5%
Caractériques des tweets	65,8%	68,0%	64,8%	66,4%
Sources	67,3%	76,1%	63,6%	69,3%

TABLE 3.7 – Résultats obtenus en *utilisant* un seul groupe d'attributs.

Nous testons donc la robustesse de notre classifieur de façon plus réaliste. Nous supposons que les capitalistes sociaux modifient un seul aspect de leur fonctionnement. Nous pouvons également imaginer la situation où **Twitter** ne nous laisserait plus accéder à certaines données. Dans ces deux cas, afin de connaître les résultats de notre classifieur, nous en démontrons les performances dans la Table 3.8.

Groupe	**Acc**	**Se**	**Spe**	$\mathbf{F}_s$
Activité	88,92%	93,29%	87,05%	90,06%
Topologie locale	88,07%	92,19%	86,31%	89,16%
Contenu des tweets	88,65%	92,03%	87,21%	89,55%
Caractériques des tweets	85,07%	88,70%	83,52%	86,03%
Sources	88,73%	92,17%	87,26%	89,65%

TABLE 3.8 – Résultats obtenus en *supprimant* un seul groupe d'attributs.

Nous observons que les performances sont très bonnes, les *F-score* étant tous supérieurs à 86% quand le *F-Score* du classifieur utilisant tous les attributs est de 91%. Notre classifieur est donc particulièrement robuste à la perte de certaines informations ou à un changement de comportement local des capitalistes sociaux.

Nous évaluons également la qualité de notre classifieur avec la courbe *ROC* de la Figure 3.12 qui montre la proportion de vrais positifs en fonction de la proportion de faux postifs. Cela permet d'évaluer la performance du classifieur si l'on fait évaluer le seuil à partir duquel on prédit un utilisateur comme positif. L'aire sous la courbe *ROC* -appelée AUC- est de $0,96$, l'idéal étant 1. L'idéal représente le cas où la proportion de vrais positifs est égale à 1.

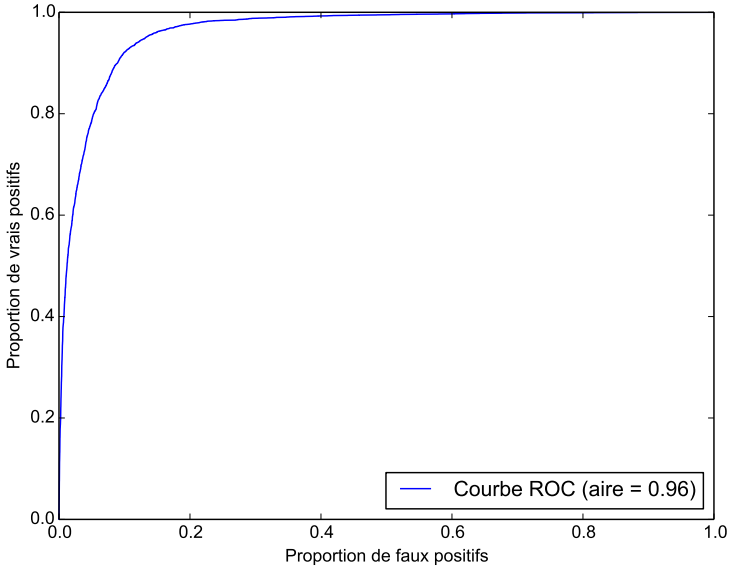

FIGURE 3.11 – Courbe ROC, proportion des faux positifs en fonction de la proportion de vrais positifs.

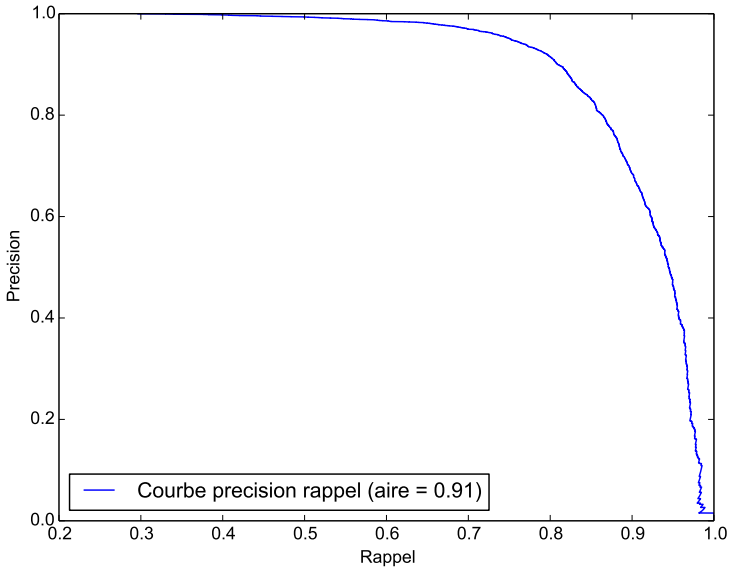

FIGURE 3.12 – Précision en fonction du rappel.

Par ailleurs, la précision est donnée par le ratio $\frac{TP}{TP+FP}$. Ainsi, la courbe de la Figure 3.12 qui présente la précision en fonction du rappel démontre la qualité de notre classifieur. Il est en effet possible de détecter un grand nombre de capitalistes sociaux -augmenter le rappel- sans obtenir trop de faux positifs -diminuer la précision.

Enfin, pour valider la robustesse de notre classifieur par rapport à l'ensemble d'apprentissage choisi, nous effectuons une *validation croisée*. Nous séparons notre jeu de données en 5 ensembles distincts. Nous choisissons tour à tour l'un de ces 5 ensembles comme ensemble de test et les quatre autres comme ensemble d'apprentissage. On obtient en moyenne sur ces 5 expérimentations une *accuracy* de 90% avec un écart-type inférieur à $0,01$ et une AUC moyenne de $0,96$ avec un écart-type inférieur à $0,005$.

Nous avons donc appris un modèle efficace pour discriminer les capitalistes sociaux des utilisateurs réguliers. Il s'agit maintenant d'utiliser ce modèle dans le cadre de l'influence des capitalistes sociaux.

3.4.2 Pondérer le Score Klout

La régression logistique évalue la probabilité pour un utilisateur donné d'être un capitaliste social que l'on note P_{Ksoc}. Nous pouvons donc utiliser cette probabilité P_{Ksoc} pour pondérer le score **Klout** S_{Klout} d'un utilisateur. Le nouveau score est appelé S_{DDP}. Comme mentionné précédemment, nous choisissons le score **Klout** car ce score nous semble être le plus populaire.

$$S_{DDP} = \begin{cases} S_{Klout} & \text{si } P_{Ksoc} \leq 0,5 \\ 2 \cdot (1 - P_{Ksoc}) \cdot S_{Klout} & \text{si } P_{Ksoc} > 0,5 \end{cases} \tag{3.1}$$

Ce score pénalise les utilisateurs classifiés comme capitalistes sociaux avec une probabilité non négligeable. Nous l'évaluons sur des capitalistes sociaux connus, des utilisateurs influents et des utilisateurs aléatoires de notre jeu de données. Les résultats sont décrits dans la Table 3.9.

Barack Obama et **Oprah Winfrey** ne sont -heureusement- pas considérés comme des capitalistes sociaux par notre classifieur. En effet, bien que les capitalistes sociaux passifs (avec un ratio inférieur à $0,7$) aient à un moment appliqué des techniques de capitalisme social, ils ont de puis cessé et sont généralement devenus réellement influents sur le réseau. Lorsque l'on considère des exemples positifs de notre jeu de données comme **teamukfollowbac** ou **TEEMF0LL0W**, nous observons que la probabilité retournée par notre classifieur est plus élevée. Ainsi, le score DDP reflète cette observation en diminuant l'influence de ces utilisateurs. C'est encore plus le cas lorsque l'on considère **seanmaxwell** et **followback_707**, deux utilisateurs également extraits de notre ensemble d'exemples posi-

Name	**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	5
scarina91	46	0,110	41
teamukfollowbac	80	0,838	13
TEEMF0LLOW	69	0,416	41
zuandoemkta	62	0,861	9
kosma003	53	0,747	14

TABLE 3.9 – Klout et DDP scores en fonction de P_{Ksoc}.

tifs. Le dernier partage uniquement des contenus destinés au capitalisme social, et ne peut donc pas être considéré comme influent. Contrairement au score **Klout**, le score *DDP* en tient compte.

3.5 Application web

Nous présentons dans cette Section l'application web *DDP* qui implémente notre méthode de détection de capitalistes sociaux par *classification supervisée*. Dans un premier temps, nous décrivons les fonctionnalités utilisateurs. Nous exposerons ensuite les fonctionnalités qui sont accessibles pour l'administration de l'application. Enfin, nous terminerons en discutant de l'implémentation de cette application.

3.5.1 Fonctionnalités utilisateurs

Une fois sur la page d'accueil (Figure 3.13), l'utilisateur peut en apprendre plus sur le projet en cliquant sur *How does it work ?* ou *About* en haut à droite. Mais il peut surtout se connecter à l'application via son compte Twitter en cliquant sur *Sign in with Twitter*. En se connectant à son compte Twitter via ce bouton, Twitter va créer une paire de *jetons* qui associent le compte Twitter à l'application. Cette paire de *jetons* permet ainsi à l'application d'effectuer des requêtes à l'API Twitter au nom de l'utilisateur. Par ailleurs, ce nombre de requêtes est plus élevé que celui autorisé normalement pour une application. Cette connexion est donc importante puisqu'elle nous permet de garantir à l'utilisateur une bonne utilisation de l'application : il ne sera pas limité par le nombre de requêtes.

Une fois enregistré, l'utilisateur peut entrer l'identifiant de n'importe quel compte Twitter pour tester si ce compte est un capitaliste social ou non. Cela permet à l'utilisateur de lan-

FIGURE 3.13 – Page d'accueil de l'application DDP.

cer l'application qui va récupérer tous les attributs listés Table 3.3 ainsi que des attributs supplémentaires du compte **Twitter** testé. L'application récupère ensuite les coefficients de la *régression logistique* -stockés en base de données- et les utilise pour retourner la probabilité que le compte choisi par l'utilisateur soit un capitaliste social. Parmi les attributs supplémentaires, certains sont utilisés pour compléter la description du contenu des tweets de l'utilisateur avec les nombres :

— d'URLS uniques ;
— de mentions uniques ;
— de hashtags uniques.

Ou encore les :

— similarités moyennes entre tweets ;
— polarités détectées des tweets.

Nous rappelons que ces attributs sont obtenus à partir des 200 derniers tweets postés par l'utilisateur. La similarité entre tweets est obtenue en utilisant la *similarité cosine* entre les tweets. Pour calculer la similarité entre deux tweets, les tweets sont d'abord représentés sous forme de vecteurs de mots puis le cosinus de l'angle entre les deux vecteurs est retourné. Par ailleurs, nous décrirons plus en détails l'analyse de polarité faite sur les tweets Section 3.5.3. Il s'agit simplement de calculer la probabilité qu'un tweet exprime un sentiment positif ou négatif pour en déterminer la plus élevée, avec laquelle il est étiqueté.

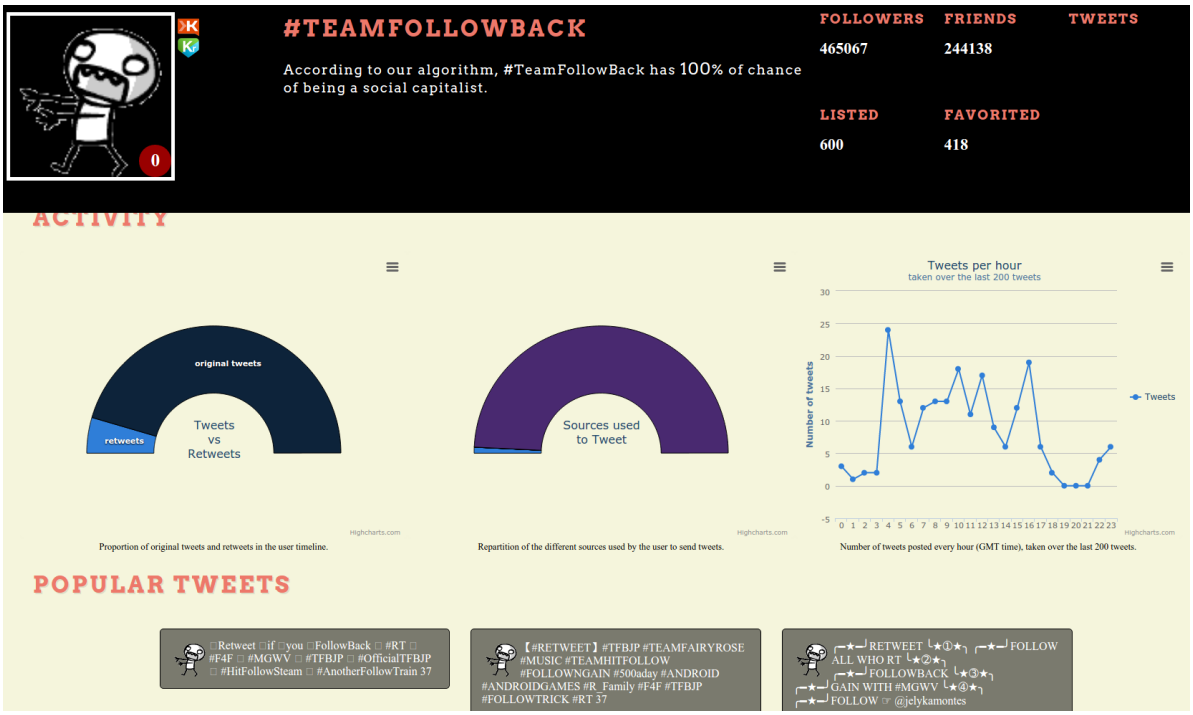

FIGURE 3.14 – Première vue des attributs de l'utilisateur testé.

Nous obtenons également d'autres attributs pour mieux décrire la topologie locale :

— l'écart-type des identifiants des 5.000 derniers abonnés ;
— l'écart-type des identifiants des 5.000 derniers abonnements ;
— l'indice de chevauchement entre les ensembles des 5.000 derniers abonnés et abonnements.

Ces attributs ne portent que sur les 5.000 derniers abonnés et abonnements qu'il est possible d'obtenir en une seule requête à l'API. De même, l'attribut 6 de la Table 3.3 est calculé uniquement à partir de l'intersection des 5.000 derniers abonnés et abonnements. Cela permet de respecter les limitations de l'API **Twitter** (Annexe C) tout en garantissant un temps de réponse raisonnable.

Enfin, la catégorie *Caractéristique des tweets* de la Table 3.3 est enrichie avec l'écart-type au nombre moyen de secondes entre tweets.

Tous ces attributs ainsi que la probabilité calculée par le classifieur basé sur les coefficients de la régression logistique sont ensuite affichés sur une page dédiée (Figures 3.14, 3.15 et 3.16). Sur la Figure 3.14, nous pouvons observer ce que l'utilisateur de l'application voit

FIGURE 3.15 – Seconde vue des attributs de l'utilisateur testé.

juste après avoir entré un identifiant **Twitter** dans notre application. En haut à gauche, nous pouvons observer la photo de profil de l'utilisateur, ses scores **Klout** et **Kred** et son score **DDP** cerclé de rouge sur la photo de profil. Plus à droite, nous pouvons voir la probabilité retournée par la régression logistique sous forme de pourçentage ainsi que le pseudonyme adopté par l'utilisateur. Enfin, tout à droite, l'application affiche les informations de la catégorie *Activité* de la Table 3.3 : les nombres d'abonnés, abonnements, tweets, favoris et listes auxquelles l'utilisateur appartient.

Sur la première vue des résultats, de gauche à droite et de haut en bas, nous pouvons également consulter :

— le pourcentage de tweets ou de retweets parmi ce qui est tweeté par l'utilisateur ;
— les sources utilisées par l'utilisateur pour tweeter ;
— les heures auxquelles sont postés les tweets de l'utilisateur ;
— les tweets populaires de l'utilisateur retournés par Twitter.

Sur la seconde vue des résultats (Figure 3.15), de gauche à droite, nous affichons sous forme de nuages de mots interactifs :

— les utilisateurs mentionnés ;
— les hashtags distincts recueillis dans les tweets ;
— les URLs distinctes postées par l'utilisateur ;
— les tweets populaires de l'utilisateur retournés par Twitter.

Dans le nuage, plus un utilisateur, un hashtag ou une URL sont de grande taille, plus ils ont été observés dans les tweets de l'utilisateur. Par ailleurs, chacun des utilisateurs, hashtags et URLs sont cliquables et mènent vers la page Twitter correspondante.

Enfin, dans la dernière vue des résultats (Figure 3.16), nous affichons le pourcentage de tweets ayant été détectés avec une *polarité* positive sur la gauche, négative sur la droite.

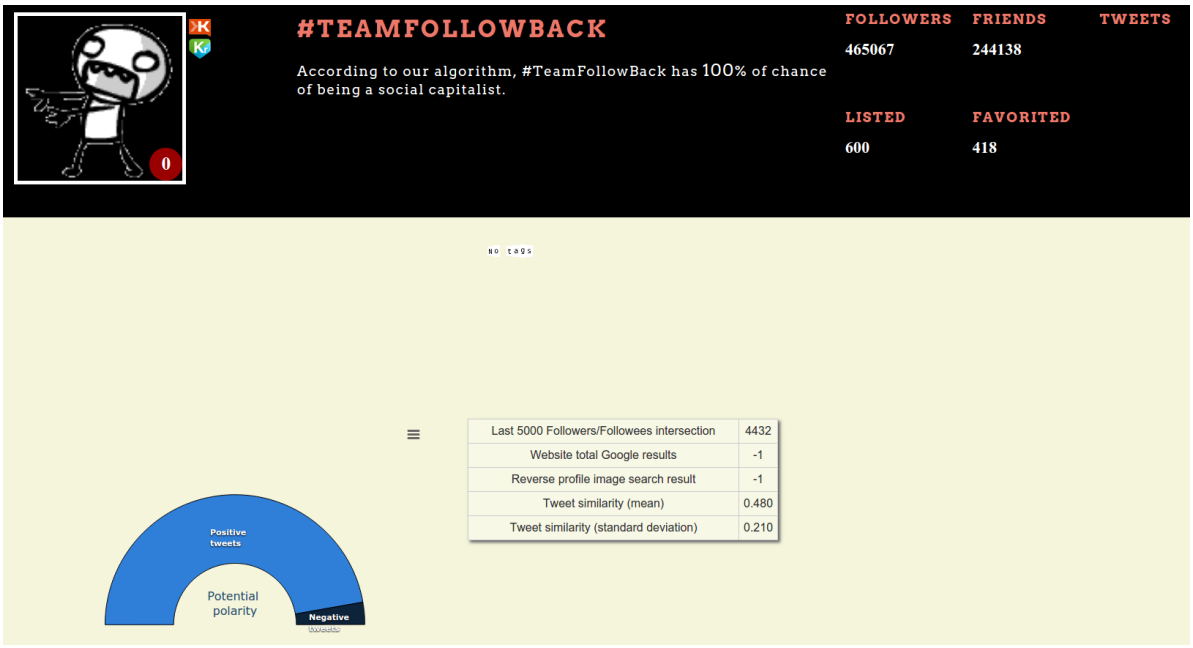

FIGURE 3.16 – Dernière vue des attributs de l'utilisateur testé.

Sur la droite, nous affichons un tableau qui contient :

— l'Indice de chevauchement entre les ensembles des 5.000 derniers abonnés et abonnements ;
— le nombre de résultats retournés par la recherche **Google** du site renseigné par l'utilisateur dans son profil **Twitter** ;
— la similarité moyenne et l'écart-type à la moyenne des 200 derniers tweets.

Le nombre de résultats retournés par la recherche inversée de l'image de profil en utilisant **Image Raider** est en cours d'implémentation. Si l'utilisateur n'a pas renseigné de site web, alors -1 est affiché pour le nombre de résultats **Google**.

3.5.2 Fonctionnalités administrateurs

Nous présentons maintenant les fonctionnalités accessibles uniquement aux administrateurs.

Compte twitters

Refresh Scores

username	Screen Name	friends	Followers	Score	klout	kred	Capitalist
SondyNews	#TeamFollowback	265	70744	0.4669	50	79	CAP
z_o_m_b_ii_e	#TeamFollowBack	244090	465098	0.9996	null	null	CAP
Leboss107	#TeamFollowBack	41461	134402	0.8062	44	79	CAP
TeamFollowWacky	#TeamFollowBack	6218	478658	0.2142	64.684994457047	100	CAP
BEZESIRO	#TeamfollowBack #FF	9206	83360	0.0556	46.614730862678	79	CAP
ManmiPaLa	#Teamfollowback	35	17378	0.5542	null	null	CAP
AliceJard	Alice Jard	1825	1244	0.3523	null	null	unknown
CookieGamerMel	Amélie Daviau	51	17	0.0429	null	null	NOT-CAP
barackobama	Barack Obama	643356	58325575	0.0001	null	null	NOT-CAP

FIGURE 3.17 – Étiqueter les utilisateurs coté administrateur.

Sur la Figure 3.17, nous pouvons observer un tableau qui liste les comptes **Twitter** qui ont été testés par des utilisateurs de l'application. Tous ces comptes et leurs attributs sont stockés dans une base de données détaillée Section 3.5.3. L'interface nous permet d'étiqueter ces utilisateurs pour confirmer ou infirmer les résultats produits par le classifieur. Les étiquettes sont visibles sur la droite : *cap*, *not-cap* et *unknown* par défaut. Nous affichons également les scores **Klout**, **Kred** et **DDP**. Les utilisateurs étiquetés manuellement sont ensuite utilisés si une régression logistique est relancée : ils fournissent des exemples supplémentaires. Par ailleurs, ils fournissent des données utilisables pour étudier l'évolution du phénomène. La régression logistique peut être lancée en cliquant sur le label de la Figure 3.18. Les coefficients et les résultats sont stockés en base de données et affichés dans le tableau côté administrateur. La ligne en vert est celle choisie par l'administrateur : les coefficients associés aux résultats choisis sont donc utilisés par la suite dans l'application pour faire les prédictions. Pour le moment, seuls les attributs décrits Table 3.3 sont utilisés pour la régression logistique. Par la suite, une fois que suffisamment de données auront été collectées, les attributs supplémentaires pourront être utilisés. Afin d'accélérer la collecte des données, un outil en mode *batch* a été créé (Figure 3.19). À gauche, il est possible

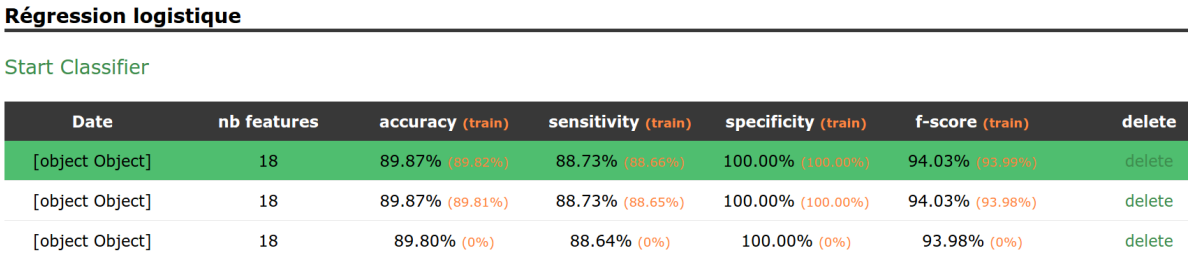

FIGURE 3.18 – La régression logistique coté administrateur.

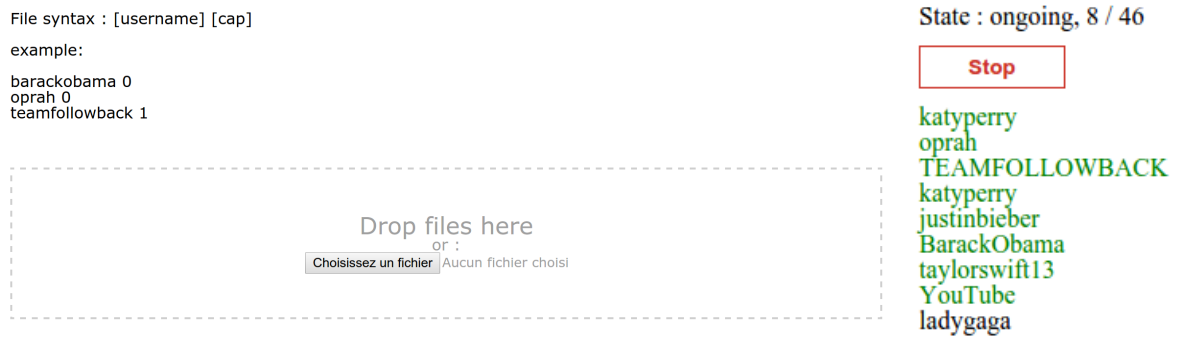

FIGURE 3.19 – Récolte des données en mode *batch*.

d'insérer un fichier d'utilisateurs étiquetés pour en récolter les attributs. Une fois que les données associées à un utilisateur ont été recueillies, celles-ci sont stockées en base de données et l'utilisateur est affiché en vert (à droite sur la Figure 3.19). Ceci permet de suivre l'avancement dans la récolte des données.

3.5.3 Implémentation

Application web. La couche présentation de l'application web est développée en *HTML/CSS* avec du *Javascript* pour ajouter de la réactivité. Pour dialoguer avec le serveur et produire des résultats de façon dynamique, nous utilisons *PHP*. Les requêtes asynchrones avec *Ajax* nous permettent notamment de requêter l'API **Twitter** -avec la librairie *Oauth*- pour obtenir les informations sur un utilisateur sans figer l'interface. Celles-ci sont effectuées en arrière-plan.

Données. Nous avons choisi pour le stockage des données d'utiliser *MongoDB*. La base de données utilise en effet *JSON* comme format, pour le stockage et le requêtage. L'API **Twitter** retournant les informations au format *JSON*, il n'est donc pas nécessaire d'effectuer de conversion des données avant de les stocker. Par ailleurs, nous proposons aux uti-

lisateurs de l'application de télécharger au format *JSON* les données obtenues sur un utilisateur. Ce format est lisible, facile à analyser et il est de plus en plus utilisé.

Langage naturel. Pour analyser la *polarité* des tweets, c'est à dire les étiqueter positivement ou négativement selon le sentiment qu'ils expriment, nous utilisons la librairie **NLP Tools**. Celle-ci fournit en effet des outils *clé-en-main* pour supprimer les *mots vides*, segmenter les tweets et apprendre un classifieur capable d'analyser la polarité des tweets. Pour effectuer la détection de polarité, nous entraînons un classifieur *Naïve Bayes* hors ligne avec **NLP Tools**. Le jeu de données utilisé est issu de Go et al. [43] et contient des tweets automatiquement étiquetés grâce aux émoticones qu'ils contiennent. À partir de 500.000 tweets de ce jeu de données, nous apprenons un classifieur qui obtient un *F-Score* (Définition 3.5) de 76% et une *Accuracy* de 77% (Définition 3.4) sur le reste du jeu de données étiqueté avec les émoticones (1.500.000 tweets environ). Par ailleurs, sur un jeu de données étiqueté manuellement, nous obtenons un *F-score* de 75% et une *Accuracy* de 76%. Ces résultats sont légèrement inférieurs à ceux obtenus par Go et al. [43]. Nous sommes actuellement incapables de détecter les tweets neutres, nous étiquetons chaque tweet avec une polarité positive ou négative. Nous sommes conscients que ceci constitue une limite importante. Néanmoins, nous souhaitons savoir s'il existe une *tendance* des capitalistes sociaux à tweeter des contenus plutôt positifs ou négatifs. Nos pourcentages ne constituent ainsi pas des chiffres utilisables tels quels, mais ils peuvent à large-échelle permettre d'évaluer cette *tendance*.

L'application développée permet ainsi de continuer à enrichir nos connaissances à propos du capitalisme social. De nouveaux attributs sont collectés, certains plus complexes comme la polarité de leurs tweets. De plus, de nouveaux utilisateurs viendront enrichir la base de données qui a vocation à pouvoir être exportée et librement utilisée.

CHAPITRE

4 Sur l'extraction de sous-graphes denses

Le Chapitre 1 nous montre qu'il existe un attachement préférentiel entre les capitalistes sociaux. Leur coefficient de clustering est ainsi plus élevé que la moyenne et leurs voisins sont majoritairement des capitalistes sociaux. Partant de ce constat, nous nous demandons si les capitalistes sociaux sont regroupés en *communautés*. Nous appliquons ainsi des méthodes de détection de communautés *ego-centrées* basées sur des *mesures de proximité* ou des *fonctions de qualité locales*. De plus, nous appliquons également l'algorithme de *Louvain orienté*. Nous constatons que ces méthodes ne regroupent pas les capitalistes sociaux en communautés. Ceci nous conduit à reconsidérer notre approche : les capitalistes sociaux ne forment peut-être pas des communautés au sens strict mais des *sous-graphes denses* (Définitions 1 et 2). Nous introduisons tout d'abord le problème d'extraction de sous-graphes denses et discutons de la complexité des algorithmes existants. Nous décrivons ensuite la méthode des cœurs de communauté et présentons quelques résultats préliminaires quant à son utilisation pour l'extraction de sous-graphes denses.

Definition 1 (Sous-graphe)**.** *Soient* $G = (V, E)$ *un graphe et* $V' \subseteq V$. *Le* sous-graphe induit *par* V', *noté* $G[V']$, *est défini par* (V', E') *où* $E' = \left\{(u, v) \in E : u \in V' \wedge v \in V'\right\}$.

Definition 2 (Densité)**.** *La densité d'un graphe non-orienté* $G = (V, E)$ *est donnée par la formule suivante :*

$$den(G) = \frac{2 \cdot |E|}{|V| \cdot (|V| - 1)}$$

4.1 Des communautés de capitalistes sociaux ?

L'objectif est d'essayer de montrer que les intuitions qui suggèrent un *attachement préférentiel* des capitalistes sociaux aux autres capitalistes sociaux (Chapitre 1) conduisent à la création de *communautés* de capitalistes sociaux au sein du réseau. Il est donc nécessaire d'utiliser des méthodes de détection de communautés capables de passer à l'échelle du réseau **Twitter** de Cha et al. [20].

Il existe des méthodes de détection de communautés dites *égo-centrées* ou *locales*. Il s'agit de détecter la communauté d'un nœud donné du réseau. Ces méthodes peuvent permettre :

— de passer outre le manque de connaissance sur une partie du réseau ;

— d'obtenir une interprétation plus directe concernant la communauté d'un nœud ;

— d'éviter des temps de calcul trop élevés en se concentrant sur un nœud.

La propriété qui nous intéresse ici est la dernière. Nous souhaitons utiliser ces méthodes locales pour tenter de détecter des communautés de capitalistes sociaux.
Nous présentons dans un premier temps les approches testées à base de mesures de proximité. Nous nous intéressons ensuite à des méthodes qui optimisent des fonctions de qualités locales.

Mesures de proximité. L'utilisation de mesures de *proximité* pour le calcul de la communauté d'un nœud peut s'exprimer ainsi : les nœuds les plus *proches* du nœud en question forment sa *communauté*. Il s'agit donc de calculer la proximité d'un nœud donné à tous les autres nœuds connus du graphe. Cette intuition exprimée simplement cache cependant plusieurs difficultés :

1. Comment calculer la proximité entre deux nœuds ? En d'autres termes, quelle mesure employer ?
2. À partir de quelle distance un nœud est-il considéré trop loin pour faire partie de la communauté du nœud choisi ?
3. Que faire si un nœud appartient à plusieurs communautés ?

Il est possible de répondre partiellement à ces questions comme suit :

1. Utilisation de deux mesures complètement différentes : l'une avec paramètre, l'une sans ; l'une basée sur de la dynamique d'opinions, l'autre sur le nombre de chemins sans cycle de longueur 2 entre les nœuds ;

2. La première méthode répond au problème de façon heuristique quand la seconde propose une méthode d'apprentissage supervisé ;
3. Utilisation des méthodes multi-ego-centrées : nous choisissons plusieurs nœuds supposés dans la même communauté et calculons leur proximité à tous les nœuds du graphe puis calculons un consensus entre les scores de proximité obtenus pour chacun des nœuds choisis.

L'*opinion propagée* [26] utilise la dynamique d'opinion pour calculer la proximité des nœuds du graphe à un nœud choisi v. La méthode fonctionne comme suit :

1. On initialise la valeur de l'opinion de v à 1 et celle des autres nœuds à 0 ;
2. Chaque nœud du réseau calcule la nouvelle valeur de son opinion : c'est la moyenne de celle de ses voisins ;
3. La valeur de l'opinion de v se voit de nouveau affecter 1 avant de renouveler l'étape 2.

Intuitivement, on comprend que plus le score d'un nœud est élevé, plus sa proximité avec le nœud est forte. La méthode employée par Danisch et al. [25] pour exclure les nœuds considérés trop *loins* consiste à observer les phases *plateau-décroissance* dans la distribution du score des nœuds (Figure 4.1).

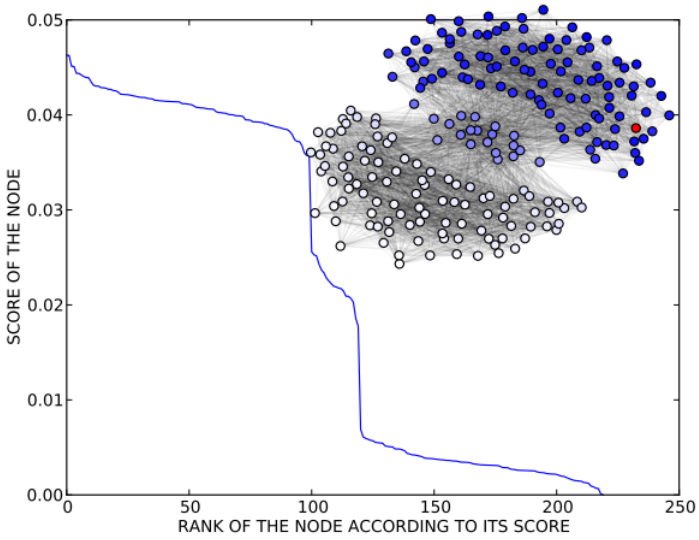

FIGURE 4.1 – Figure tirée de Danisch et al. [25] qui illustre les scores obtenus pour les nœuds de deux graphes aléatoires d'Erdos-Rényi chevauchants. Le score des nœuds est en ordonnée, leur rang en abscisse.

On observe une forte décroissance dans le score des nœuds autour du nœud de rang 100. On considère ainsi que les nœuds avant cette décroissance font partie de la même communauté. Dans le cas *multi-ego-centré*, on itère l'algorithme présenté ci-dessus avec k nœuds initialement choisis. On dit alors que la proximité d'un nœud du graphe à ces k nœuds est la proximité minimale entre ce nœud et chacun des k nœuds choisis.

La mesure *Katz+* présentée par Danisch et al. [27] propose de modifier l'indice de Katz qui calcule la proximité entre deux nœuds u et v comme dans la Définition 4.1.

Définition 4.1 (Indice de Katz)**.** *L'*indice de Katz *est* $Katz(u,v) = \sum_{l=0}^{\infty} \beta^l P_l(u,v)$ *où* P_l *représente le nombre de* chemins *de longueur* l *entre* u *et* v*, et* β *est un facteur d'atténuation compris entre* 0 *et* 1.

Plus l est grand, plus $P_l(v_i, v_j)$ l'est aussi et il s'agit de donner moins d'importance aux chemins de longueur élevée. Danisch et al. [27] proposent de remplacer le nombre de chemins P_l par le nombre de chemins *sans cycle de longueur* 2, plus rapides à calculer. Par ailleurs, les auteurs ajoutent un paramètre pour pénaliser les *hubs*, nœuds de degré élevé qui se retrouvent naturellement dans un plus grand nombre de chemins. Les auteurs proposent par ailleurs une méthode pour estimer les meilleurs paramètres en utilisant de la classification supervisée. Il s'agit de disposer d'exemples de nœuds que l'on sait dans la même communauté pour apprendre à compléter cette communauté de façon optimale.

Fonction de qualité locale. Nous nous sommes intéressés à des fonctions de qualité basées sur le découpage des nœuds du réseau comme selon la Figure 4.2.

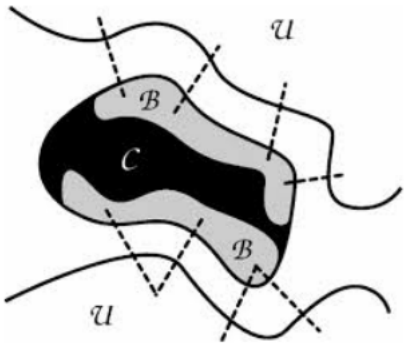

FIGURE 4.2 – Figure tirée de Clauset [24] qui illustre les trois ensembles considérés par certains algorithmes locaux : $\mathscr{B}$ la bordure, C le centre de la communauté, et $\mathscr{U}$ l'extérieur.

L'ensemble C est appelé *centre* de communauté. Il est constitué des nœuds ayant tous leurs voisins dans la communauté $D = \mathscr{B} \cup C$. L'ensemble $\mathscr{B}$ est appelé *bordure* de la

communauté, il est constitué des nœuds ayant au moins un voisin à l'extérieur de la communauté. L'ensemble $\mathscr{U}$ à l'extérieur de la communauté est constitué des nœuds qui ne sont pas dans la communauté et qui ont au moins un voisin dans la communauté.

Clauset [24] définit à partir de cette partition des nœuds du graphe ce qu'il appelle la *modularité locale*.

Définition 4.2 (Modularité locale R)**.** *La* modularité locale *est la valeur* $R = \frac{I}{T}$*, où* I *représente le nombres de liens entre les ensembles* D *et* $\mathscr{B}$*, et* T *vaut la somme des degrés des nœuds de* $\mathscr{B}$*, i.e.* $\sum_{v \in \mathscr{B}} d(v)$.

L'objectif est d'obtenir un plus grand nombre de liens au sein de la communauté que de la communauté vers le reste du graphe.

Chen et al. [21] proposent eux la fonction locale suivante : $L = \frac{L_{in}}{L_{ex}}$ avec L_{in} le degré interne moyen des nœuds de D et L_{ex} le degré externe moyen des nœuds de $\mathscr{B}$.

Méthode globale. Nous appliquons également l'algorithme de détection de communautés du Louvain orienté (Chapitre 2). En effet, celui-ci est appliquable sur le réseau complet.

Résultats. Aucune des méthodes locales n'a permis d'établir que les capitalistes sociaux forment des communautés. Les capitalistes sociaux détectés ayant un degré très élevé, les fonctions de qualité locale ont donc tendance à ne pas les insérer à l'ensemble. Par ailleurs, les capitalistes sociaux sont plus densément connectés entre eux que les autres nœuds de degré similaire sur Twitter. En revanche, la densité d'une telle quantité de nœuds avec des degrés particulièrement élevés reste faible. De plus, le chevauchement de communautés et la multiplicité des communautés peuvent poser problème. En effet, il peut exister plusieurs groupes différents de capitalistes sociaux plus densément connectés entre eux, et les capitalistes sociaux peuvent prendre également part à d'autres communautés.

La méthode globale de Louvain n'a pas non plus détecté de communautés de capitalistes sociaux. En effet, la méthode a tendance à créer des partitions de très grandes ou très petites tailles [37]. Par ailleurs, d'une exécution à une autre, les résultats varient.

Néanmoins, si les capitalistes sociaux ne forment pas de *communautés* au sens de ces algorithmes ou au sens de la modularité, il semble bien qu'ils en forment dans la réalité : des communautés d'utilisateurs réunis par des hashtags destinés à gagner des abonnés et à se faire retweeter. Ceci nous conduit donc à réfléchir à de nouvelles méthodes pour extraire ces groupes de capitalistes sociaux densément connectés. Nous nous intéressons ainsi dans la prochaine Section au problème d'extraction de *sous-graphes denses*. Nous présentons la méthode des *cœurs de communautés* [104] et étudions son utilisation non

pas dans le cadre de la détection de communautés stables, mais dans celui de l'extraction de *sous-graphes denses.*

4.2 Extraction de sous-graphes denses

4.2.1 Définition du problème

Le problème de détection de sous-graphes denses est un problème difficile à formaliser. Il peut se concevoir sous de nombreuses formes. Nous rappelons dans un premier temps les principales définitions utiles pour ce problème.

Définition 4.3 (Sous-graphe)**.** *Soient* $G = (V, E)$ *un graphe et* $V' \subseteq V$. *Le* sous-graphe induit *par* V', *noté* $G[V']$, *est défini par* (V', E') *où* $E' = \left\{(u, v) \in E : u \in V' \wedge v \in V'\right\}$.

Définition 4.4 (Densité)**.** *La densité d'un graphe non-orienté* $G = (V, E)$ *est donnée par la formule suivante :*

$$den(G) = \frac{2 \cdot |E|}{|V| \cdot (|V| - 1)}$$

Observons tout d'abord qu'il n'existe pas une définition de *ce qu'est* un **sous-graphe dense** mais de nombreuses définitions de *ce que peut être* un **sous-graphe dense**. Nous en faisons ici un panorama dont le but n'est pas d'être exhaustif mais de nous aider à définir le cadre de notre étude.

Avant de dresser cet état de l'art, nous donnons les intuitions des propriétés énoncées par Kosub [61] que l'on peut souhaiter d'un sous-graphe dense :

1. **Compacité** : le diamètre du sous-graphe est faible. Pour rappel, le diamètre d'un sous-graphe est donné par le plus long des plus courts chemins entre toutes les paires de sommets de ce sous-graphe ;
2. **Adjacence** : les nœuds du sous-graphe sont adjacents ;
3. **Densité** : le nombre d'arêtes dans le sous-graphe est élevé (Définition 4.4) ;
4. **Séparation** : Les nœuds du sous-graphe sont plus connectés entre eux qu'avec l'extérieur ;
5. **Robustesse** : La robustesse du sous-graphe à la suppression d'arcs.

La **clique** est l'une des structures de sous-graphe dense les plus communément étudiées.

Définition 4.5 (Clique)**.** *Soient $G = (V, E)$ un graphe non-orienté et $V' \subseteq V$. Le sous-graphe $G[V']$ est une* clique *si pour tout $u, v \in V'$, on a $(u, v) \in E'$. Une clique est maximale s'il n'existe aucun ensemble V'' avec $V' \subset V''$ tel que $G[V'']$ est aussi une clique.*

La **clique** possède les quatre premières propriétés énumérées ci-dessus. Son **diamètre** est 1 puisque tous ses sommets sont par définition **adjacents**. La **densité** d'une clique est de 1, la valeur maximale. La quatrième propriété peut ne pas être vérifiée si la clique n'est pas maximale. En revanche, une clique n'est absolument pas **robuste** : une seule suppression d'arête au sein du sous-graphe et la clique n'en est plus une.

Si l'on s'intéresse à un réseau social d'individus dont les liens indiquent qu'ils se connaissent, la clique forme un sous-graphe d'invidus qui se connaissent tous : une famille ou un groupe d'amis par exemple. La définition de la clique jugée trop stricte pour certains cas a donné lieu à de nombreuses extensions dont certaines sont notamment décrites par Mokken [88]. Tout d'abord, la **n-clique** (Définition 4.6), qui supprime la contrainte d'**adjacence**.

Définition 4.6 (n-clique)**.** *Soient $G = (V, E)$ un graphe non-orienté et $V' \subseteq V$. Le sous-graphe $G[V']$ est une n-*clique *si pour toute paire de sommets (u, v) tels que $u, v \in V'$, le plus court chemin dans G entre u et v est inférieur à n.*

Remarquons qu'une 1-clique est une clique au sens classique. Les n-cliques [88] ne sont pas indépendantes du réseau autour d'elles. En effet, la définition n'impose pas que les chemins entre chaque paire de sommets de V' passent uniquement par des sommets de V'. Ainsi, si l'on s'intéresse à sa **compacité**, le diamètre d'une n-clique n'est pas nécessairement inférieur à n. Par ailleurs, du point de vue de la **robustesse**, supprimer des arcs à l'extérieur d'une n-clique peut en modifier la structure.

Définition 4.7 (n-clan)**.** *Soient $G = (V, E)$ un graphe non-orienté et $V' \subseteq V$. Le sous-graphe $G[V']$ est un n-*clan *si pour toute paire de sommets (u, v) tels que $u, v \in V'$, le plus court chemin dans $G[V']$ entre u et v est inférieur à n.*

Ainsi, un n-clan [88] est une n-clique de diamètre n. La **compacité** de ce sous-graphe est donc garantie par sa définition. Par ailleurs, le n-clan est un sous-ensemble plus **robuste** puisqu'une modification dans le réseau global mais à l'extérieur du clan n'a aucune influence sur le clan. Par ailleurs, les membres du clan sont au moins aussi **densément** connectés que ceux de la n-clique.

A défaut de restreindre le diamètre des n-cliques, il peut également être intéressant de s'intéresser aux sous-graphes maximaux de diamètre n.

Définition 4.8 (n-club)**.** *Un n-*club *d'un graphe $G = (V, E)$ est un sous-graphe maximal de G de diamètre n.*

Ces nouvelles structures de sous-graphes denses permettent aux théoriciens des réseaux de disposer d'outils plus souples pour décrire des ensembles de nœuds fortement connectés. Si l'on reprend l'exemple d'un réseau social d'indivus, ces sous-graphes réprésentent maintenant des groupes de *connaissances mutuelles*. Borgatti et al. [12] introduisent également d'autres définitions de sous-graphes denses : les LS-Sets et les Lambda-Sets. Ces structures ont pour objectif de donner des garanties sur la propriété de **séparation**. En revanche, les nœuds de ces sous-graphes ne sont pas nécessairement **adjacents**, leur diamètre peut être élevé et aucune garantie sur la **densité** n'est donnée.

Comme nous venons de le voir, les définitions issues de celle de la clique donnent de fortes garanties quant aux propriétés précédemment évoquées. Cependant l'énumération de ces sous-graphes maximaux est bien souvent un problème NP-**difficile**. Par exemple, Komusiewicz et al. [59] montrent que trouver le sous-graphe de taille k le plus dense est NP-**difficile**. D'après Lee et al. [72], l'extraction de *quasi-cliques*, sous-graphes dont la densité est supérieure à un certain seuil est un problème NP-**complet**. Ainsi, nous utilisons pour ces travaux deux autres définitions : celles de *k-cœur* et de *k-cliques-communauté* dont les algorithmes de détection sont plus efficaces.

Définition 4.9 (k-cœur)**.** *Un k-*cœur *d'un graphe non orienté $G = (V, E)$ est un sous graphe maximal de G dans lequel chacun des nœuds est de degré supérieur ou égal à k.*

Définition 4.10 (k-cliques-communauté)**.** *Une k-*cliques-communauté *d'un graphe non orienté $G = (V, E)$ est l'union de toutes les cliques de taille k qui partagent $k-1$ nœuds chacune avec au moins l'une des autres cliques de l'union.*

La détection des *k-cœurs* se fait en complexité $O(m)$ [1] d'après Batagelj et al. [6], avec m le nombre d'arêtes du graphe. La détection des *k-cliques-communautés* telle que définie par Palla et al. [92] est basée sur l'énumération des cliques de taille k. Puisque l'on considère k fixé, énumérer toutes les cliques de taille k est un problème de la classe **XP**, i.e. il existe un algorithme en $O(n^{f(k)})$ avec n le nombre de sommets du graphe. En l'occurence, Vassilevska [121] montre qu'il existe un algorithme en $O(n^{0.792 \cdot k})$. Avec k suffisamment faible, le calcul devient donc possible. Par ailleurs, puisque les réseaux du réel sont creux, les algorithmes à base de stratégies d'*élagage* permettent de résoudre des problèmes difficiles en temps raisonnable comme le montrent Pattabiramn et al. [93] qui s'intéressent au problème NP-**difficile** de recherche de la clique de taille maximum.

Par ailleurs, dans ces deux définitions, la densité des sous-graphes détectés augmente avec le paramètre k. Ce genre de propriété est particulièrement intéressant pour notre problème. En revanche, la taille des sous-graphes détectés ainsi que leur nombre diminuent avec l'augmentation du paramètre k : les conditions sont plus strictes. On observe

1. La notation O (*Big-Oh*) permet de donner des *ordres de grandeur* sur la complexité d'un algorithme.

donc un compromis entre la taille des sous-graphes détectés et leur densité. Un paramètre k bas permet une extraction plus exhaustive quand un paramètre k élevé offre des garanties plus fortes sur la densité des sous-graphes retournés par l'algorithme.

Théorème 1. *Le nombre d'arêtes minimal d'un k-cœur à l sommets est donné par*

$$\frac{l \cdot k}{2}$$

Notons qu'un k-cœur de taille $l = k+1$ est une clique.

Théorème 2. *Le nombre d'arêtes minimal d'une k-cliques-communauté à l sommets est donné par*

$$\frac{(k-1) \cdot k}{2} + (l-k) * (k-1)$$

Preuve 1. La k-cliques-communauté de plus petite taille est la k-clique à k sommets. C'est une clique de taille k qui contient donc $\frac{(k-1)\cdot k}{2}$ arêtes. Pour ajouter une autre clique à cette k-cliques-communauté, celle-ci doit partager $k-1$ nœuds avec une autre clique de la k-cliques-communauté. Donc en ajoutant une nouvelle clique de taille k, on ajoute au moins $\frac{(k-1)\cdot k}{2} - \frac{(k-2)\cdot(k-1)}{2} = (k-1)$ arêtes. Par conséquent, si la k-cliques-communauté est de taille l, le nombre d'arêtes minimum est donné par celui de la clique de taille k plus $(l-k)$ ajouts de $k-1$ arêtes.

Nous avons introduit le problème de détection de sous-graphes denses ainsi que les propriétés attendues pour un sous-graphe dense. Nous étudions maintenant l'utilisation de la méthode des *cœurs de communautés* pour la détection de *sous-graphes denses*.

4.2.2 Les cœurs de communautés

Par définition, l'optimisation de la modularité [42] permet d'obtenir une partition du graphe telle que pour chaque partie, ses nœuds sont plus connectés entre eux que vers l'extérieur. Il s'agit ainsi d'obtenir un ensemble de parties les plus denses possibles. Les algorithmes d'optimisation de la modularité tels que l'algorithme de Louvain par Blondel et al. [10] sont généralement non-déterministes et il est ainsi impossible d'obtenir une partition stable en les appliquant. Les cœurs de communautés ont été introduits par Seifi et al. [105] pour fournir un consensus stable entre différentes partitions retournées par un algorithme d'optimisation non-déterministe.

Méthodologie. Soit $G = (V, E)$ un graphe non orienté. Nous lançons λ exécutions de l'algorithme de Louvain en mode non-déterministe sur G. Nous construisons à partir des résultats un nouveau graphe non-orienté et valué $G_C = (V, E_C, \omega_C)$ où pour tous sommets

u, v de G, l'arête (u, v) appartient à E_C s'ils font partie *au moins une fois* de la même communauté dans les différentes partitions produites par l'algorithme de Louvain. La fonction de poids $\omega_C(u, v)$ a dans ce cas pour valeur le nombre de fois où u et v apparaissent dans une même communauté. Nous utilisons enfin un seuil $\alpha \in [0, 1]$ pour supprimer les arêtes (u, v) de E_C vérifiant $\omega_C(u, v) < \alpha \times \lambda$. Nous conservons ainsi les arêtes représentant le fait que des sommets sont apparus ensemble au moins $\alpha \times \lambda$ fois lors des différentes exécutions de l'algorithme de Louvain [10]. Les cœurs de communautés sont finalement obtenus en appliquant un algorithme de détection de composantes connexes : chaque composante est un cœur.

Puisque la modularité optimise une partition telle que les parties sont des ensembles dont les nœuds sont plus *densément connectés* (Définition 4.4) entre eux qu'avec l'extérieur, nous avons l'intuition que si l'algorithme de Louvain place des noeuds dans la même communauté à plusieurs reprises, c'est que ces nœuds sont plus fortement connectés les uns aux autres. Nous formulons donc l'hypothèse suivante :

Hypothèse 1. *Les cœurs de communautés permettent l'extraction de sous-graphes denses. Par ailleurs, le paramètre α permet de guider cette extraction : plus le paramètre est élevé, plus les sous-graphes extraits sont denses.*

Nombre, taille et densité des cœurs. La Thèse de Seifi [104] fournit des résultats quant au nombre de cœurs retournés par l'algorithme des *cœurs de communautés*, et ce en fonction du seuil α. Ces résultats sont générés pour trois réseaux :

— Le réseau *email* [48] qui représente les messages électroniques échangés entre les membres de l'Université de Rovira i Virgili ;

— Le réseau de collaboration *condmat* [91] qui relie les chercheurs ayant co-écrit au moins un article ;

— Le réseau *internet* qui est une capture de la structure créée par les systèmes autonomes de l'internet.

Comme nous pouvons le constater sur la Figure 4.3, plus α augmente, plus le nombre de cœurs augmente. Par ailleurs, la Figure 4.4 nous montre que plus α augmente, plus la taille des cœurs diminue. Seifi [104] précise qu'avec un seuil proche de 0, la taille des cœurs observés est parfois supérieure à la taille des communautés. Par contre, avec α proche de 1, un grand nombre de cœurs ne sont constitués que d'un seul sommet que Seifi [104] appelle *cœurs triviaux*. Ces valeurs du seuil α ne sont donc pas pertinentes dans notre contexte de détection de sous-graphes denses.
Nous observons également sur la Figure 4.5 que jusqu'à un α proche de $0,5$, la taille du cœur le plus grand -*cœur géant*- est proche de celle du réseau. Ainsi, dans la suite de

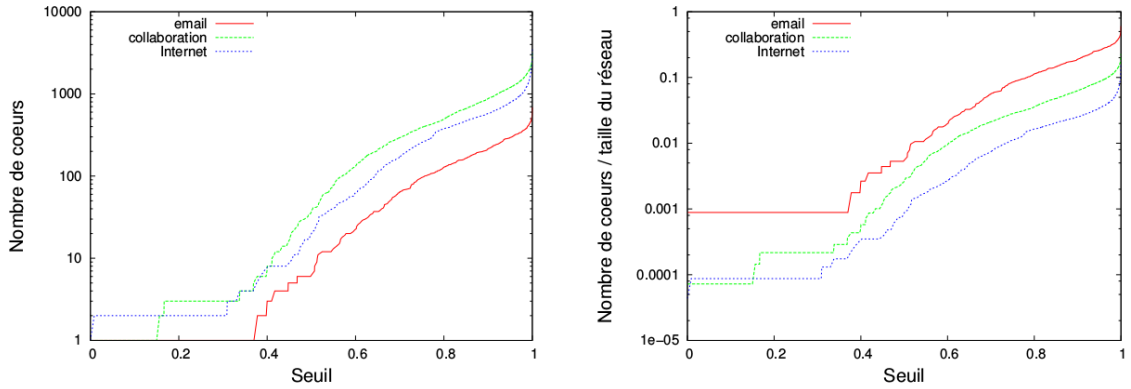

FIGURE 4.3 – Le nombre de cœurs en fonction du paramètre α sur les réseaux *email*, *condmat* et *condmat*. Figure issue de Seifi [104].

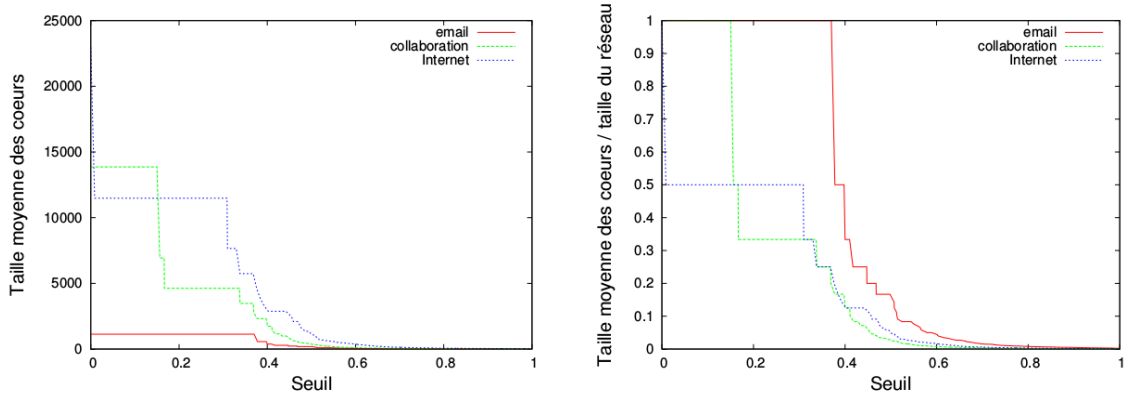

FIGURE 4.4 – Les tailles des cœurs en fonction du paramètre α sur les réseaux *email*, *condmat* et *condmat*. Figure issue de Seifi [104].

nos expérimentations, nous nous intéresserons la plupart du temps à des valeurs de α supérieures ou égales à 0,5.

Dans la suite de nos expérimentations, nous utilisons les réseaux :

— *Hamsterster* [60] qui contient les relations d'amitié et familiales entre les utilisateurs du site web hamsterster.com ;

— *arXiv astro-ph* [74] qui représente les liens entre chercheurs ayant co-publié dans la section Astrophysique d'arXiv ;

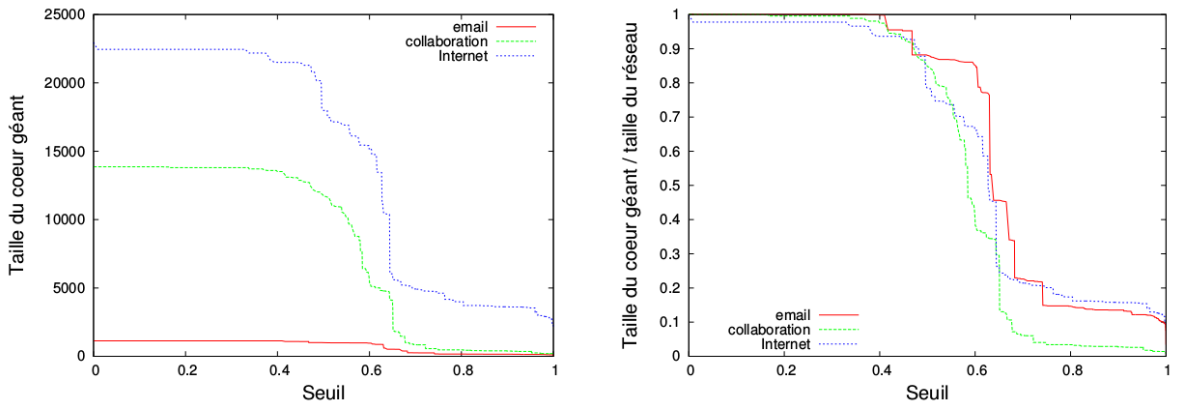

FIGURE 4.5 – Les tailles des plus grands cœurs en fonction du paramètre α sur les réseaux *email*, *condmat* et *condmat*. Figure issue de Seifi [104].

— *Facebook wall* [122] dont les liens représentent des messages Facebook d'utilisateurs sur le mur d'autres utilisateurs ;

— *Brightkite* [22], un réseau social.

Réseau	**n**	**m**	$\sum_{i,j} w_{ij}$
Hamsterster	1.858	12.534	25.068
arXiv astro-ph	18.771	198.050	396.100
Facebook wall	46.952	274.086	876.993
Brightkite	58.228	214.078	428.156

TABLE 4.1 – Nombre de nœuds n, de liens m et somme des poids $\sum_{i,j} w_{ij}$ des réseaux de nos expérimentations.

Nous pouvons observer les propriétés des réseaux dans la Table 4.1. Nous remarquons notamment que *Facebook wall* est un graphe valué : nous considérons des poids sur les arcs. Par ailleurs, *Facebook wall* est un réseau orienté. Les trois autres réseaux sont non-orientés et non valués. Nous voyons par l'intermédiaire de ce constat l'un des avantages de la méthode des cœurs de communautés : elle prend en entrée des graphes orientés ou non, valués ou non et en tient compte lors du calcul. Considérer les poids des liens est important si l'on considère par exemple des flux de mails où plusieurs mails peuvent être échangés entre deux personnes, des réseaux de co-publication où plusieurs publications peuvent être co-écrites par deux chercheurs, ou encore des réseaux sociaux où l'intensité des liens entre individus peut être valuée. La méthode des k-cliques-communauté n'est par exemple pas capable de considérer des liens valués. Notons qu'il n'existe pas de défi-

nition d'une clique spécifique aux réseaux valués.

Tout d'abord, remarquons que la méthode des *cœurs de communautés* produit des résultats interprétables de façon hiérarchique (Figure A.2). En effet, il existe une relation arborescente entre les cœurs obtenus avec un seuil α faible et les cœurs retournés par l'algorithme avec un seuil α plus élevé. Cette construction se fait naturellement étant donnée la définition des *cœurs de communautés* : si $x \geq y$ et que deux nœuds sont dans la même communauté x fois alors ils le sont aussi y fois. Plus α augmente et plus les cœurs sont éclatés : on obtient des cœurs de plus petite taille qui sont plus significatifs puisque les sommets de ces cœurs sont plus souvent regroupés par l'algorithme de détection de communautés.

Dans les Figures 4.6, 4.7 et 4.8, nous pouvons observer la distribution de la densité des cœurs détectés par la méthode, et ce en fonction du seuil α. Notons que la densité sur la Figure 4.8 est parfois supérieure à 1 puisque *Facebook wall* est valué. Comme nous pouvons le voir, l'augmentation du seuil conduit globalement à l'augmentation du nombre de cœurs. Mais il est surtout pertinent pour nous d'observer que parmi ces cœurs, nous observons des cœurs plus denses.

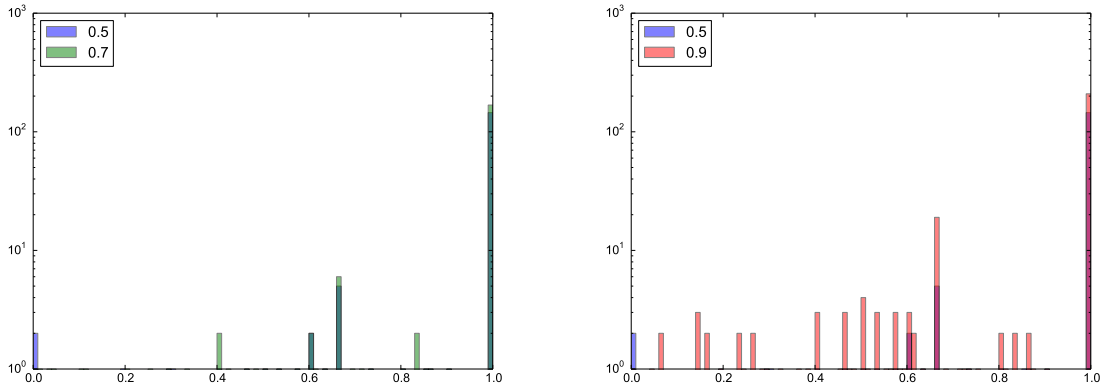

FIGURE 4.6 – La distribution de la densité des cœurs en fonction du paramètre α sur le réseau *Hamsterster*. L'ordonnée est à l'échelle logarithmique.

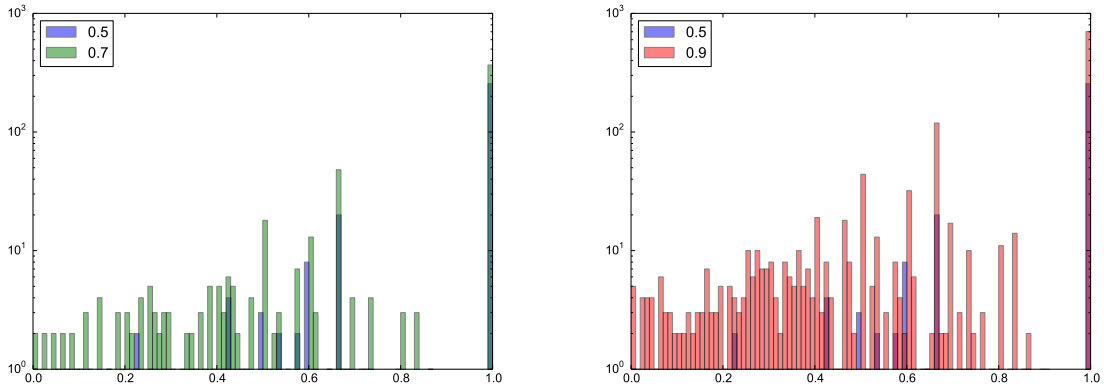

FIGURE 4.7 – La distribution de la densité des cœurs en fonction du paramètre α sur le réseau *astro-ph*. L'ordonnée est à l'échelle logarithmique.

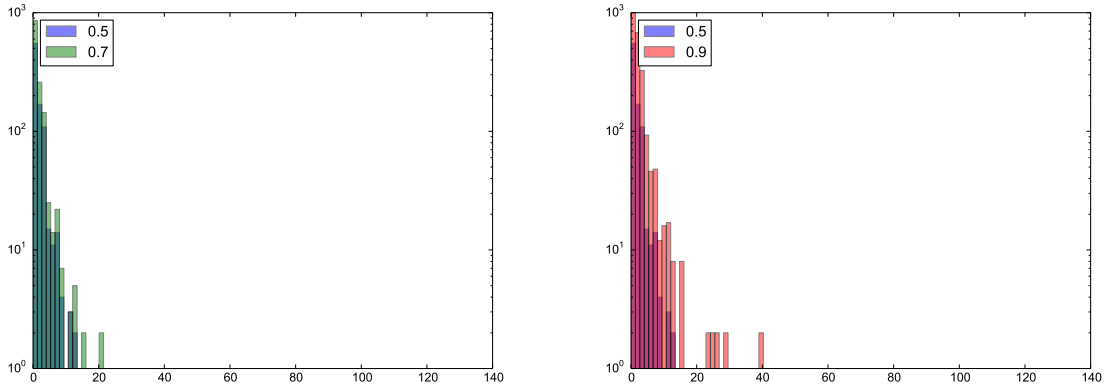

FIGURE 4.8 – La distribution de la densité des cœurs en fonction du paramètre α sur le réseau *Facebook wall*. L'ordonnée est à l'échelle logarithmique.

Sur les Figures 4.9 et 4.10, on observe la taille des cœurs en fonction de leur densité, et ce, relativement au seuil α. On remarque que la taille des cœurs diminue, observation déjà faite par Seifi [104], mais qu'ils sont également plus denses. Il est vrai que le dénominateur dans la Définition 4.4 de la densité décroît de façon quadratique en fonction du nombre de nœuds. Cependant, nous conjecturons que l'augmentation du seuil α est une bonne heuristique pour obtenir des sous-graphes réellement plus denses.

Nous venons de décrire plus en détails la méthode des cœurs de communautés. Dans la prochaine Section, nous proposons quelques évaluations préliminaires de cette méthode

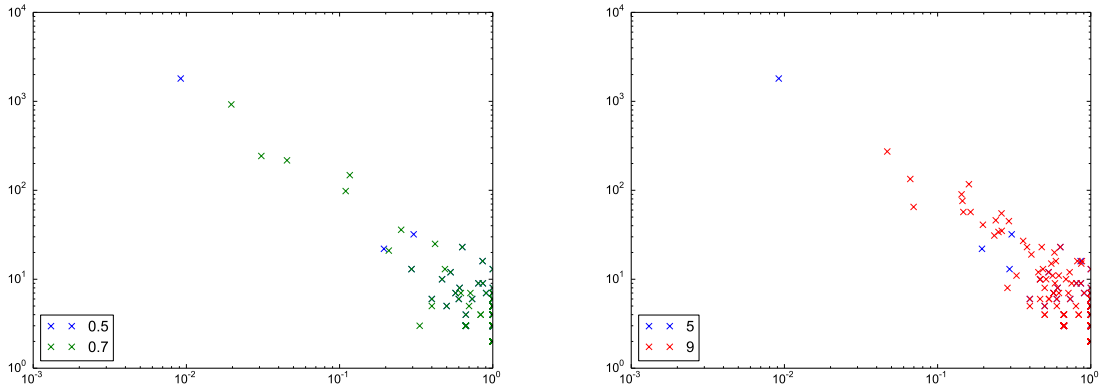

FIGURE 4.9 – Les tailles des cœurs (en ordonnée) en fonction de leur densité (en abscisse) et du paramètre α sur le réseau *Hamsterster*. Les deux axes sont à l'échelle logarithmique.

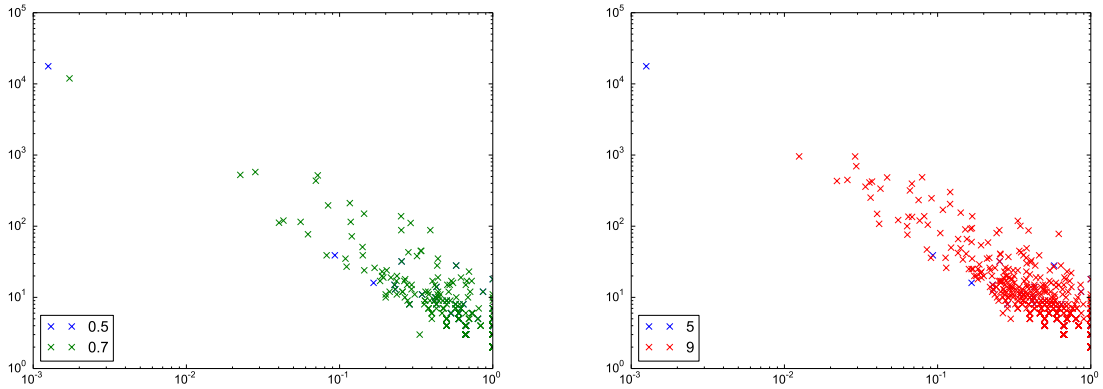

FIGURE 4.10 – Les tailles des cœurs (en ordonnée) en fonction de leur densité (en abscisse) et du paramètre α seuil sur le réseau *astro-ph*. Les deux axes sont à l'échelle logarithmique.

pour la détection de sous-graphes denses. Nous comparons notamment les résultats de cette méthode en terme de densité et de complexité sur des réseaux jouets et du réel.

4.2.3 Cœurs de communautés : pistes d'évaluation

Nous commençons tout d'abord par évaluer la performance des algorithmes des *k-cœurs*, *k-cliques-communauté* et *cœurs de communautés* sur des graphes *jouets*. Nous verrons que ces graphes nous permettent de constater que l'algorithme des *k-cœurs* ne semble pas adapté pour la détection de sous-graphes denses. Nous constaterons également que, bien que l'algorithme des *k-cliques-communauté* produisent des résultats

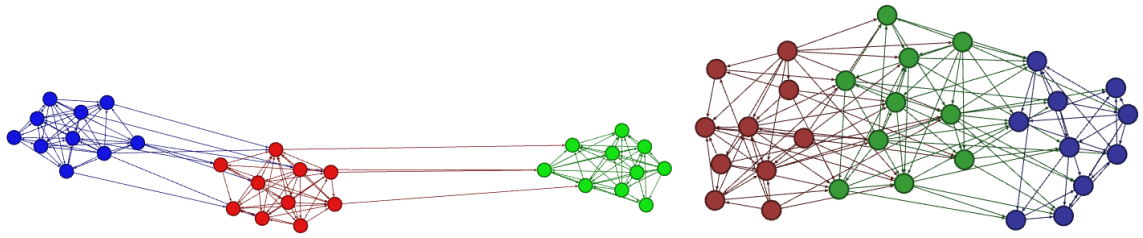

FIGURE 4.11 – Deux graphes constitués de 3 pseudo-cliques à 10 sommets. À gauche, 10% des arêtes sont supprimées aléatoirement et ajoutées entre celles-ci. À droite, les deux paramètres r_{in} et r_{out} ont pour valeur 40%.

performants sur cet ensemble de graphes, son temps d'exécution est relativement lent sur de petits réseaux du réel, rendant son utilisation difficilement envisageable.

Bien que ces observations n'aient pas de réelle valeur de vérité, elles semblent cependant indiquer le besoin de développer une méthode plus efficace pour calculer de tels sous-graphes. Si nos résultats ne permettent pour l'instant pas d'affirmer que les *cœurs de communautés* constitueront une solution fiable, ils semblent néanmoins indiquer que cette approche pourrait être pertinente.

Un benchmark de pseudo-cliques. Notre benchmark d'évaluation est constitué de p cliques de taille n. Un pourcentage r_{in} de leurs arêtes est supprimé dans chaque clique et r_{out} pourcent du nombre total d'arêtes est ajouté entre les cliques. Lors de ces deux opérations, les arêtes sont choisies aléatoirement. Ces paramètres permettent de transformer les cliques en *pseudo-cliques* qui peuvent être difficiles à détecter. Plus ces paramètres sont élevés, plus le graphe généré ressemble à un graphe aléatoire (Figure 4.11).

Nous appliquons à ce benchmark de *pseudo-cliques* les algorithmes de détection de *k-cœurs* et de *k-cliques-communautés* implémentés dans *SNAP* [75], ainsi que la méthode des cœurs de communautés. Nous observons avec quels paramètres k les algorithmes détectent les *cliques initiales* auxquelles ont été enlevées des arêtes comme des sous-graphes denses. Tout d'abord, même en modifiant la valeur du paramètre k, on observe sur la Table 4.2 que la méthode des k-cœurs n'est pas adaptée à la détection de ce genre de sous-graphes denses. En effet, lorsque k est suffisamment petit, le graphe entier est un *k-cœur*. Sinon, aucun *k-cœur* n'est détecté. En revanche, le temps d'exécution de l'algorithme est très rapide. L'algorithme de détection des *k-cœurs* va donc nous servir comme point de comparaison en terme de temps d'exécution, il s'agira de s'en rapprocher le plus possible tout en augmentant la qualité des sous-graphes détectés. On observe en revanche que l'algorithme de détection des *k-cliques-communautés* retourne des résultats pertinents.

$r_{in} = r_{out}$	**k-cœur**	**k-cliques-communauté**	**cœurs**
0,1	Jamais	$k \geq 4$	$\alpha \geq 0,1$
0,2	Jamais	$k \geq 4$	$\alpha \geq 0,1$
0,3	Jamais	$k \geq 5$	$\alpha \geq 0,2$
0,4	Jamais	Jamais	$\alpha \geq 0,7$

TABLE 4.2 – Valeurs de k pour lesquelles l'algorithme retourne les cliques initiales comme sous-graphes denses.

Avec un k suffisamment grand, les cliques initialement construites sont retournées par l'algorithme. Lorsque k est plus petit, le graphe complet est retourné par l'algorithme. Ceci prouve que le paramètre k permet effectivement d'ajuster le compromis entre la densité des graphes retournés et leur nombre et taille. Ceci semble important dans le cadre d'un algorithme d'exploration de sous-graphes denses. Enfin, les cœurs de communautés retournent les cliques initiales dans tous les cas lorsque α est suffisamment grand. Plus les *pseudo-cliques* sont floues, plus il est nécessaire d'augmenter α. Lorsque α est petit, tout le graphe est en général retourné. On retrouve donc encore ce compromis entre la densité des graphes retournés et leur nombre et taille.

Complexité. Nous comparons les temps d'exécution des méthodes des *cœurs de communautés*, des *k-cliques-communautés* et des *k-cœurs*. Nous évaluons deux versions de l'algorithme des *cœurs de communautés* : la version *classique* qui calcule les cœurs pour α de $0,1$ à $0,9$ et la version *rapide* qui ne s'intéresse qu'au sous-ensemble de ces valeurs de α qui sont supérieures ou égales à $0,5$. Cela a pour effet de réduire l'utilisation de la mémoire et le temps d'exécution comme nous le voyons sur la Table 4.3. Par ailleurs, nous faisons également varier le nombre d'exécutions λ du Louvain, qui passe de 20 à 40.

Réseau	**k-cœur**	**rapide**$_{40}$	**rapide**$_{20}$	**classique**$_{20}$	**k-cliques-communauté**
Hamsterster	< 1s	5s	4s	5s	16s
astro-ph	5s	2m53s	2m26s	3m42s	18m36s
Facebook wall	5s	26m40s	23m29s	33m8s	340m24s
Brightkite	3s	34m7s	24m54s	36m48s	None

TABLE 4.3 – Temps d'exécution des *k-cœurs*, des *k-cliques-communautés* et de la méthode des *cœurs de communautés classique* et *rapide* avec 20 ou 40 exécutions du Louvain : le nombre d'exécutions est mis en indice.

Le temps d'exécution de l'algorithme des *k-cliques-communautés* et sa complexité en espace sont trop élevés pour traiter tous les graphes du réel. Même avec une machine à 64Go de RAM, l'algorithme est à court de mémoire sur le réseau *Brightkite*. Au contraire,

les k-cœurs procèdent en dessous des 10 secondes pour chacun de nos réseaux. L'algorithme des *cœurs de communautés* présente quant à lui des temps d'exécution autour de la demie-heure. En revanche, sa complexité en espace est élevée : il s'agit de retenir les paires de sommets placées dans la même communauté. Il est donc malheureusement impossible de l'utiliser sur un réseau de la taille de **Twitter**, même avec $64Go$ de RAM. Même la méthode *rapide* en est incapable. Ainsi, même si les performances de l'algorithme sont prometteuses et son temps d'exécution également, l'utilisation de la mémoire est encore beaucoup trop coûteuse pour réellement faire un passage à l'échelle du très grand réseau de **Twitter**.

Conclusion et perspectives de recherche

Résumé des travaux réalisés

Les méthodes des capitalistes sociaux. Dans cette thèse, nous avons proposé une analyse du *capitalisme social* sur **Twitter**. Les capitalistes sociaux utilisent des procédés tels que **IFYFM** ou **FMIFY** (Section 1.1) pour gagner des abonnés et ainsi maximiser leur *capital social*. Dans un premier temps, nous nous sommes basés sur ces méthodes d'échange d'abonnements pour proposer une technique de détection de ces utilisateurs à l'échelle du réseau de **Twitter.** Cette dernière est basée sur deux mesures de similarité des ensembles d'abonnés et d'abonnements : le *ratio* et l'*indice de chevauchement* (Section 1.1.1), et est calibrée en utilisant une liste de 100.000 capitalistes sociaux détectés par Ghosh et al. [40].

Nous nous intéressons ensuite aux hashtags dédiés au capitalisme social. Nous montrons qu'un éco-système de hashtags dédiés à l'utilisation des méthodes **IFYFM** et **FMIFY** existe. Nous décrivons un *jeu de données* contenant des tweets tagués avec l'un de ces hashtags : *#TeamFollowBack*. Grâce à ces données, nous montrons notamment que la plupart de ces utilisateurs tweetent manuellement. Nous utilisons également les tweets récoltés pour créer un compte automatique qui applique les méthodes des capitalistes sociaux. Ce *bot* gagne un grand nombre d'abonnés, jusqu'à 250 par jour. Ce dernier est par ailleurs retweeté jusqu'à 700 fois par jour. Ce *bot* nous permet ainsi de confirmer l'efficacité des méthodes des capitalistes sociaux. D'ailleurs, l'efficacité de leurs méthodes est également confirmée par l'étude de leur évolution entre 2009 et 2013, qui met en lumière l'augmentation de leur nombre d'abonnés -plus forte que celle de leur nombre d'abonnements- et la forte baisse de leur ratio pour un grand nombre d'entre eux.

Visibilité. Après avoir montré l'efficacité des méthodes de capitalisme social, nous procédons à l'analyse de leur visibilité sur le réseau **Twitter**. Il s'agit de connaître la position de ces utilisateurs dans le réseau pour savoir s'ils sont isolés entre eux, connectés à une vaste frange du réseau, centraux ou périphériques. Pour cela, nous décrivons une approche pour calculer les rôles communautaires des capitalistes sociaux. Cette approche propose l'étude de la position des nœuds du réseau relativement à leur communauté. Cela nous permet de disposer d'une information plus riche que celle donnée par l'étude des simples voisinages, tout en évitant les coûts calculatoires trop élevés qu'auraient entraîné des mesures à l'échelle du réseau.

Nous introduisons donc tout d'abord la notion de rôle communautaire telle que présentée par Guimerà et Amaral [49], qui proposent d'utiliser une mesure de *connectivité interne* à la communauté et une mesure de *connectivité externe*, puis de déterminer les rôles à partir de *seuils* sur ces dernières. Nous montrons que la mesure de connectivité externe proposée par Guimerà et Amaral [49] mélange plusieurs aspects de la connectivité externe. Celle-ci est donc imprécise et difficile à interpréter, surtout dans le cas de grands réseaux comme le nôtre. Ainsi, puisque nous révisons cette dernière en proposant trois nouvelles mesures de connectivité externe, nous sommes amenés à faire évoluer les seuils destinés à la détermination des rôles. Nous montrons que la méthode employée par les auteurs ne semble ni prendre en compte les différences causées par le changement de la méthode de détection de communautés, ni être adaptée à nos données. Pour déterminer les seuils, nous proposons donc d'utiliser une méthode d'apprentissage non supervisé capable de s'adapter aux changements précédemment mentionnés.

Nous appliquons ainsi notre approche au réseau de Cha et al. [20] afin d'étudier les positions des capitalistes sociaux dans le réseau. Nous réussissons avec notre approche à obtenir une séparation des noeuds du réseau en rôles communautaires interprétables selon la terminologie de Guimerà et Amaral [49]. Nous montrons ainsi que les capitalistes sociaux occupent des rôles qui leur donnent de la *visibilité* dans le réseau. Ceux de haut degré sont des *hubs* très *connectés* aux autres communautés. Quant à ceux de faible degré, la majorité sont bien *connectés* aux autres communautés.

Influence. Puisque la méthode des rôles communautaires démontre la visibilité des capitalistes sociaux sur une large frange du réseau, nous nous intéressons à l'influence de ces utilisateurs sur le réseau. Nous choisissons l'angle de l'influence selon les outils très largement utilisés par les entreprises marketing, les gestionnaires de communauté en ligne et les utilisateurs **Twitter**. Nous abordons ainsi l'influence selon **Klout**, **Kred** et **Twitalizer**, et nous concentrons en particulier sur **Klout**, l'outil le plus utilisé selon nous. Nous montrons que ces outils considèrent un grand nombre de capitalistes sociaux comme *influents*. À partir de ce constat, nous construisons un jeu de données constitué de *capitalistes so-*

ciaux et d'utilisateurs *réguliers* contenant un grand nombre d'attributs décrivant l'utilisation de **Twitter** par ces utilisateurs. Nous étudions ce jeu de données puis implémentons une méthode de classification supervisée afin de discriminer les utilisateurs réguliers des *capitalistes sociaux*. Nous utilisons pour cela le modèle de la *régression logistique* qui nous permet notamment d'interpréter quels sont les attributs les plus corrélés au *capitalisme social*. Enfin, nous implémentons notre méthode dans une application en ligne, enrichie par de nouveaux attributs et destinée à permettre l'évolution du modèle.

Extraction de sous-graphes denses. Dans le Chapitre 1, nous montrons que les capitalistes sociaux sont très connectés entre eux : leurs voisinages sont constitués majoritairement de capitalistes sociaux et leur coefficient de clustering est plus élevé en moyenne. Nous cherchons donc à établir que les capitalistes sociaux forment des communautés. Pour cela, nous appliquons plusieurs méthodes de détection de communautés *ego-centrées* et *multi-ego-centrées* représentatives de l'état de l'art, ainsi que l'algorithme de Louvain [10]. Ces méthodes ne forment pas de communautés de *capitalistes sociaux*. Nous conjecturons néanmoins que ces derniers forment des *sous-graphes denses*. Nous formalisons donc un cadre au problème d'extraction de *sous-graphes denses*. Puis nous étudions la possibilité d'utiliser la méthode des cœurs de communautés dans ce cadre. Nous proposons ainsi une étude préliminaire sur la densité des cœurs retournés par la méthode. Puis nous montrons un début d'évaluation en comparant cette méthode à celle des *k-cœurs* et des *k-cliques-communauté* sur des *graphes jouets*, notamment en terme de complexité.

Perspectives de recherche

Détection. Nous avons réduit notre champ d'étude aux capitalistes sociaux sur **Twitter** dans cette thèse. Néanmoins, comme nous l'avons mentionné dans l'Introduction, des outils d'abonnement et de désabonnement massifs existent également pour **Instagram** par exemple. On trouve également beaucoup d'applications d'échanges de *likes* entre pages **Facebook**. Certaines applications permettent par ailleurs d'obtenir des vues **Youtube**. Ainsi, il peut être intéressant d'observer si des méthodes de détection basées sur des mesures de similarité entre voisinages sont capables de détecter le même genre d'utilisateurs sur ces différents réseaux.

Rôles communautaires. La méthode de détection des rôles communautaires que nous avons développée dans le Chapitre 2 nous a permis de conduire notre étude de la visibilité des capitalistes sociaux. Cependant, nous souhaiterions considérer cette méthode pour l'étude d'autres réseaux.

Dans un premier temps, afin de réduire la taille des données, il peut être intéressant de ne plus s'intéresser aux noeuds ultra-périphériques. En effet, ceux-ci constituent la grande majorité du réseau et ne pas les considérer engendrerait donc une réduction importante des calculs. Ainsi, nous proposons d'effectuer la détection de communautés sur la totalité du réseau mais d'appliquer l'analyse de regroupement à un sous-ensemble de noeuds excluant ceux de très faible degré. Nous pourrions ainsi caractériser plus précisément quels noeuds supprimer de l'analyse et les gains en terme de temps de calcul. Par ailleurs, cela permettrait d'appliquer une méthode d'apprentissage non supervisé plus stable.

Plusieurs méthodes existent pout stabiliser les *k-moyennes*. Celles-ci proposent par exemple des heuristiques pour le choix des centres comme la méthode *k-moyennes++* introduite par Arthur et Vassilvitskii [3]. La méthode *x-moyennes* proposée par Pelleg et al. [95] permet elle de calculer efficacement le k optimal. D'autres comme les *k-moyennes globales* introduites par Likas et al. [77] suggèrent d'utiliser les centres détectés avec $k-1$ pour réexécuter la méthode avec k. Par ailleurs, à moins large échelle, la méthode de détection de communautés utilisée peut également être changée. On pourra par exemple opter pour OSLOM [67] dont les performances sont meilleures sur les benchmarks du Chapitre 2.

Ceci nous amène donc à la question de l'influence de l'algorithme de détection de communautés sur nos mesures. Nous avons vu que différentes méthodes conduisent à différentes distributions de nos mesures dans le Chapitre 2. Cependant, une distribution différente des valeurs des mesures n'implique pas forcément une distribution différente des utilisateurs au sein des rôles détectés. L'influence des méthodes de détection de communautés pourrait ainsi être un champ d'expérimentation. Par ailleurs, similairement, les caractéristiques des graphes telles que celles décrites pour les réseaux du LFR dans le Chapitre 2 (lois de puissances, degré moyen, etc) peuvent influer sur la distribution de nos mesures et des rôles : il serait ainsi intéressant d'observer l'impact de ces caractéristiques.

Influence des capitalistes sociaux. Tout d'abord, nous allons pouvoir, grâce à notre application web, continuer à récolter des données sur les capitalistes sociaux et sur les utilisateurs **Twitter** réguliers. Pour chacun de ces utilisateurs, nous aurons plus d'attributs comme décrits dans le Chapitre 3. Il s'agira donc d'étudier l'utilité de ces nouveaux attributs pour la détection de capitalistes sociaux et ainsi de faire évoluer notre classifieur et l'application web de détection des capitalistes sociaux.

Par ailleurs, nous l'avons déjà évoqué dans le Chapitre 3, le problème d'évaluer l'influence d'un compte **Twitter** est très complexe. Nous avons débuté des travaux utilisant un jeu de données qui contient des utilisateurs **Twitter** manuellement annotés comme influents ou non par une entreprise spécialiste du domaine. Les méthodes basées sur la classification supervisée qui utilisent des données issues de **Twitter** produisent des modèles

insuffisamment précis si les attributs fournis sont ceux du Chapitre 3. Des méthodes à base de traitement automatique de la langue semblent par exemple parvenir à obtenir des résultats plus pertinents. Il est donc intéressant d'explorer de nouvelles pistes telles que celles-ci pour tenter de caractériser les utilisateurs influents.

Néanmoins, nous restons persuadés que les efforts conjugués d'informaticiens, de sociologues et de spécialistes des médias permettraient d'étudier de façon approfondie la question : caractériser ce qu'est l'influence sur **Twitter** et comment elle se traduit sur le réseau ou dans la vie réelle, comprendre d'où elle vient, étudier la dualité entre influence sur **Twitter** et dans la vie réelle, etc. Ces questions sont trop complexes pour être étudiées d'un seul point de vue et sur un seul jeu de données. Ainsi, des articles comme Bond et al. [11] apportent des premières réponses intéressantes quant à l'impact que peuvent avoir les réseaux sociaux numériques sur la vie réelle. Les auteurs montrent qu'un message **Facebook** peut avoir des conséquences sur l'abstention : voir que nos amis ont voté nous incite à voter.

Extraction de sous-graphes denses. Notre étude des *cœurs de communautés* est focalisée sur la densité. En revanche, nous voyons dans le Chapitre 4 qu'un sous-graphe dense peut être caractérisé par 4 autres propriétés : l'adjacence, la compacité, la séparation et la robustesse. Nous savons que les cœurs ne garantissent pas l'adjacence des noeuds qui les composent. En revanche, il serait intéressant d'étudier empiriquement la compacité des coeurs de communautés en fonction du paramètre α. Un diamètre faible serait un bon indicateur de densité pour ces sous-graphes. De même, nous savons que la modularité permet d'optimiser de pair densité et séparation. Nous pourrions donc dans des travaux futurs considérer la propriété de séparation, afin de voir si celle-ci est améliorée avec l'augmentation du paramètre α. Il serait pertinent d'utiliser des fonctions comme l'*expansion* qui mesure respectivement le nombre de liens qui sortent du sous-graphe par noeud ou le *cut ratio* qui quantifie le nombre de liens qui sortent du sous-graphe par rapport au nombre de liens possibles. Enfin, la robustesse des coeurs de communautés en fonction de modifications du réseau pourrait également être étudiée. Si augmenter α garantit des communautés plus stables et des sous-graphes plus denses, il est possible que ces sous-graphes soient plus robustes. Karrer et al. [55] définissent une méthode pour caractériser la robustesse des communautés à des modifications mineures du réseau. Les auteurs proposent un algorithme pour *recâbler* aléatoirement certains liens du réseau initial tout en gardant la même distribution des degrés. Les communautés détectées sur le réseau initial sont comparées à celles obtenues sur le réseau dont certains liens ont été recâblés, avec des mesures comme celles présentées dans la Section 2.1.3. Cette méthode pourrait ainsi être transposée pour l'évaluation de la robustesse des cœurs de communauté.

Par ailleurs, nous n'avons pas pu appliquer la méthode des cœurs de communautés sur le réseau de **Twitter** à cause de l'espace mémoire requis. En effet, pour obtenir les résultats des cœurs de communautés, il est nécessaire de stocker les paires de sommets considérées comme faisant partie de la même communauté par les λ exécutions de l'algorithme de Louvain [10]. Même si certaines optimisations triviales sont faites pour ne pas stocker toutes les paires, la taille requise en mémoire est bien trop élevée. Il s'agirait ainsi de réfléchir à des méthodes pour améliorer le passage à l'échelle. Dans un premier temps, pour étudier les sous-graphes denses de capitalistes sociaux, nous pourrions nous focaliser uniquement sur le stockage des paires de capitalistes sociaux.

Enfin, notre caractérisation de la densité des coeurs de communautés dans le Chapitre 4 est basée sur des constats empiriques qui semblent confirmer nos intuitions. Nous souhaiterions montrer de façon appliquée que les coeurs de communautés peuvent se révéler utiles pour l'extraction de sous-graphes denses. Nous avons débuté des travaux en ce sens qui s'intéressent à des structures topologiques locales appellées *motifs* [30]. Pour en donner une définition intuitive, les *motifs* d'un réseau orienté sont les types de triangles (Figure 4.12) les plus fréquents sur ce réseau et qui semblent en caractériser sa construction. Nous souhaitons nous intéresser aux sous-graphes denses en motifs dans les réseaux orientés pour mieux comprendre comment les réseaux se forment, s'organisent et étudier s'il existe des inégalités dans la distribution des motifs dans ces réseaux. Pour cela, nous définissons le réseau de co-occurence en triangles sur lequel nous appliquons la méthode des cœurs de communautés qui est particulièrement adaptée à ce réseau valué.

Définition 4.11 (Réseau de co-occurence en triangles)**.** *Soient $D = (V, A)$ un réseau orienté et $1 \leqslant i \leqslant 13$ un type de triangle. Le* réseau de co-occurence en triangles *de type i de D est le réseau valué et non orienté $G_i = (V_i, E_i, \omega_i)$ avec :*

- $V_i = \{v \in V \ : \ v$ *apparaît dans un triangle de type* $i\}$
- $E_i = \{(v, v') \in V \times V \ :$ *il existe un triangle de type i qui contient v et v'*$\}$
- $\omega_i : E_i \rightarrow \mathbb{N}^+$ *et $\omega_i(v, v')$ représente le nombre de fois où v et v' apparaissent ensemble dans un triangle de type i.*

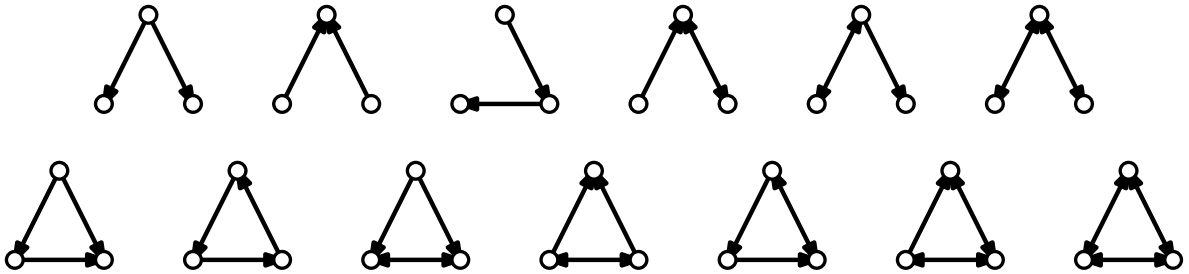

FIGURE 4.12 – Les treize triangles différents possibles dans un réseau orienté.

Détection d'évènements. Guille s'intéresse dans une partie de sa thèse [45] à la détection d'évènements, notamment en utilisant la méthode MABED qui considère les tweets contenant des mentions, ceux où le niveau d'implication de l'utilisateur est plus élevé [46]. Guille a collecté des tweets mentionnant François Hollande de Décembre 2013 à Mars 2014 dans lequel des *faux évènements* ont été détectés. Par exemple, un tweet qui véhicule une information fausse (i.e. une photo retirée par l'AFP sur demande de François Hollande d'après le tweet) posté en Septembre 2013, suscite périodiquement des vagues intenses de retweets. Une étude préliminaire qui utilise nos méthodes de détection indique que cette fausse information est majoritairement retweetée par des capitalistes sociaux. Ainsi, il serait intéressant de voir si brancher la détection de capitalistes sociaux sur MABED peut aider à détecter ce genre de *faux évènements*, à les comprendre, les quantifier et les éliminer de MABED afin d'avoir une restitution des évènements encore plus pertinente.

Bibliographie

[1] Luca Maria Aiello, Martina Deplano, Rossano Schifanella, and Giancarlo Ruffo. People are strange when you're a stranger : Impact and influence of bots on social networks. In *ICWSM*. The AAAI Press, 2012. Cité page 33.

[2] 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. Cité pages 77 and 78.

[3] David Arthur and Sergei Vassilvitskii. k-means++ : The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007. Cité page 130.

[4] Eytan Bakshy, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Everyone's an influencer : quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 65–74, 2011. Cité page 12.

[5] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439) :509–512, 1999. Cité page 60.

[6] Vladimir Batagelj and Matjaz Zaversnik. An o (m) algorithm for cores decomposition of networks. *arXiv preprint cs/0310049*, 2003. Cité page 115.

[7] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting Spammers on Twitter. In *Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)*, July 2010. Cité pages 29 and 31.

[8] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. Springer, 2006. Cité page 91.

[9] Youtube Official Blog. 1 billion subscriptions and counting. http ://youtube-global.blogspot.fr/2010/10/1-billion-subscriptions-and-counting.html. Cité page 10.

[10] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *J. of Stat. Mech. : Theory and Experiment*, 2008(10) :P10008, 2008. Cité pages 50, 51, 52, 54, 116, 117, 129, 132, and 159.

[11] Robert M Bond, Christopher J Fariss, Jason J Jones, Adam DI Kramer, Cameron Marlow, Jaime E Settle, and James H Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415) :295–298, 2012. Cité page 131.

[12] S.P. Borgatti, M.G. Everett, and P.R. Shirey. Ls sets, lambda sets, and other cohesive subsets. *Social Networks*, (12) :337–358, 1990. Cité page 115.

[13] Pierre Bourdieu. Le capital social. *Actes de la recherche en sciences sociales*, 31(1) :2–3, 1980. Cité page 13.

[14] Danah Boyd, Scott Golder, and Gilad Lotan. Tweet, tweet, retweet : Conversational aspects of retweeting on twitter. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10. IEEE, 2010. Cité page 10.

[15] Danah M. Boyd and Nicole B. Ellison. Social network sites : Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, 13(1) :210–230, 2007. Cité page 9.

[16] Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Gorke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2) :172–188, 2008. Cité page 51.

[17] Ronald S. Burt. Detecting role equivalence. *Social Networks*, 12(1) :83 – 97, 1990. Cité page 56.

[18] Suzan Burton and Alena Soboleva. Interactive or reactive ? : marketing with twitter. 28 :491–499, 2011. Cité page 12.

[19] A. Capocci, V.D.P. Servedio, G. Caldarelli, and F. Colaiori. Detecting communities in large networks. *Physica A : Statistical Mechanics and its Applications*, 352(2–4) :669 – 676, 2005. Cité page 49.

[20] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in Twitter : The million follower fallacy. In *Proceedings of ICWSM*. AAAI, 2010. Cité pages 21, 23, 25, 27, 40, 66, 77, 109, 128, 179, and 180.

[21] 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. Cité page 112.

[22] Eunjoon Cho, Seth A. Myers, and Jure Leskovec. Friendship and mobility : User movement in location-based social networks. In *Proc. Int. Conf. on Knowledge Discovery and Data Mining*, pages 1082–1090, 2011. Cité page 119.

[23] 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. Cité pages 77 and 85.

[24] Aaron Clauset. Finding local community structure in networks. *Physical review E*, 72(2) :026132, 2005. Cité pages 111, 112, and 150.

[25] Maximilien Danisch, J.-L. Guillaume, and Bénédicte Le Grand. Multi-ego-centered communities in practice. *Social Network Analysis and Mining*, 4(1), 2014. Cité pages 50, 110, 150, and 181.

[26] Maximilien Danisch, Jean-Loup Guillaume, and Bénédicte Le Grand. Towards multi-ego-centred communities : a node similarity approach. *International Journal of Web Based Communities*, 9(3) :299–322, 2013. Cité page 110.

[27] 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. Cité page 111.

[28] Shaun W. Davenport, Shawn M. Bergman, Jacqueline Z. Bergman, and Matthew E. Fearrington. Twitter versus facebook : Exploring the role of narcissism in the motives and usage of different social media platforms. *Computers in Human Behavior*, 32(0) :212 – 220, 2014. Cité page 13.

[29] David L. Davies and Donald W. Bouldin. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1(2) :224–227, February 1979. Cité page 65.

[30] 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*, 2015. Cité page 132.

[31] Nicolas Dugué and Anthony Perez. Detecting social capitalists on twitter using similarity measures. In *Complex Networks IV*, pages 1–12. Springer, 2013. Cité page 179.

[32] Nicolas Dugué and Anthony Perez. Social capitalists on Twitter : detection, evolution and behavioral analysis. *Social Network Analysis and Mining*, 4(1) :1–15, 2014. Springer. Cité page 85.

[33] Peter Eades and Qing-Wen Feng. Multilevel visualization of clustered graphs. In *Proceedings of the Symposium on Graph Drawing*, GD '96, pages 101–112, 1997. Cité page 49.

[34] P. Erdös and A. Rényi. On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6 :290–297, 1959. Cité page 60.

[35] Forbes. Social Media Metrics Startup Simply Measured Raises 20M. http://www.forbes.com/sites/kellyclay/2014/03/18/social-media-metrics-startup-simply-measured-raises-20m/. Cité page 81.

[36] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3) :75–174, 2010. Cité page 54.

[37] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1) :36–41, 2007. Cité page 112.

[38] Santo Fortunato and Andrea Lancichinetti. Community detection algorithms : A comparative analysis : Invited presentation, extended abstract. In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS '09, pages 27 :1–27 :2, ICST, Brussels, Belgium, Belgium, 2009. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). Cité page 52.

[39] L. C. Freeman. Centrality in social networks i : Conceptual clarification. *Soc. Net.*, 1(3) :215–239, 1978. Cité page 49.

[40] 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. Cité pages 12, 13, 14, 18, 19, 21, 25, 26, 32, 45, 127, and 148.

[41] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Inferring web communities from link topology. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space—structure in hypermedia systems : links, objects, time and space—structure in hypermedia systems*, pages 225–234. ACM, 1998. Cité page 49.

[42] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12) :7821–7826, 2002. Cité pages 49 and 116.

[43] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009. Cité page 106.

[44] Neil Zhenqiang Gong, Wenchang Xu, Ling Huang, Prateek Mittal, Emil Stefanov, Vyas Sekar, and Dawn Song. Evolution of social-attribute networks : measurements, modeling, and implications using google+. In *Proceedings of the 2012 ACM conference on Internet measurement conference*, pages 131–144. ACM, 2012. Cité page 9.

[45] Adrien Guille. *Information diffusion in social media : modeling and analysis*. Theses, Université Lumière Lyon 2, November 2014. Cité page 133.

[46] Adrien Guille and C. Favre. Mention-anomaly-based Event Detection and Tracking in Twitter. In *2014 IEEE/ACM International Conference on Advances in Social Network Analysis and Mining*, Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Network Analysis and Mining, pages 375–382, Beijing, China, August 2014. Cité page 133.

[47] R. Guimera and Luis A Nunes Amaral. Cartography of complex networks : modules and universal roles. *Journal of Statistical Mechanics : Theory and Experiment*, 2005(02) :P02001, 2005. Cité page 60.

[48] R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Phys. Rev. E*, 68 :065103, Dec 2003. Cité page 117.

[49] Roger Guimera and Luis A Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028) :895–900, 2005. Cité pages 47, 56, 57, 58, 59, 60, 61, 62, 63, 66, 67, 68, 128, 148, and 152.

[50] C.P.M.C.J.A. Hendricks, J.A. Hendricks, A.P.C.S.D. Schill, and D. Schill. *Presidential Campaigning and Social Media : An Analysis of the 2012 Campaign*. Oxford University Press, 2014. Cité page 12.

[51] P. W. Holland, Kathrin Blackmond Laskey, , and Samuel Leinhardt. Stochastic blockmodels : First steps. *Social Networks*, 5(2) :109 – 137, 1983. Cité page 56.

[52] Interbrand. The best 100 brands, 2014. http ://bestglobalbrands.com/2014/ranking/. Cité pages 11 and 12.

[53] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37 :547–579, 1901. Cité page 24.

[54] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter : understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, WebKDD/SNA-KDD '07, pages 56–65, 2007. Cité page 10.

[55] Brian Karrer, Elizaveta Levina, and Mark EJ Newman. Robustness of community structure in networks. *arXiv preprint arXiv :0709.2108*, 2007. Cité page 131.

[56] Klout. Klout, the standard for influence. http://www.klout.com. Cité pages 13, 75, 77, and 78.

[57] Klout. Identifying and Measuring Influencers in Social Marketing with Klout, 2014. http://simplymeasured.com/blog/2014/06/03/guide-identifying-and-measuring-influencers-in-social-marketing-with-klout/. Cité page 81.

[58] Klout. Your Klout Score : Why You Can't Afford to Ignore It, 2015. http://www.inc.com/john-boitnott/your-klout-score-why-you-can-t-afford-to-ignore-it.html. Cité page 81.

[59] Christian Komusiewicz and Manuel Sorge. Finding dense subgraphs of sparse graphs. In *Proceedings of the 7th International Conference on Parameterized and Exact Computation*, IPEC'12, pages 242–251, 2012. Cité page 115.

[60] Konect. Hamster full network dataset – KONECT, April 2015. Cité page 118.

[61] Sven Kosub. Local density. In *Network analysis*, pages 112–142. Springer, 2005. Cité page 113.

[62] Kred. Kred story. http://www.kred.com. Cité pages 13, 75, 77, and 78.

[63] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media ? In *Proc. of the 19th int. conference on World wide web*, WWW '10, pages 591–600, 2010. Cité page 40.

[64] Avinash Lakshman and Prashant Malik. Cassandra : a structured storage system on a p2p network. In *Proc. of the 28th ACM symp. on Princ. of distributed comput.*, PODC '09, pages 5–5, 2009. Cité page 179.

[65] Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E*, 80(1) :016118, 2009. Cité pages 52, 54, and 55.

[66] Andrea Lancichinetti, Mikko Kivelä, Jari Saramäki, and Santo Fortunato. Characterizing the community structure of complex networks. *PloS one*, 5(8) :e11976, 2010. Cité page 61.

[67] Andrea Lancichinetti, Filippo Radicchi, Jose J. Ramasco, and Santo Fortunato. Finding statistically significant communities in networks. *CoRR*, abs/1012.2363, 2010. Cité pages 55, 130, and 159.

[68] Kyumin Lee, James Caverlee, Krishna Y Kamath, and Zhiyuan Cheng. Detecting collective attention spam. In *Proceedings of the 2nd Joint WICOW/AIRWeb Workshop on Web Quality*, pages 48–55. ACM, 2012. Cité page 13.

[69] Kyumin Lee, James Caverlee, and Steve Webb. Uncovering social spammers : social honeypots+ machine learning. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 435–442. ACM, 2010. Cité page 13.

[70] Kyumin Lee, Brian David Eoff, and James Caverlee. Seven months with the devils : A long-term study of content polluters on twitter. Citeseer, 2011. Cité pages 13, 16, and 147.

[71] Kyumin Lee, Prithivi Tamilarasan, and James Caverlee. Crowdturfers, campaigns, and social media : Tracking and revealing crowdsourced manipulation of social media. 2013. Cité page 13.

[72] Victor E. Lee, Ning Ruan, Ruoming Jin, and Charu Aggarwal. A survey of algorithms for dense subgraph discovery. Cité page 115.

[73] E. A. Leicht and M. E. J. Newman. Community structure in directed networks. *Phys. Rev. Lett.*, 100(11) :118703, 2008. Cité pages 50 and 52.

[74] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution : Densification and shrinking diameters. *ACM Trans. Knowledge Discovery from Data*, 1(1) :1–40, 2007. Cité page 118.

[75] Jure Leskovec and Rok Sosič. SNAP : A general purpose network analysis and graph mining library in C++. http://snap.stanford.edu/snap, June 2014. Cité page 123.

[76] W.-K. Liao. Parallel k-means data clustering, Oct 2009. Cité page 63.

[77] Aristidis Likas, Nikos Vlassis, and Jakob J. Verbeek. The global k-means clustering algorithm. *Pattern Recognition*, 36(2) :451 – 461, 2003. Biometrics. Cité page 130.

[78] François Lorrain and Harrison C. White. Structural equivalence of individuals in social networks. *The Journal of Mathematical Sociology*, 1(1) :49–80, 1971. Cité page 56.

[79] Grzegorz Malewicz, Matthew H Austern, Aart JC Bik, James C Dehnert, Ilan Horn, Naty Leiser, and Grzegorz Czajkowski. Pregel : a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 135–146. ACM, 2010. Cité pages 150, 181, and 182.

[80] Norbert Martínez-Bazan, M. Ángel Águila Lorente, Victor Muntés-Mulero, David Dominguez-Sal, Sergio Gómez-Villamor, and Josep-L. Larriba-Pey. Efficient Graph Management Based On Bitmap Indices. In *Proc. of the 16th Int. Database Eng. & Appl. Symp.*, IDEAS '12, pages 110–119, 2012. Cité page 179.

[81] M. McCord and M. Chuah. Spam detection on twitter using traditional classifiers. In *Proceedings of the 8th international conference on Autonomic and trusted computing*, ATC'11, pages 175–186, 2011. Cité page 92.

[82] Bruce C. McKinney, Lynne Kelly, and Robert L. Duran. Narcissism or openness ? : College students' use of facebook and twitter. *Communication Research Reports*, 29(2) :108–118, 2012. Cité page 13.

[83] Simply Measured. Research Shows 94% of Top Brands Tweet at Least Once per Day, 2014. http://simplymeasured.com/blog/2014/11/19/research-shows-94-of-top-brands-tweet-at-least-once-per-day/. Cité pages 11 and 12.

[84] 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. Cité pages 77 and 83.

[85] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42. ACM, 2007. Cité page 9.

[86] Moderateur. Instagram passe les 300 millions d'utilisateurs actifs par mois, 12 2014. http ://www.blogdumoderateur.com/instagram-300-millions-utilisateurs/. Cité page 10.

[87] Moderateur. Chiffres rÉseaux sociaux – 2015, 01 2015. http ://www.blogdumoderateur.com/chiffres-reseaux-sociaux/. Cité page 10.

[88] Robert J. Mokken. Cliques, clubs and clans. *Quality & Quantity*, 13(2) :161–173, April 1979. Cité page 114.

[89] Seth A Myers, Aneesh Sharma, Pankaj Gupta, and Jimmy Lin. Information network or social network ? : The structure of the twitter follow graph. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 493–498. International World Wide Web Conferences Steering Committee, 2014. Cité page 12.

[90] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23) :8577–8582, 2006. Cité page 50.

[91] Mark EJ Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2) :404–409, 2001. Cité page 117.

[92] 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. Cité pages 50 and 115.

[93] Bharath Pattabiraman, Md Mostofa Ali Patwary, Assefaw H Gebremedhin, Wei-keng Liao, and Alok Choudhary. Fast algorithms for the maximum clique problem on massive sparse graphs. In *Algorithms and Models for the Web Graph*, pages 156–169. Springer, 2013. Cité page 115.

[94] 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. *Journal of Machine Learning Research*, 12 :2825–2830, 2011. Cité pages 88 and 90.

[95] Dan Pelleg, Andrew W Moore, et al. X-means : Extending k-means with efficient estimation of the number of clusters. In *ICML*, pages 727–734, 2000. Cité page 130.

[96] PETER POLLNER, GERGELY PALLA, and TAMAS VICSEK. Parallel clustering with cfinder. *Parallel Processing Letters*, 22(01), 2012. Cité page 50.

[97] Huffington Post. Twitter : We now have over 200 million accounts, 2011. http://www.huffingtonpost.com/2011/04/28/twitter-number-of-users_n_855177.html. Cité page 12.

[98] U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3), 2007. Cité page 50.

[99] Simon Rodgers. Twitter blog, august 2013. https://blog.twitter.com/2013/behind-the-numbers-how-to-understand-big-moments-on-twitter. Cité page 12.

[100] Andrew Rosenberg and Julia Hirschberg. V-measure : A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning(EMNLP-CoNLL)*, pages 410–420, 2007. Cité page 54.

[101] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4) :1118–1123, 2008. Cité page 50.

[102] Yousef Saad. *Iterative methods for sparse linear systems*. Siam, 2003. Cité page 180.

[103] A. Sameh. A Twitter analytic tool to measure opinion, influence and trust. *Journal of Industrial and Intelligent Information*, 1(1) :37–45, 2013. Cité page 77.

[104] M. Seifi. *Coeurs stables de communautés dans les graphes de terrain*. PAF, 2012. Cité pages 112, 117, 118, 119, 121, and 150.

[105] Massoud Seifi, Ivan Junier, Jean-Baptiste Rouquier, Svilen Iskrov, and Jean-Loup Guillaume. Stable community cores in complex networks. In *Complex Networks*, pages 87–98. Springer, 2013. Cité page 116.

[106] Catherine Shu. Tech Crunch, march 2014. Cité page 79.

[107] George Gaylord Simpson. Mammals and the nature of continents. *Am. J. of Science*, (241) :1–41, 1943. Cité page 23.

[108] Meredith M Skeels and Jonathan Grudin. When social networks cross boundaries : a case study of workplace use of facebook and linkedin. In *Proceedings of the ACM 2009 international conference on Supporting group work*, pages 95–104. ACM, 2009. Cité page 9.

[109] Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3 :583–617, 2002. Cité page 54.

[110] B. Suh, Lichan Hong, P. Pirolli, and Ed H. Chi. Want to be retweeted ? large scale analytics on factors impacting retweet in twitter network. In *Social Computing (SocialCom), 2010 IEEE Second International Conference on*, pages 177–184, 2010. Cité page 10.

[111] Telegraph. The Telegraph : Twitter in numbers, 2013. http://www.telegraph.co.uk/technology/twitter/9945505/Twitter-in-numbers.html. Cité page 12.

[112] 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. Cité page 77.

[113] La Tribune. Youtube passe d'un à deux milliards de vidéos vues par...jour !, 05 2010. http ://www.latribune.fr/technos-medias/internet/20100517trib000509822/youtube-passe-d-un-a-deux-milliards-de-videos-vues-parjour.html. Cité page 9.

[114] TweetGrader. Review your Twitter account | free Twitter analyzer | Twitter grader - http ://twittergrader.mokumax.com/. http://twittergrader.mokumax.com/. Cité page 77.

[115] Twitalyzer. Twitalyzer, serious analytics for social business. http://www.twitalyzer.com - As of September 28, 2013 Twitalyzer has decided to no longer sell new subscriptions. Cité pages 77 and 78.

[116] Twitter. Règles et bonnes pratiques d'abonnement. https ://support.twitter.com/articles/95609-regles-et-bonnes-pratiques-d-abonnement. Cité page 15.

[117] Twitter. Règles de twitter, 2010. https ://support.twitter.com/articles/75576-regles-de-twitter. Cité page 15.

[118] Twitter. Twitter support center : Why can't I follow people ?, 2013. https://support.twitter.com/groups/52-connect/topics/213-following/articles/66885-why-can-t-i-follow-people. Cité pages 15, 35, and 42.

[119] Twitter. Twitter API V 1.1 overview, 2013. https://dev.twitter.com/docs/api/1.1/overview. Cité pages 26, 40, and 85.

[120] Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the facebook social graph. *CoRR*, abs/1111.4503, 2011. Cité page 9.

[121] Virginia Vassilevska. Efficient algorithms for clique problems. *Information Processing Letters*, 109(4) :254–257, 2009. Cité page 115.

[122] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P. Gummadi. On the evolution of user interaction in Facebook. In *Proc. Workshop on Online Social Networks*, pages 37–42, 2009. Cité page 119.

[123] Alex Hai Wang. Don't follow me : Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, pages 1–10, 2010. Cité page 12.

[124] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684) :440–442, 1998. Cité page 38.

[125] 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. Cité page 77.

[126] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank : finding topic-sensitive influential twitterers. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 261–270, 2010. Cité page 13.

[127] Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, pages 1–33, 2013. Cité pages 9 and 50.

[128] W.W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33 :452–473, 1977. Cité page 49.

[129] Ying Zhao and George Karypis. Criterion functions for document clustering : Experiments and analysis. Technical report, 2002. Cité page 54.

Table des figures

Liste des tableaux

ANNEXE

A

Figures

Nous présentons dans cette annexe deux figures liées à la chronologie de l'émergence des réseaux sociaux A.1 et à la décomposition hiérarchique obtenue en utilisant la méthode des cœurs de communautés A.2.

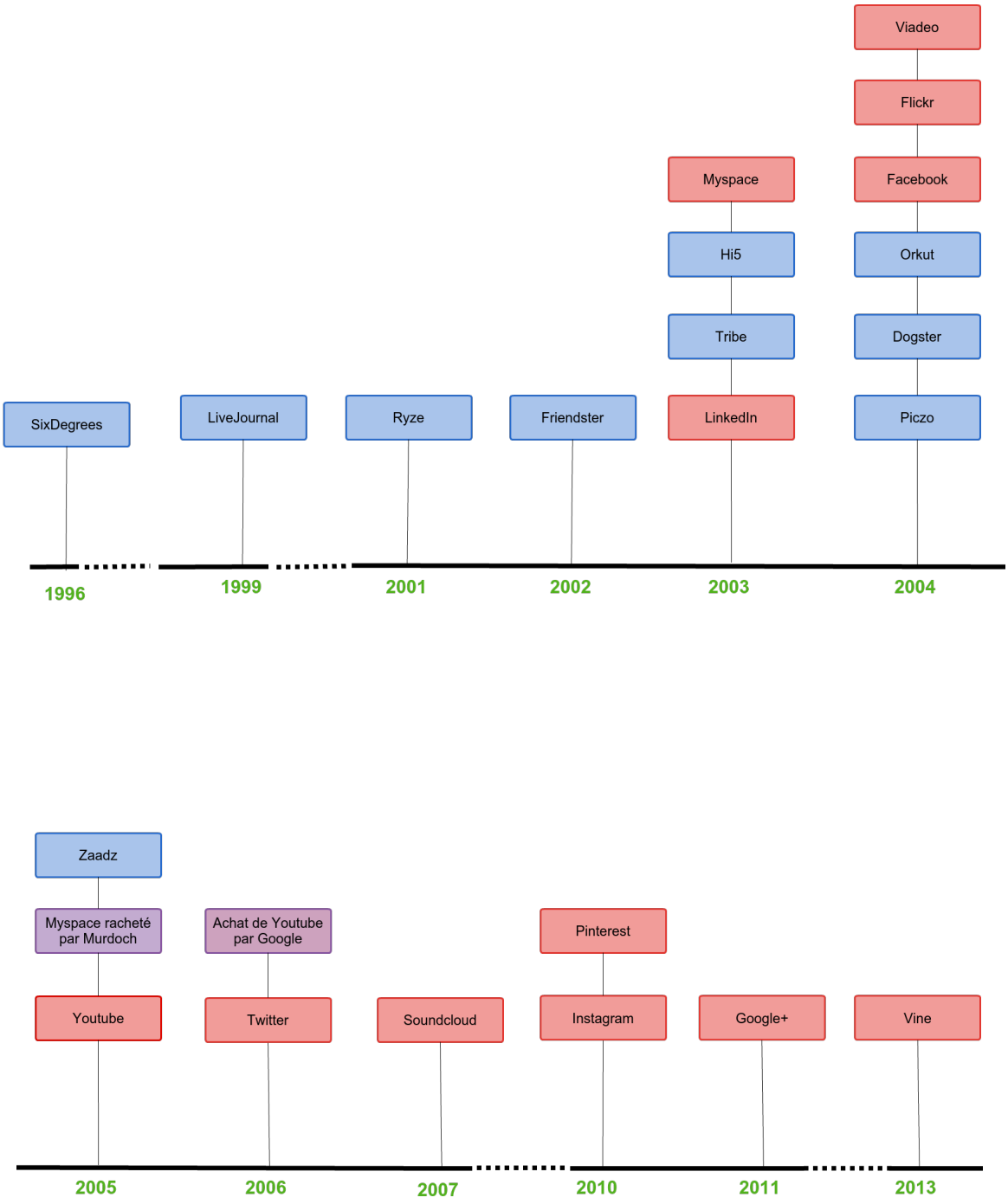

FIGURE A.1 – Chronologie d'apparition des réseaux sociaux les plus célèbres.

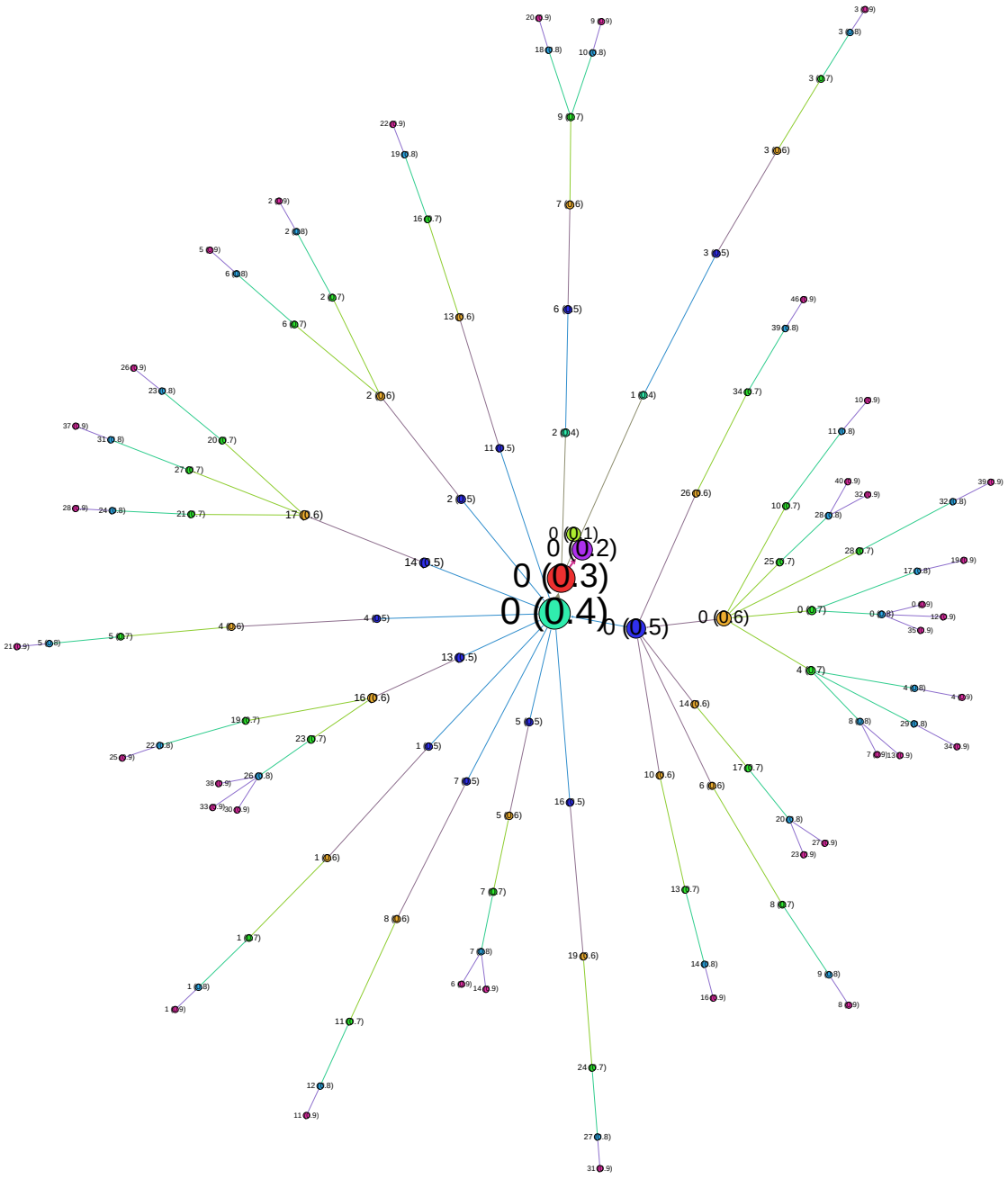

FIGURE A.2 – Un arbre de décomposition hiérarchique de cœurs détectés sur le réseau *Hamsterster*. Entre parenthèses, on retrouve le paramètre α utilisé pour détecter le cœur. Le chiffre à côté est un identifiant.

ANNEXE

B

Annexe B : Modularité orientée

Dans cette Annexe, nous présentons en détails l'algorithme de **Louvain orienté** que nous introduisons dans le Chapitre 2. Nous commençons par définir la *modularité* ainsi que la *modularité orienté* dont nous donnons motivons l'intérêt : soient deux sommets u et v de faible dégré entrant (resp. degré entrant élevé), un arc de v vers u est moins probable qu'un arc de u vers v. Nous montrons théoriquement que la modularité orientée modélise bien cette intuition. Par ailleurs, nous présentons des comparatifs exhaustifs entre les résultats retournés par les algorithmes de **Louvain** [10], **Louvain orienté** et **OSLOM** [67]. Nous montrons ainsi que les performances de **Louvain orienté** sont meilleures que celles de **Louvain** [10] sur des réseaux orientés. En revanche, **OSLOM** [67] retourne des partitions des meilleure qualité que **Louvain orienté**. Suite à ce constat, nous mesurons les temps d'exécution de **Louvain orienté** et **OSLOM** [67], et montrons qu'**OSLOM** [67] est incapable de produire des résultats sur des réseaux de très grandes taille.

Directed Louvain : maximizing modularity in directed networks

Nicolas Dugué Anthony Perez

August 27, 2015

Abstract

In this paper we consider the community detection problem from two different perspectives. We first want to be able to compute communities for *large directed networks*, containing *million* vertices and *billion* arcs. Moreover, in a large number of applications, the graphs modelizing such networks are *directed*. Nevertheless, one is often forced to forget the direction between the connections, either for the sake of simplicity or because no other options are available. This is in particular the case on large networks, since there are only a few scalable algorithms at the time. We thus turn our attention to one of the most famous scalable algorithms, namely Louvain's algorithm [3], based on modularity maximization. We modify Louvain's algorithm to handle directed networks based on the notion of directed modularity defined by Leicht and Newman [13], and provide an empirical and theoretical study to show that one should prefer directed modularity. To illustrate this fact, we use the LFR benchmarks by Lancichinetti and Fortunato [8] to design an evaluation benchmark of directed graphs with community structure. We also give some examples and insights on the situations where one should *really* consider direction when maximizing modularity. Finally, for the sake of completeness, we compare the results obtained with Oslom [12], one of the best algorithms to detect communities in directed networks. While the results obtained with such an algorithm are by far better on the LFR benchmarks, we emphasize that it is still not well-suited to deal with very large networks.

1 Introduction

In various domains such as social networks or bioinformatics, being able to detect communities efficiently constitutes a very important research interest [6]. In most cases, the underlying graphs representing data are *directed*. This happens for instance when considering some social network graphs, where relations between two users u and v can be represented by stating that u has an influence *over* v rather than simply saying that they both interact. It thus seems quite obvious to consider direction when detecting communities, and several algorithms were proposed in this sense, such as Oslom [12] or InfoMap [19, 20]. In this article, we are interested in detecting communities in very large networks such as the Twitter graph [4], which contains more than 50 *millions* vertices and almost 2 *billions* arcs. As we shall see Section 5, Oslom [12] fail to efficiently produce communities when considering such networks [4], especially if one wants to use only a few computer ressources. Due to this fact, a common solution is to simply *forget* direction when detecting communities in really large network and to run Louvain's algorithm [3] (which is extremely well-suited for large networks). To the best of our knowledge, there is no version of this algorithm maximizing *directed modularity* [13]. This fact can also be seen in a survey comparing algorithms for community detection, where Lancichinetti and Fortunato [10] did not even consider Louvain's

algorithm for their directed networks analysis. Instead, they used simulated annealing for modularity optimization [6] even if they confirmed on undirected networks that Louvain's algorithm performs better and runs a lot faster.

Our results. In this work, we give some insights about the importance of direction while detecting communities. To that aim, we consider Louvain's algorithm [3], which is implemented for non-directed graphs *only*. By modifying the existing source code [2], we manage to deal with directed graphs, following the notion of directed modularity introduced by Leicht and Newman [13] (Section 2). We then generated a benchmark of directed graphs using the framework provided by Fortunato et al. [8], and computed communities on such graphs using both versions of the Louvain's algorithm[1]. Our results show strong evidence that direction is important when detecting communities (Section 5). Finally, we also compare these results to communities obtained by a recent community detection algorithm called OSLOM [12], both from the semantic and complexity viewpoints (Section 5). We emphasize that OSLOM [12] cannot deal with large graphs such as the Twitter graph [4] (billions of edges), while Louvain's algorithm produces results in a couple of hours.

2 Detecting community in large (directed) networks

Modularity. A classic way of detecting communities is to find a partition of the vertex set that maximizes an optimization function. One of the most famous optimization function is called *modularity* [16]. This function provides a way to value the existence of an edge between two vertices of an undirected network by comparing it with the probability of having such an edge in a random model following the same degree distribution than the original network. For instance, an edge between two vertices of large degree is not surprising, and thus does not contribute much to the modularity of a given partition, whereas an edge between two vertices of small degree is more surprising. Formally, the modularity Q of a partition $\mathcal{C}$ of an undirected graph $G = (V, E)$ is defined as follows :

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{d_i d_j}{2m} \right] \delta(c_i, c_j)$$

where m stands for the number of edges of G, A_{ij} represents the weight of the edge between i and j (set to 0 if such an edge does not exist), d_i is the degree of vertex i (i.e. the number of neighbors of i), c_i is the community to which vertex i belongs and the δ-function $\delta(u, v)$ is defined as 1 if $u = v$, and 0 otherwise.

Leicht and Newman [13] adapted the notion of modularity for directed graphs, motivated by the following observation: if two vertices u and v have small in-degree/large out-degree and small out-degree/large in-degree, then having an arc from v to u should be considered more surprising than having an arc from u to v. Taking this into account, the definition for directed modularity of a partition of a directed network can be easily formulated:

[1] Louvain's algorithm is usually non-deterministic, but in order to obtain consistent results, we *always* consider the vertices in the same order.

$$Q_d = \frac{1}{m} \sum_{i,j} \left[A_{ij} - \frac{d_i^{in} d_j^{out}}{m} \right] \delta(c_i, c_j)$$

where A_{ij} now represents the existence of an *arc* between i and j and d_i^{in} (resp. d_j^{out}) stands for the *in-degree* (resp. *out-degree*) of i.

Louvain's algorithm. We now briefly describe the behavior of Louvain's algorithm to maximize modularity. The algorithm is the same for both the classic and directed versions of modularity. It relies on a greedy procedure: starting from any partition of the vertices (usually the partition into singletons), the algorithm tries to increase the value of modularity by moving vertices from their community to any other neighbor one. In other words, the algorithm computes the *gain* of modularity obtained by adding vertex i to community C as follows (for the undirected case):

$$\begin{aligned} \Delta_Q &= \left[\frac{\sum_{in} + d_i^C}{2m} - \left(\frac{\sum_{tot} + d_i}{2m} \right)^2 \right] - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{d_i}{2m} \right)^2 \right] \\ &= \frac{d_i^C}{2m} - \frac{\sum_{tot} \cdot d_i}{2m^2} \end{aligned}$$

where d_i^C denotes the degree of node i in community C, $\sum_{in}$ the number of edges contained in community C and $\sum_{tot}$ the total number of edges incident to community C. Actually, the first formula is the one as described in [3], but one can see that it reduces to the second one. The algorithm does a similar calculation to compute the *gain* obtained by *removing* vertex i from its own community C_i in a first place. The algorithm carries on as long as it exists a move that improves the value of modularity.

The behavior of the algorithm is *exactly* the same in the directed case, the main difference lying in the calculation for the gain of modularity obtained by adding vertex i to community C, which can now be done using the following:

$$\Delta_{Q_d} = \frac{d_i^C}{m} - \left[\frac{d_i^{out} \cdot \sum_{tot}^{in} + d_i^{in} \cdot \sum_{tot}^{out}}{m^2} \right]$$

where $\sum_{tot}^{in}$ (resp. $\sum_{tot}^{out}$) denotes the number of *in-going* (resp. *out-going*) arcs incident to community C.

3 Theoretical comparison between undirected and directed modularity

Our goal is to validate the observation made by Leicht and Newman [13] by showing what happens if one uses the modularity Q [16] on a directed graph instead of using its directed version Q_d [13]. To that aim, we use a straightforward case study where we consider two subgraphs C_1 and C_2 which both are communities of a directed network (see Figure 1). We are basically studying when merging these communities lead to a value increase of both modularities. To calculate the undirected modularity on a network which is usually directed, we have to ignore the links direction.

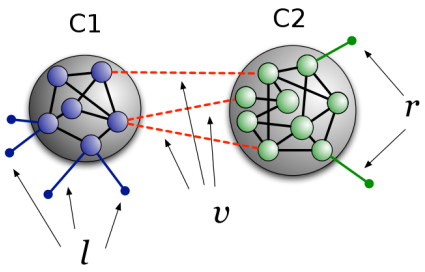

Figure 1: Figure extracted from the article of Lancichinetti and Fortunato [11].

Thus, if we are processing a directed network $D = (V, A)$ where $(u, v) \vee (v, u) \in A$, then $(u, v) \in E$ in the undirected version $G = (V, E)$. We use $Q^{C_1 \setminus C_2}$ (resp. $Q_d^{C_1 \setminus C_2}$) to refer to the undirected (resp. directed) modularity value of the network with C_1 and C_2 distinct communities. In the same way, we use $Q^{C_1 \cup C_2}$ (resp. $Q_d^{C_1 \cup C_2}$) to refer to the modularity value of the network where C_1 and C_2 are part of the same community. We name $A_{1,2}$ arcs between communities C_1 and C_2, and $E_{1,2}$ the corresponding edges in the undirected network. Considering the undirected case, $|E_{1,2}| = |A_{1,2}|$ if $\forall (u, v) \in A_{1,2}$ then $(v, u) \notin A_{1,2}$. We also have that $|E_{1,2}| = \frac{1}{2} \cdot |A_{1,2}|$ if $\forall (u, v) \in A_{1,2}$ then $(v, u) \in A_{1,2}$. Thus, $|E_{1,2}| \leqslant |A_{1,2}| \leqslant 2 \cdot |E_{1,2}|$.

3.1 Undirected case

When C_1 and C_2 are considered as part of the same community, $E_{1,2}$ links contribute to increase modularity value, as shown in bold in the following formula.

$$Q^{C_1 \cup C_2} = \left(\frac{d_{C_1}^{int}}{m} + \frac{d_{C_2}^{int}}{m} + \frac{\mathbf{|E_{1,2}|}}{\mathbf{m}} \right) - \left(\sum_{i,j \in C_1} \frac{d_i d_j}{4m^2} + \sum_{i,j \in C_2} \frac{d_i d_j}{4m^2} + \sum_{\mathbf{i \in C_1, j \in C_2}} \frac{\mathbf{d_i d_j}}{\mathbf{2m^2}} \right)$$

When C_1 and C_2 are splitted in two different communities, both the terms in bold before disappear.

$$Q^{C_1 \setminus C_2} = \left(\frac{d_{C_1}^{int}}{m} + \frac{d_{C_2}^{int}}{m} \right) - \left(\sum_{i,j \in C_1} \frac{d_i d_j}{4m^2} + \sum_{i,j \in C_2} \frac{d_i d_j}{4m^2} \right)$$

Thus, if summing these bold terms results in a positive number, C_1 and C_2 are merged. At the contrary, if the sum is negative, C_1 and C_2 are considered as two distinct communities. Therefore, studying when these communities are merged or not consists in studying the sum of these terms as follows.

$$\begin{aligned} \delta_Q &= Q^{C_1 \cup C_2} - Q^{C_1 \setminus C_2} \\ &= \frac{|E_{1,2}|}{m} - \sum_{i \in C_1, j \in C_2} \frac{d_i d_j}{2m^2} \\ &= \frac{1}{m} \Big(|E_{1,2}| - \sum_{i \in C_1, j \in C_2} \frac{d_i d_j}{2m} \Big) \end{aligned}$$

Hence:

$$\delta_Q > 0 \quad \Leftrightarrow \quad |E_{1,2}| > \sum_{i \in C_1, j \in C_2} \frac{d_i d_j}{2m} \tag{1}$$

3.2 Directed case

In the directed case, we obtain a quite similar result. Indeed, when C_1 and C_2 are considered as being part of the same community, we obtain:

$$\begin{aligned} \delta_{Q_d} &= Q_d^{C_1 \cup C_2} - Q_d^{C_1 \setminus C_2} \\ &= \frac{|A_{1,2}|}{2m} - \sum_{i \in C_1, j \in C_2} \frac{d_i^{in} d_j^{out}}{4m^2} - \sum_{i \in C_1, j \in C_2} \frac{d_i^{out} d_j^{in}}{4m^2} \end{aligned}$$

Hence:

$$\delta_{Q_d} > 0 <=> |A_{1,2}| > \sum_{i \in C_1, j \in C_2} \left(\frac{d_i^{in} d_j^{out}}{2m} + \frac{d_i^{out} d_j^{in}}{2m} \right)$$

3.3 Comparison

To compare the choices made by both modularities, we replace the vertex degree of Equation 1 by its in- and out-going counterparts.

$$d_i d_j = (d_i^{in} + d_i^{out})(d_j^{in} + d_j^{out})$$

We thus obtain the following equivalence :

$$|E_{1,2}| > \sum_{i \in C_1, j \in C_2} \left(\frac{d_i^{in} d_j^{out}}{2m} + \frac{d_i^{out} d_j^{in}}{2m} \right) + \sum_{i \in C_1, j \in C_2} \left(\frac{d_i^{in} d_j^{in}}{2m} + \frac{d_i^{out} d_j^{out}}{2m} \right)$$

Let us define the following terms :

$$S = \sum_{i \in C_1, j \in C_2} \left(\frac{d_i^{in} d_j^{out}}{2m} + \frac{d_i^{out} d_j^{in}}{2m} \right)$$

$$T = \sum_{i \in C_1, j \in C_2} \left(\frac{d_i^{in} d_j^{in}}{2m} + \frac{d_i^{out} d_j^{out}}{2m} \right)$$

Thus, in the undirected case, C_1 and C_2 are merged when $|E_{1,2}| > S + T$ while in the directed case, the fusion is done when $|A_{1,2}| > S$, T being absent from the equation.

The term S confirms the observation made by Leicht and Newman [13]. Furthermore, we can see that T is not relevant at all. Multiplying the incoming degrees in one side and the outgoing degrees in the other side does not allow to estimate links probability to exist between communities in a random network. This may explain the better results obtained with the Louvain algorithm implementing the directed modularity.

4 Analysis of the differences

We now study the conditions that make differences arise between the two versions of Louvain's algorithms. We first give some intuition on the configurations that can lead to two different moves in the algorithm, and then provide several examples where the difference is significant.

Sufficient conditions to have a difference. To complete the previous observations, we give some conditions that will influence community detection between the two methods. Recall that the *gain of modularity* can be easily computed (in both cases) using the following:

$$\begin{aligned} \Delta_Q &\sim d_i^C - \frac{\sum_{tot} \cdot d_i}{2m} \\ \Delta_{Q_d} &= d_i^C - \left[\frac{d_i^{out} \cdot \sum_{tot}^{in} + d_i^{in} \cdot \sum_{tot}^{out}}{m} \right] \end{aligned}$$

We thus have to study the behavior of the terms $\frac{\sum_{tot} \cdot d_i}{m}$ and $\frac{d_i^{out} \cdot \sum_{tot}^{in} + d_i^{in} \cdot \sum_{tot}^{out}}{m}$ for a given vertex i and a given community C. In particular, we want to express the conditions when the first one is positive and the second one negative, or vice-versa. Recall that, in the first case, the classic Louvain's algorithm will not consider adding vertex i to community C to increase modularity, while the directed version will do so.

Cases that make a difference. We first provide some simple examples when the classic Louvain's algorithm fails at detecting communities, whereas the algorithm maximizing directed modularity finds a perfect match with the ground truth communities.

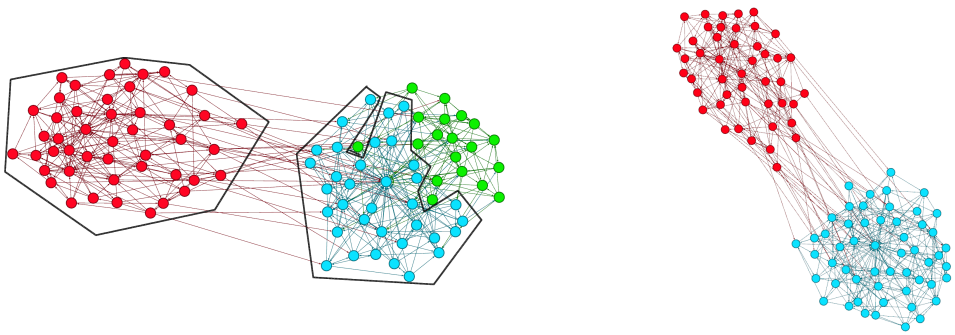

Figure 2: On the left, the three communities obtained by maximizing standard modularity are represented. On the right, the ones obtained using directed modularity.

Consider the graph represented Figure 2, which contains 100 vertices and 2 communities. When maximizing directed modularity, Louvain's algorithm succeeds in retrieving the communities whereas the classic modularity maximization fails to merge two communities. The explanation for this situation follows from our previous arguments. Indeed, the graph contains vertices with unbalanced in and out-degrees, who thus influence the greedy method of Louvain's algorithms. Such a situation can also be observed on larger graphs[2] (see Figure 3).

[2]For the sake of visiblity we do not consider larger graphs, but mention that similar situations happen also.

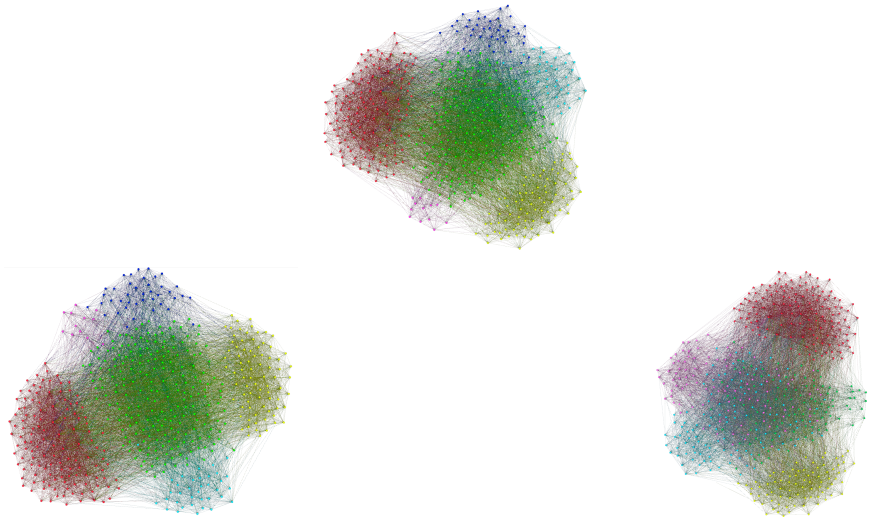

Figure 3: On top, the groundtruth communities. On the bottom left, the communities found by the directed version of the algorithm and on the bottom right the ones provided by the classic one.

5 Experimental results

We now present empirically the differences that arise between classic modularity maximization and the directed one. To that aim, we evaluate the results of both the modulairies over the so-called directed LFR benchmarks [10].We consider partitions (that is non-overlapping communities).

5.1 The LFR benchmarks

To validate the efficiency of Louvain algorithm adapted to directed graphs, we use benchmarks introduced by Lancichinetti and Fortunato [9]. These benchmarks allow to test community detection algorithms on directed graphs, and are designed in order to be as realistic as possible with respect to real networks. It is indeed possible to set important features such as the power-law distribution of the degrees of the nodes or of the communities sizes, as well as the maximum and average degrees of nodes in the graphs. Another major feaure introduced in these benchmarks is the mixing parameter. The mixing parameter allows to create graphs with communities more or less well-defined. A low mixing parameter indicates communities well-defined, and hence easy to detect. Reciprocally, a high mixing parameter allows to create graphs with communities which will be hard to detect.

5.2 Measures

To compare the results obtained by the community detection methods, we use three evaluation measures. The results of the community detection algorithms are thus compared with the communities defined by the benchmarks we use. In the following, we use *clustering* to denote the community sets obtained by the algorithms used. The term *cluster* is thus one community of these sets. We use *community* to talk about the groundtruth communities established by the benchmark.

The first measure, called *V-Measure* [18] is made of two criteria: *homogeneity* and *completeness*. This may be compared to the *F-measure* based on precision and recall measures. A clustering maximizes the homogeneity if for each cluster, we find only elements of a same community. Symetrically, completeness is maximized when for each community, all elements of a same community are in a single cluster. By computing the harmonic mean of these two values, we obtain the so-called *V-measure*. The second one is the NMI [21] for *Normalized Mutual Information*. Built upon concepts from information theory, this measure is commonly used to compare clusterings. Roughly speaking, this measure defines how much knowing one of two clusterings reduces uncertainty about the other. Thus, the higher the NMI, the more information the two clusterings share. We use the normalization introduced by Strehl and Gosh [21] defined as follows.

definition 1 (NMI [21]) *Let $\mathcal{U}$ and $\mathcal{V}$ be two clusterings. Then the* Normalized Mmutual Information *is defined as a function of the* mutual information *I and the* conditional entropy *H:*

$$NMI(\mathcal{U},\mathcal{V}) = \frac{I(\mathcal{U},\mathcal{V})}{\sqrt{H(\mathcal{U}) \cdot H(\mathcal{V})}}$$

Finally, to compute the Purity [24], we assign each cluster to the community which nodes are more frequent in the cluster. Then, by summing all well-classified nodes for each of these clusters and dividing it by the number of vertices, we obtain the accuracy of our clustering.

5.3 Classic LFR benchmarks

We begin this section by giving some results obtained by generating LFR benchmarks using parameters described by Lancichinetti and Fortunato [10]. Those parameters consider two different cases, namely graphs having *small* and *big* community sizes. In the first case, the communities are set to contain between 10 and 50 vertices, while they are required to contain between 20 and 100 vertices in the latter one. In such graphs, the average degree is set to 20 and the maximum degree is set to 50. We consider graphs having respectively 1000 and 5000 nodes, and make the mixing parameter go from 0.1 (*i.e.* well-defined communities) to 0.6. Finally, we set the power law distribution to 2 in all cases. We first give a general picture of the results we obtained w.r.t. to the number of vertices and the size of the communities (Table 1).

n	$minc$	$maxc$	# graphs	$\geqslant$	$\geqslant^{+0.05}$	$\geqslant^{+0.1}$	$\geqslant^{+0.2}$
1000	10	50	900	839	83	54	5
1000	20	100	900	776	70	27	3
5000	10	50	900	804	0	0	0
5000	20	100	900	785	54	31	3

Table 1: Proportion of graphs where the NMI of the classic modularity (nmi_o) is greater than the one of the directed modularity (nmi_d) by a given percentage.

Louvain's algorithm maximizing *directed* modularity is better in almost 75% of the cases. Moreover, we would like to mention that if the improvement is rather low on average, there are some interesting cases where the improvement is drastic.

We then compare the outputs of both version of Louvain's algorithms according to the aforementioned quality measures. We conducted such an experimentation by considering different networks sizes, and also by modifying the *mixing parameter*. As one can observe Table 2, the directed version (which corresponds to the bottom table) is better in most cases.

n	mu	NMI	V-measure	Homogeneity	Completeness	Purity
1000	0.1	0.987	0.987	1.000	0.975	1.000
1000	0.6	0.965	0.964	0.999	0.932	0.999
5000	0.1	0.966	0.965	1.000	0.934	1.000
5000	0.6	0.909	0.905	0.999	0.828	0.999

n	mu	NMI	V-measure	Homogeneity	Completeness	Purity
1000	0.1	**0.995**	**0.995**	1.000	**0.990**	1.000
1000	0.6	**0.978**	**0.978**	**1.000**	**0.958**	**1.000**
5000	0.1	**0.978**	**0.978**	**1.000**	**0.957**	1.000
5000	0.6	**0.920**	**0.917**	0.999	**0.848**	0.999

Table 2: Results obtained on the classic LFR benchmarks with the classic and the directed versions of Louvain algorithm. Each mesure indicates the average taken over 100 graphs with the indicated parameters.

5.4 Generated LFR benchmarks

In this Section, we present similar observations on a new set of benchmarks that we generated for this purpose. Recall that the average and maximum degree are respectively fixed to 20 and 50 in the classic LFR benchmarks. This seems quite unrealistic when trying to simulate social networks: on Table 3 we observe in several well-known complex networks that the average degree is in general much lower, while the maximum degree is much higher. Hence, it seems quite restrictive to impose a maximum degree of 50, when it is really common for vertices of such networks to have a degree close to $\frac{n}{3}$.

Network	Nodes	Edges	Avg degree	Max degree	Power-law
Zachary karate club [22]	34	78	4.59	17	2.16
Openflight [17]	2,939	30,501	20.76	473	1.79
Arxiv Astro-ph [14]	18,771	198,050	21.10	504	2.60
Internet AS [23]	34,761	171,403	9.86	5,305	1.86
SlashDot [5]	51,083	140,778	5.51	3,357	1.91
Flickr [15]	2,302,925	33,140,018	28.78	34,174	2.02
DBPedia [1]	2,152,642	7,494,124	6.96	329,714	2.15

Table 3: Basic properties of several physical, social or reference based network from literature.

To complete our analysis, we generated about 40000 graphs with different parameters closely related to those observed in real networks. In particular, we infered the average and maximum

degrees w.r.t. the power law and mixing parameter. For every combination of such parameters, we generated 100 different graphs and ran a statistical analysis. We first give a general picture of our observations.

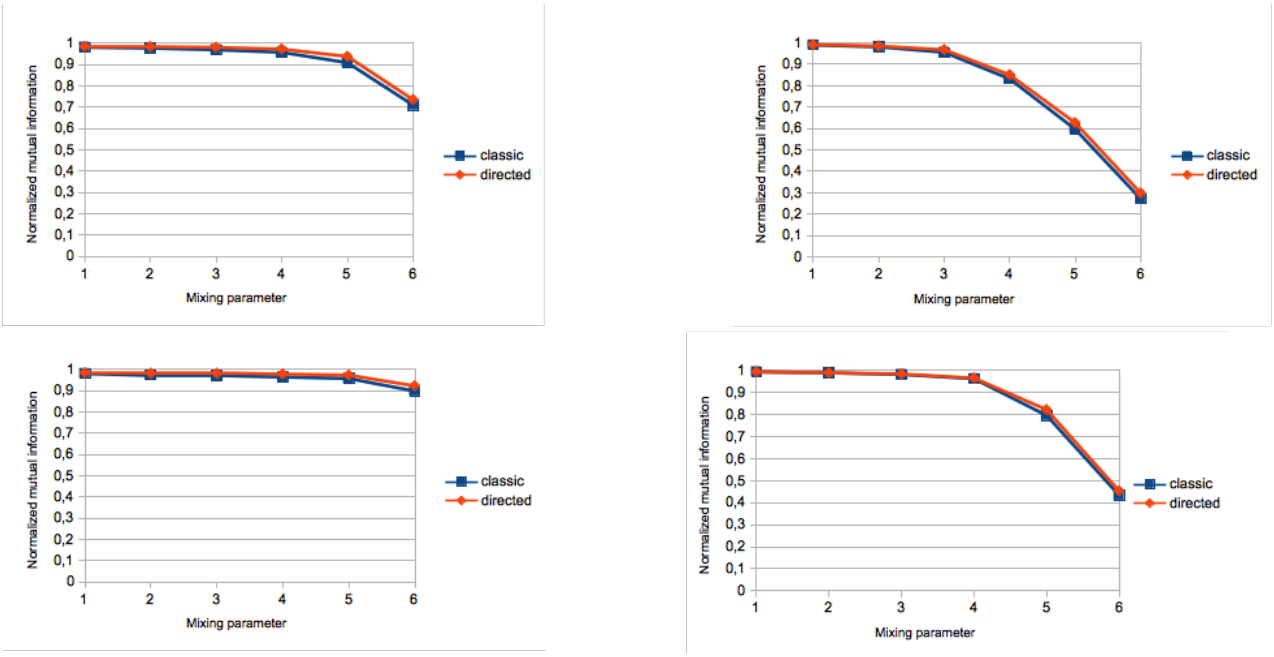

Figure 4: The values of the normalized mutual information are represented on the Y axis, while the mixing parameter is on the X axis.

n	# graphs	$\geqslant$	$\geqslant^{+0.05}$	$\geqslant^{+0.1}$	$\geqslant^{+0.25}$
100	9900	6648	2331	966	209
500	9900	6833	1193	356	33
1000	9900	6789	926	263	22
5000	9900	6674	770	211	31

Table 4: Proportion of graphs where nmi_o is greater than nmi_d by a given percentage.

It arises from Table 4 that in 70% of the cases, directed modularity provides better results than the classic Louvain's algorithm. We would like to mention that very similar results can be observed when considering other similarity measures such as F-measure, V-measure, purity, completeness, homogeneity. To be consistent with the results presented in [10], we have a closer look on the measures for 1000 and 5000 nodes, respectively. The results presented Figure 4 were obtained by taking the average of the measures over 100 different graphs with the same parameters.

We would like to mention that Louvain's algorithm seems to be particularly well-suited to maximize directed modularity, since the results obtained seem to be really better than the ones presented by Fortunato et al. [], obtained using simulated annealing for modularity optimization [6].

5.5 A comparative analysis with OSLOM [12]

To conclude this part of our work, we would like to compare Louvain's algorithm optimizing directed modularity with another algorithm used to detect communities in directed networks, namely OSLOM [12]. We are particularly interested in the *time complexity* needed to run such algorithms. We want to emphasize that such an algorithm is really not well-suited for handling large (directed) networks.

Results. We now present the results obtained by running OSLOM [12] on the classic LFR benchmarks. We focus on both the accurency and the time complexity needed to obtain such results.

n	mu	NMI	V-measure	Homogeneity	Completeness	Purity
1000	0.1	0.999	0.999	0.999	0.999	0.999
1000	0.6	0.999	0.999	0.999	0.999	0.999
5000	0.1	0.994	0.999	0.999	0.999	0.999
5000	0.6	0.999	0.999	0.999	0.999	0.999

Table 5: Results obtained on the classic LFR benchmarks. Each mesure indicates the average taken over 100 graphs with the indicated parameters.

As one can see on Table 5, OSLOM [12] is drastically better than Louvain's algorithm on the classic LFR benchmarks. This is not a surprising result, since OSLOM [12] seems to provide really good results [10]. However, as we shall see in the remaining of the paper, OSLOM [12] cannot provide results in a reasonnable amount of time for networks with relatively small size.

Time complexity. We now focus on the time complexity needed to obtain results when using OSLOM or Louvain's algorithm with directed modularity. We conduct our analysis on a set of real networks extracted from Konect Database [7] (see Table 6).

Name	**Vertices**	**Arcs**	**Louvain**	OSLOM
E-COLI	99	212	~ 0 seconds	1.8 seconds
BIO-YEAST	688	1079	~ 0 seconds	~ 35 seconds
SPANISH-BOOK	12643	57772	~ 0.2 seconds	~ 27 minutes
WORD-ASSOCIATION	10617	72168	~ 0.26 seconds	~ 30 minutes
EDINBURG	23219	325624	~ 1 seconds	> 10 hours

Table 6: Directed networks used for time complexity analysis.

As mentioned previously, we do not consider here the quality of the output (OSLOM performs better than Louvain's algorithm), but we only focus on the time needed to obtain such an output. For networks with 10000 vertices, OSLOM [12] already takes a significant amount of time[3] to compute the results. On the largest graph we consider, the classic configuration of OSLOM [12] fails at producing any output even after hours of computation.

Thus, if one should clearly prefer using OSLOM [12] on *small networks*, its use is absolutely impossible for networks as large as the ones that really often arise in the literature. We hope that our study will enlight that the version of Louvain's algorithm that maximizes directed modularity should be considered as more reliable to deal with such networks.

References

[1] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: a nucleus for a web of open data. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ISWC'07/ASWC'07, pages 722–735, 2007.

[2] Vincent Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Louvain method: Finding communities in large networks. https://sites.google.com/site/findcommunities/.

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

[4] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and Krishna P. Gummadi. Measuring User Influence in Twitter: The Million Follower Fallacy. In *ICWSM '10: Proc. of int. AAAI Conference on Weblogs and Social*, 2010.

[5] Vicenç Gómez, Andreas Kaltenbrunner, and Vicente López. Statistical analysis of the social network and discussion threads in slashdot. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 645–654, 2008.

[3] We would like to mention that our results differ significantly from the ones presented in [12]. This comes from the fact that a new version of OSLOM [12] has been released. While such a version provides better results than the previous ones, it seems that the time needed to obtain the results is more important.

[6] Roger Guimera and Luis A Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 2005.

[7] Konect. KONECT datasets, June 2014.

[8] Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E*, 80(1), 2009.

[9] Andrea Lancichinetti and Santo Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E*, 80(1):016118, 2009.

[10] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical Review E*, 80(5), 2009.

[11] Andrea Lancichinetti and Santo Fortunato. Limits of modularity maximization in community detection. *Physical Review E*, 84(6):066122, 2011.

[12] Andrea Lancichinetti, Filippo Radicchi, José J. Ramasco, and Santo Fortunato. Finding Statistically Significant Communities in Networks. *PLoS ONE*, 6(5), 2011.

[13] E. A. Leicht and M. E. J. Newman. Community structure in directed networks. *Phys. Rev. Lett.*, 100, 2008.

[14] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, 1(1), 2007.

[15] Alan Mislove, Hema Swetha Koppula, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Growth of the flickr social network. In *Proceedings of the first workshop on Online social networks*, WOSN '08, pages 25–30, 2008.

[16] M. E. J. Newman. The Structure and Function of Complex Networks. *SIAM Review*, 45(2):167–256, 2003.

[17] Tore Opsahl, Filip Agneessens, and John Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245 – 251, 2010.

[18] Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning(EMNLP-CoNLL)*, pages 410–420, 2007.

[19] M. Rosvall, D. Axelsson, and C. T. Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23, 2009.

[20] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[21] Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, 2002.

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

[23] Beichuan Zhang, Raymond Liu, Daniel Massey, and Lixia Zhang. Collecting the internet as-level topology. *SIGCOMM Comput. Commun. Rev.*, 35(1):53–61, 2005.

[24] Ying Zhao and George Karypis. Criterion functions for document clustering: Experiments and analysis. Technical report, 2002.

ANNEXE

C

API Twitter

L'API -*Application Programming Interface*- **Twitter** est une interface qui permet de communiquer avec **Twitter**. Nous utilisons cette API, qui décrit un language utilisable pour requêter les données **Twitter** dont nous avons besoin. L'API est séparée en deux composantes principales :

— la composante *Streaming*, qui permet de récupérer sous forme de flux continu, des tweets correspondant à une requête ;
— la composante *REST*, qui donne accès à des informations plus ciblées via des requêtes HTTP.

Nous avons dans ce manuscrit utilisé l'API REST, qui nous permet d'obtenir toutes les informations utilisées dans les Chapitres 1 et 3. En effet, cette API nous donne accès pour n'importe quel compte qui n'est pas *privé* aux :

— abonnés et abonnements ;
— informations de profil ;
— tweets et informations liées à ces tweets ;

Tous les attributs décrits Table 3.3 sont ainsi disponibles via l'API REST. Afin de requêter l'API, nous avons utilisé la librairie **Twitter4J** qui nous a simplifié l'utilisation de l'API **Twitter** en *Java*. Néanmoins, l'API **Twitter** -peu importe la librairie utilisée- impose de fortes restrictions en terme de quantités de requêtes (Figure C.1). Ainsi par exemple, il n'est possible d'effectuer la requête *GET followers/ids* que 15 fois par 15 minutes. Sachant que cette requête ne retourne que 5.000 abonnés d'un utilisateur à chaque fois, il est possible

Rate Limits: Chart

Title	Resource family	Requests / 15-min window (user auth)	Requests / 15-min window (app auth)
GET application/rate_limit_status	application	180	180
GET favorites/list	favorites	15	15
GET followers/ids	followers	15	15
GET followers/list	followers	15	30
GET friends/ids	friends	15	15
GET friends/list	friends	15	30
GET friendships/show	friendships	180	15
GET help/configuration	help	15	15
GET help/languages	help	15	15
GET help/privacy	help	15	15
GET help/tos	help	15	15
GET lists/list	lists	15	15
GET lists/members	lists	180	15
GET lists/members/show	lists	15	15
GET lists/memberships	lists	15	15
GET lists/ownerships	lists	15	15

FIGURE C.1 – Limitations de certaines requêtes de l'API Rest.

d'obtenir une liste de 300.000 abonnés chaque heure, soit 7.200.000 par jour. Obtenir les 58.400.000 abonnés de Barack Obama se révèle ainsi une tâche fastidieuse. Cela représente un peu plus de 8 jours de requêtes. Ainsi, récupérer les abonnés et abonnements de plus de 100.000 capitalistes sociaux, qui ont pour la plupart des degrés élevés, impose des délais non négligeables. Par ailleurs, il s'agit d'avoir un algorithme tolérant aux pannes et aux exceptions afin d'éviter d'avoir à reprendre la tâche d'extraction des informations de zéro. Nous avons ainsi développé des programmes capables d'être exécutés en continu, et qui, en cas de panne de machine, sont capables de reprendre là où ils ont été stoppés avant la panne.

ANNEXE

D

Stocker et manipuler les graphes

Bases de données. Dans notre premier article [31], nous présentons des outils pour stocker et manipuler un réseau comme celui de **Twitter** fourni par Cha et al. [20] avec une approche *haut-niveau* et en utilisant une quantité de *ressources informatiques* -RAM et CPUs- *raisonnables*. Bien sûr, afin d'être efficaces dans le traitement, il est tout de même nécessaire de disposer d'une quantité suffisante de RAM (environ 40Go). Néanmoins, nous n'utilisons ni cluster de calculs, ni supercalculateur. Par ailleurs, la machine que nous utilisons possède un disque dur classique et non un SSD. Nous pensons qu'il est intéressant d'étudier ce genre de procédés pour que les chercheurs de tous les champs d'études des réseaux complexes -informaticiens, physiciens, biologistes, économistes et sociologues- puissent *reproduire* nos résultats aisément.

Puisque nous souhaitons stocker et manipuler des données, nous nous sommes naturellement focalisés sur l'utilisation de *bases de données*. Nous étudions donc la possibilité d'utiliser des bases de données *SQL*, *NOSQL* et *orientées graphe*. Les résultats de nos expérimentations montrent que la plupart des bases de données ne peuvent gérer de si grands réseaux sans distribution, et ce même pour des calculs très simples. En effet, même si **MySQL** permet de stocker le graphe de Cha et al. [20], calculer des intersections de voisinages entrants et sortants reste très fastidieux : un peu moins d'une semaine de calcul. Avec **Cassandra** [64], le format de stockage fait que même obtenir le degré des sommets peut demander plusieurs heures. Par ailleurs, nous ne sommes pas parvenus à stocker le graphe dans des bases de données orientées graphe comme **Neo4J** et **OrientDB**. Avec un simple disque dur et une seule machine, les temps de lecture/écriture étaient considérablement lents au moment de notre étude avec ces outils. Finalement, nous avions opté pour **Dex** [80], renommée **Sparksee** : haute-performance et orientée graphe, bien documentée, avec une API haut niveau, et utilisable en mémoire. Au moment de notre étude,

nous avions été capables de charger et d'étudier un sous-ensemble du graphe de Cha et al. [20] en quelques heures.

Le format CSR. **Dex** fournissait certes un moyen de traiter les données de façon haut-niveau, mais certains bugs en limitaient l'utilisation sur tout le réseau **Twitter**. Nous avons donc choisi pour la suite de nos travaux de coder en **C++** des outils capables de manipuler les graphes au format **CSR**. Le format **CSR** pour *Compressed Sparse Row* est fréquemment utilisé pour le stockage efficace de matrices creuses comme le décrit Saad [102]. Puisque nos graphes sont creux, c'est à dire que le nombre des liens du réseau est approximativement de l'ordre de celui du nombre de nœuds -on écrit parfois $O(m) \approx O(n)$, leurs matrices d'adjacence le sont également.

Pour implémenter le format CSR dans le cas d'un graphe non orienté, nous utilisons deux *Vecteurs* -au sens C++- d'entiers. Dans le premier, la distribution cumulative des degrés des sommets ordonnés par leur identifiant est stockée. Sa taille est égale au nombre de sommets n. Dans le second, les arcs du graphe sont stockés, représentés par la liste des voisins des sommets ordonnés par identifiants. Sa taille vaut deux fois le nombre d'arêtes : $2m$. Les degrés cumulatifs du premier vecteur sont utilisés pour trouver les voisins des sommets dans le second vecteur. Soit $0 \leq i < n$ l'identifiant d'un noeud. On note $cumulative(i)$ la valeur stockée à la case i du premier vecteur. Les voisins du sommet d'identifiant i se trouvent donc dans le second vecteur aux positions entre $cumulative(i-1)$ et $cumulative(i)-1$ inclus. Par exemple, comme nous pouvons le voir sur la Figure D.1, les voisins du sommet d'id 1 sont les éléments entre 1 ($cumulative(0)$) et 3 ($cumulative(1)-1$) inclus dans le second vecteur. Dans le cas d'un graphe orienté, nous utilisons 4 vecteurs, deux pour les arcs entrants, deux pour les arcs sortants.

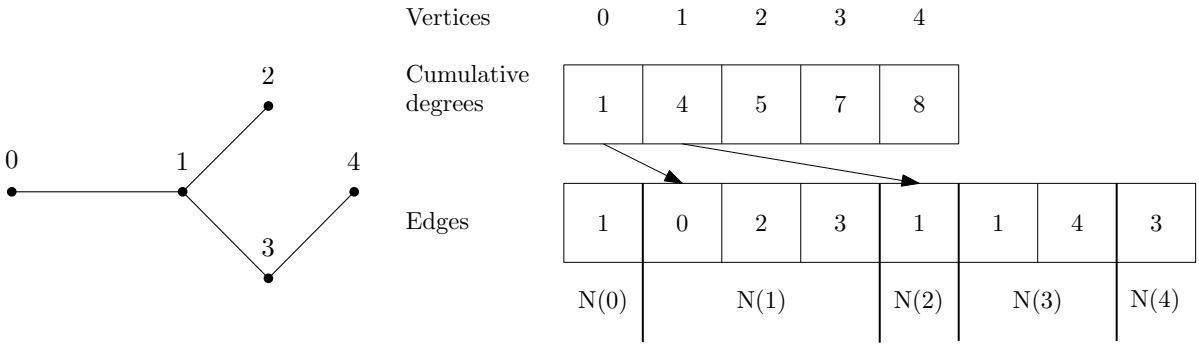

FIGURE D.1 – À droite, la représentation au format **CSR** du graphe de gauche.

OpenMP. Puisque nous avons opté pour l'utilisation de **C++**, nous avons à notre disposition un outil pour paralléliser automatiquement les traitements sur nos réseaux. Cet

outil s'appelle **OpenMP** pour *Open Multi-Processing*. Avec l'ajout d'une ligne *#pragma omp parallel* dans le code au-dessus des boucles, ces dernières sont automatiquement parallélisées. Il s'agit néanmoins de préciser quelles variables sont spécifiques à chaque processus -*private*- et lesquelles sont globales.

Nous utilisons notamment **OpenMP** dans notre implémentation des rôles communautaires du Chapitre 2 ou encore pour améliorer l'implémentation de Danisch et al. [25] fournie pour la mesure de proximité basée sur la dynamique d'opinion (Chapitre 4). Puisque les graphes que nous manipulons ont une distribution des degrés qui suit une loi de puissance, et qu'en général les traitements effectués sont faits pour chaque nœud, l'équilibrage de la charge est particulièrement difficile et les performances ne sont pas considérablement améliorées. Néanmoins, avec 40 cœurs, nous observons une division du temps d'exécution qui n'est pas négligeable même si celle-ci est loin d'être linéaire.

MapReduce. Nous avons considéré pendant la thèse l'utilisation d'outils basés sur le paradigme de programmation *Map Reduce*. **Apache Hadoop** implémente directement le paradigme *MapReduce*. Pour faire simple, il s'agit de programmer un algorithme respectant deux étapes : *Map* et *Reduce*. La première étape *Map* est utilisée pour appliquer une fonction à l'ensemble des nœuds (exemple : compter le nombre de voisins). L'étape Reduce permet elle de fusionner les résultats produits par l'étape *Map* (exemple : retourner le nombre de voisins maximal obtenu de la phase *Map*). **Apache Giraph** est une surcouche à **Apache Hadoop** qui implémente une version libre de Pregel (Malewicz et al. [79]), qui est une adaptation du modèle *BSP -Bulk Synchronous Parallel-* aux graphes. Le modèle *BSP* propose d'organiser un algorithme en plusieurs *super-étapes* (Figure D.2). Chaque super-étape est exécutée en parallèle par tous les processus, puis suivie d'une phase de communication où les processus se communiquent entre eux les résultats d'exécution de la *super-étape*. Enfin, chaque phase de communication se termine par une *barrière de synchronisation* où il s'agit d'attendre que chaque processus ait effectivement fini sa *super-étape*, communiqué et reçu tous les résultats dont il a besoin pour la *super-étape* suivante. Avec **Apache Giraph**, chaque nœud est un processus qui peut exécuter des *super-étapes* et communiquer avec ses voisins.

Pour tester ces deux approches, nous avons implémenté plusieurs algorithmes simples en utilisant **Apache Hadoop** et **Apache Giraph** : obtenir les degrés d'un noeud, calculer l'Indice de chevauchement, détecter les composantes connexes. Néanmoins, nous avons vite abandonné l'utilisation de ces outils pour adopter le format **CSR** codé en **C++** et combiné à **OpenMP**. En effet, malgré l'utilisation d'un cluster de 5 machines à 16 coeurs et 16 Go de RAM, nous avons dû affronter des problèmes de montée en charge avec **Apache Giraph**. Quant à **Apache Hadoop**, les plusieurs itérations nécessaires des étapes *Map* et *Reduce* pour la programmation d'un algorithme complet sont à chaque fois suivies d'une

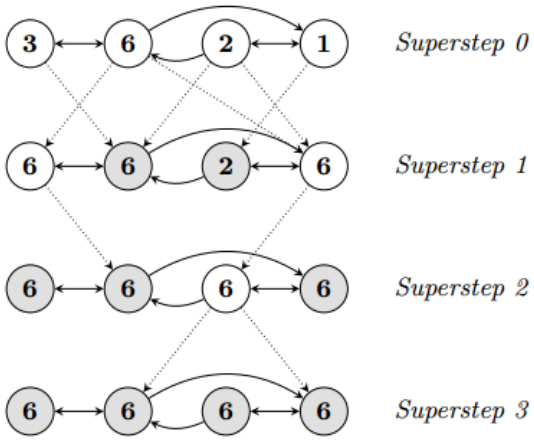

FIGURE D.2 – Figure issue de Malewicz et al. [79] qui montre comment calculer l'identifiant maximal de quatre noeuds ayant chacun un voisin avec le modèle Pregel.

lecture/écriture des résultats. Ceci ralentit considérablement l'exécution de l'algorithme. Nous obtenions donc de meilleures performances en stockant tout le graphe en mémoire au format CSR et en le manipulant de façon parallèle avec **OpenMP**. Par ailleurs, il n'est pas nécessaire dans ce format de repenser tout l'algorithme, ce qui est le cas pour utiliser **Apache Hadoop**. Nous pensons néanmoins que des approches comme **Spark** et **GraphX** qui implémentent le paradigme *Map Reduce* mais limitent la quantité d'entrées/sorties en conservant les données en mémoire sont des pistes à explorer pour la manipulation de graphes de façon parallèle ou distribuée.

Nicolas Dugué

Analyse du capitalisme social sur Twitter.

Résumé. Le sociologue Bourdieu définit le *capital social* comme : *"L'ensemble des ressources actuelles ou potentielles qui sont liées à la possession d'un réseau durable de relations"*. Sur Twitter, les abonnements, mentions et retweets créent un *réseau de relations* pour chaque utilisateur dont les *ressources* sont l'obtention d'informations pertinentes, la possibilité d'être lu, d'assouvir un besoin narcissique, de diffuser efficacement des messages. Certains utilisateurs Twitter -appelés *capitalistes sociaux*- cherchent à maximiser leur nombre d'abonnements pour maximiser leur *capital social*. Nous introduisons leurs techniques, basées sur l'échange d'abonnements et l'utilisation de *hashtags* dédiés. Afin de mieux les étudier, nous détaillons tout d'abord une méthode pour détecter à l'échelle du réseau ces utilisateurs en se basant sur leurs abonnements et abonnés. Puis, nous montrons avec un compte Twitter automatisé que ces techniques permettent de gagner efficacement des abonnés et de se faire beaucoup retweeter. Nous établissons ensuite que ces dernières permettent également aux capitalistes sociaux d'occuper des positions qui leur accordent une bonne *visibilité* dans le réseau. De plus, ces méthodes rendent ces utilisateurs *influents* aux yeux des principaux outils de mesure. Nous mettons en place une méthode de classification supervisée pour détecter avec précision ces utilisateurs et ainsi produire un nouveau score d'influence.
Mots clés : Réseaux complexes, Analyse de réseaux sociaux, Communautés, Rôles communautaires, Classification supervisée, Twitter, Influence

Social capitalism on Twitter : a survey.

Abstract. Bourdieu, a sociologist, defines *social capital* as : *"The set of current or potential ressources linked to the possession of a lasting relationships network"*. On Twitter, the friends, followers, users mentionned and retweeted are considered as the *relationships network* of each user, which *ressources* are the chance to get relevant information, to be read, to satisfy a narcissist need, to spread information or advertisements. We observe that some Twitter users that we call *social capitalists* aim to maximize their follower numbers to maximize their social *capital*. We introduce their methods, based on mutual subscriptions and dedicated *hashtags*. In order to study them, we first describe a large-scale detection method based on their set of followers and followees. Then, we show with an automated Twitter account that their methods allow to gain followers and to be retweeted efficiently. Afterwards, we bring to light that social capitalists methods allows these users to occupy specific positions in the network allowing them a high *visibility*. Furthermore, these methods make these users *influent* according to the major tools. We thus set up a classification method to detect accurately these user and produce a new influence score.
Keywords : Complex networks, Social network analysis, Communities, Community roles, Classification, Twitter, influence

Laboratoire d'Informatique Fondamentale d'Orléans
Bâtiment 3IA, rue Léonard de Vinci, B.P. 6759
45067 ORLEANS cedex 2, FRANCE