PSC MAT01

CORMAN Laura
CARRÉ Nathaniel
MORALES Juan-Pablo
MORILLEAU Rémi
NICOLAS Alexandre

Projet encadré par :
ANANTHARAMAN Nalini
ROTH Camille

04/05/2009

[ANALYSE MATHÉMATIQUE DES RÉSEAUX SOCIAUX]

RAPPORT FINAL

Introduction

L'étude des réseaux sociaux est en premier lieu **un problème de sociologues**. Comment catégoriser les entités sociales ? Comment se forment et évoluent les communautés humaines ? Le sujet est aujourd'hui plus que jamais sur le devant de la scène, avec l'émergence de méta-réseaux sociaux de grande ampleur sur Internet, via le Web 2.0 (Facebook, Twitter, les blogs...). Qui détient le pouvoir (pouvoir d'opinion, d'innovation) dans ce nouveau contexte ? Comment s'organise la diffusion des idées ?

Face à ces multiples problématiques, nul ne saurait se contenter d'une approche purement qualitative, ni d'une observation de la réalité en catégories prédéfinies telle que le propose la sociologie traditionnelle. Une analyse mathématique des données, qui va au-delà du traitement statistique, est nécessaire pour étayer les modèles et appuyer des résultats probants. Un seul exemple : c'est par une application des méthodes d'analyse de réseaux sociaux que les épidémiologistes établissent leurs **modèles de propagation des pandémies**, utilisés comme outils de décision cruciaux face au risque de pandémie de grippe porcine[1].

L'analyse mathématique des réseaux sociaux se fonde sur un formalisme qui conçoit les réseaux en termes de sommets et de liens. Les sommets sont les individus et les liens représentent les relations qui les unissent : des relations d'amitié, de famille, professionnelles, ou universitaires selon les problèmes considérés (Voir annexe pour une introduction au formalisme de graphe).

L'étude des réseaux sociaux se distingue à la fois des approches traditionnelles en sciences sociales et des mathématiques pures en ce sens qu'elle permet l'**expérimentation**. Prenons l'exemple de l'**hypothèse de petit-monde** émise en 1967 par Stanley Milgram à la suite d'une célèbre expérience. Cette hypothèse est l'idée révolutionnaire selon laquelle chacun d'entre nous pourrait être relié à n'importe quelle personne sur Terre, par l'intermédiaire d'un nombre très restreint de liens, typiquement moins de six. Cette découverte a stimulé un regain d'intérêt pour une recherche conceptuelle foisonnante sur les graphes aléatoires, menée notamment par Erdös, puis plus tard par Watts et Strogatz, recherche qui a permis en retour d'éclairer les origines de ce phénomène.

Notre projet a été d'explorer les méthodes qui permettent d'élucider les **structures sous-jacentes** à un réseau social, et de comprendre leurs fondements conceptuels. Le campus de l'Ecole nous a fourni un champ d'expérimentation idoine, puisqu'il nous a permis d'obtenir des données sur le réseau des étudiants.

FIG. 0.1.: Le réseau social en termes de sommets et de liens. Exemple d'un graphe de relations d'amitié sur Facebook

Table des matières

Première partie .

Présentation de la démarche

Des allers-retours constants entre théorie mathématique et expérimentation

Dès lors que l'on considère une modélisation mathématique de si grande ampleur, il convient de développer des outils mathématiques **en rapport étroit avec la réalité**. L'approche traditionnelle d'un problème mathématique est bien souvent compliquée, dans le sens où les raisonnements nécessaires sont eux-mêmes difficiles, mais leur enchaînement est linéaire : tel lemme entraîne tel théorème dont le corollaire nous intéresse. Ici nous devons faire face à la gestion de beaucoup de paramètres qui agissent simultanément, rendant l'approche non pas compliquée mais complexe. S'il est facile de définir certains paramètres sur les graphes, comme le coefficient de clustering, le degré moyen, le diamètre, la difficulté est de déterminer l'influence de ces indices sur le comportement des individus du graphe.

C'est pourquoi il nous a paru intéressant de voir à quel point nous pouvions modéliser ces graphes réels, en élaborant notamment un ensemble d'algorithmes qui caractérisent leurs propriétés intéressantes. Notre démarche a donc été plutôt expérimentale. Nous avons fait des **allers-retours permanents** entre les graphes théoriques, les différents algorithmes et les graphes réels : l'amélioration des algorithmes nous a permis d'évaluer la pertinence des graphes aléatoires par rapport au problème considéré et réciproquement, la génération de graphes possédant des caractéristiques particulières a permis d'évaluer la pertinence des résultats obtenus sur les graphes réels.

Notre démarche s'est donc partagée entre ces principales tâches : **chercher des données** (sur le réseau polytechnicien, ou sur Facebook) et surtout les **exploiter**, c'est-à-dire y rechercher des structures significatives et les corrélations entre ces structures.

Le fil directeur : détection des structures sous-jacentes aux réseaux sociaux.

Modéliser un réseau social sous forme de graphe social est une prémière étape qui apporte peu d'informations en soi. Les graphes ainsi obtenus sont souvent immenses, illisibles, et leurs structures ne sont pas accessibles directement à l'oeil humain. Voici tout l'enjeu qui a guidé notre travail : déployer des méthodes pour détecter des structures sous-jacentes. Ce sont bien ces structures qui nous renseigneront sur le réseau initial.

Néanmoins, lorsque nous disons : "notre travail consiste à mettre en jeu des méthodes mathématiques et des algorithmes pour détecter des structures dans un graphe social", l'interêt de notre projet n'apparaît pas immédiatement. Il faut bien voir les divers enjeux que revêt l'expression générique "recherche de structure". Ainsi, nous nous sommes penchés sur les problèmes suivants :

- regrouper les individus en **communautés**
- trouver les individus les plus **influents**
- prévoir les **liens futurs** entre individus
- **catégoriser** les différents graphes sociaux (appliquée à la catégorisation des forums et des fils de discussion)

Comme support de ces expérimentations, nous avons recueilli des informations sur le réseau étudiant de l'École polytechnique, via deux moyens : les appartenances aux binets, et les échanges sur les forums de discussion. Nous avons également recueilli un graphe de très grande taille sur le réseau Facebook. In fine, il est pertinent de tester si les structures trouvées sont indépendantes ou corrélées. Ainsi, vérifier que la création de nouveaux liens se fait préférentiellement à l'intérieur des communautés permet de valider les méthodes de recherche de communautés et de prévision des liens.

Deuxième partie .

Graphes sociaux et analyse de structure

1. Comment récupérer les données d'un graphe social ?

1.1. Simulation de graphes aléatoires

Les graphes aléatoires, introduits par Erdös, sont, tout comme la théorie des graphes en général, un domaine des mathématiques relativement jeune. Ils se développent bien entendu en étroite interaction avec l'étude de phénomènes réels dans le domaine de l'épidémiologie, de l'informatique (avec les enjeux inhérents à l'obtention du graphe d'Internet) et des graphes sociaux. En effet, dans les simulations, ils peuvent remplacer les réseaux réels, difficiles à obtenir. Nous avons choisi quelques modèles importants pour les graphes aléatoires par rapport à leurs propriétés, que nous avons ensuite programmés afin d'y effectuer nos simulations. Nous les présentons ici.

1.1.1. Le Modèle d'Erdös-Rényi

Soit n un nombre de sommets fixé. On peut alors construire $2^{n(n-1)/2}$ graphes non orientés, non pondérés. Cependant, on ne contrôle pas du tout les caractéristiques du graphe tiré. C'est pourquoi il est intéressant de définir des règles plus précises de tirage de graphe.

Le modèle d'Erdös permet de définir la probabilité de présence d'une arête. On choisit un nombre p entre 0 et 1, et on construit le graphe de manière à ce que chaque arête existe avec une probabilité p. Cette construction est l'une des plus simples qu'on puisse imaginer. Néanmoins, ce type de graphe possède des propriétés intéressantes qui en font un objet utile pour simuler des graphes de manière générale. Dans l'ensemble des graphes que nous avons étudiés, nous avons déterminé quelques paramètres qui permettent de décrire de manière efficace les graphes sociaux. Le type d'étude sera donc le même pour tous les graphes aléatoires. Il faut préciser que la plupart des résultats mathématiques concernant ce graphe sont obtenus dans le cas où $pn \to \lambda$ et $n \to \infty$.

Degré Le degré d'un sommet d'un graphe est le nombre de ses voisins. On regarde alors la distribution du degré, c'est-à-dire le nombre de sommets ayant un degré fixé en fonction du degré. Quand le nombre de sommets tend vers l'infini, on s'intéresse à un équivalent de cette distribution. Des études [10] ont montré que la répartition du degré de graphes réels était asymptotiquement équivalente à une fonction puissance, ie

$$\mathbb{P}(d_i = k) \propto k^{-\alpha}$$

où d_i est le degré du sommet i et α un nombre souvent compris entre 1 et 3.

Pour le graphe d'Erdös-Rényi cependant, la distribution du degré est exponentielle, c'est-à-dire qu'on a plutôt $\mathbb{P}(d_i = k) \propto \exp(-\lambda k)$ (distribution de Poisson).

Composante connexe Une propriété très importante des graphes est leur connexité. Par exemple, si le graphe du réseau physique du Web (routeurs...) n'est pas connexe, cela signifie

qu'il existe deux machines non reliées qui ne peuvent donc pas échanger d'information - ou au contraire que l'une ne peut pas attaquer l'autre. Dans un graphe de type Erdös-Rényi, plusieurs cas de figure peuvent se produire :

- $\lambda < 1$: Le graphe se compose de plusieurs composantes non reliées entre elles. De plus, aucune composante ne peut être trop grande.

Théorème Soient $\lambda < 1$, C_x la composante connexe contenant x, $\alpha = \lambda - 1 - \log(\lambda)$ et $a > 1/\alpha$. Alors

$$\mathbb{P}(\max_{1 \leqslant x \leqslant n} |C_x| \geqslant a \ln n) \to 0$$

ce qui signifie qu'aucune composante connexe ne peut être trop grande.

- $\lambda > 1$: Le graphe possède une composante connexe géante de taille $\mathcal{O}(n)$, les autres composantes ayant une probabilité tendant vers 0 d'avoir plus de $\mathcal{O}(\ln n)$ sommets.
- $\lambda = a \ln n$, $a > 1/2$: Avec une probabilté qui tend vers 1 lorsque n tend vers l'infini, les graphes obtenus comportent une composante géante et des sommets isolés.

Le cas $\lambda = 1$ nécessite une analyse plus fine, on considère souvent λ comme une fonction de n qui tend vers 1 en l'infini, et dont on fait un développement asymptotique.

Diamètre Il est intéressant de savoir quelle est la distance moyenne entre deux sommets. Dans les réseaux sociaux, en général, si le nombre de sommets est très grand, cette distance est courte :

Théorème Soit $\lambda > 1$ et x et y deux points choisis au hasard dans la plus grande composante connexe. Alors si $d(x, y)$ est la distance entre les sommets, $d(x, y)/\ln n \to 1/\ln \lambda$ en probabilité.

Structure locale On comprend facilement que la probabilité qu'il existe des triangles dans un graphe réel est élevée : si A connaît B et C, il est probable que B et C se connaissent. Dans un graphe d'Erdös-Rényi, il est peu probable que cela se produise, donc ce graphe modélise mal la structure locale des graphes sociaux.

1.1.2. Graphe de Barabási-Albert

Ce graphe se construit étape par étape. À chaque instant, on ajoute un sommet a qui a un nombre de voisins k fixé. La probabilité que a s'attache au sommet b est proportionnelle à $f(d_b)$ où f est une fonction croissante et d_b le degré de b (par exemple la probabilité est proportionnelle à d_b). Ainsi ce graphe est naturellement connexe, et sa répartition de degré suit une fonction puissance.

Si la probabilité est proportionnelle à d_b, alors $\mathbb{P}(d_i = k) \propto k^{-3}$. Des choix judicieux de f peuvent faire varier l'exposant de la fonction puissance de 2 à l'infini. La distance moyenne entre deux sommets est toujours de l'ordre de $\mathcal{O}(\ln n)$. Ce modèle est donc plus proche des graphes sociaux que celui d'Erdös. Cependant la structure locale n'est toujours pas bien représentée.

1.1.3. Graphe de Watts et Strogatz

Ce graphe respecte la structure locale des graphes sociaux où la probabilité de trouver des triangles est forte. On part de n sommets que l'on dispose en cercle. Chaque sommet

est relié à ses k voisins à droite et à gauche. On détermine ensuite une probabilité de rebranchement p. Toutes les arêtes qui vont du sommet i à ses voisins de droite voient leur destination modifiée avec une probabilité p, la destination étant choisie au hasard parmi tous les sommets possibles.

On observe figure 1.2 différents résultats de tirage de graphe.

En ce qui concerne la répartition du degré, elle est influencée par la distribution initiale, et a un écart-type qui augmente (cf figure 1.1).

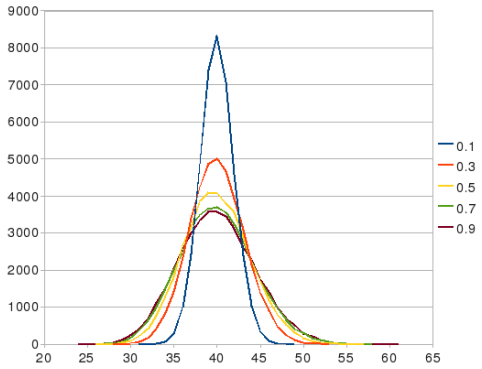
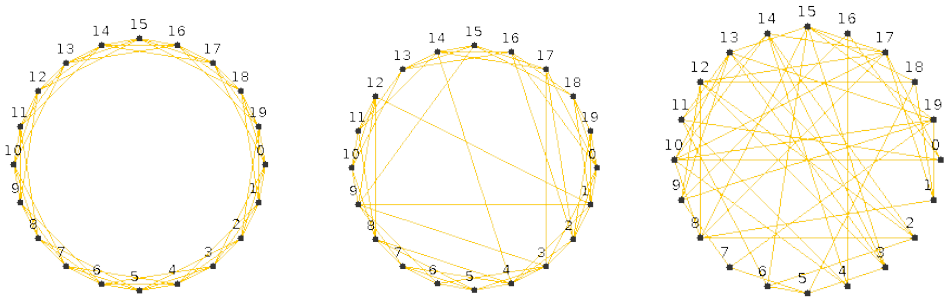


FIG. 1.1.: Répartition du degré cumulée pour différentes valeurs de p. Tests effectués sur 20 graphes tel que $n = 2000$ et $k = 20$.

En ce qui concerne la distance moyenne, on voit qu'elle est grande lorsque les arêtes ne sont pas redirigées (équivalente à $n/(4k)$). En revanche quand on modifie quelques arêtes, certaines distances sont raccourcies brusquement. Cependant cette diminution de distance n'est pas très importante [11] : supposons qu'on modifie seulement un nombre fini de liens α, ce qui revient à choisir $p = \alpha/n$. Alors la distance moyenne entre deux sommets reste un $\mathcal{O}(n)$.



(a) $p = 0$ (b) $p = 0.2$ (c) $p = 1$

FIG. 1.2.: Différents graphes de Watts et Strogatz pour $n = 20$ et $k = 3$

1.2. La récupération de graphes sociaux : le graphe de Facebook

Le fait que l'organisation des réseaux sociaux puisse se modéliser sous forme de graphe est assez intuitif. Malheureusement, quand il s'agit de récupérer des données relatives à cette structure sous-jacente, on se rend compte des difficultés que cela représente (possibilité

d'un sondage, respect des informations privées...) surtout si ces données ont pour but d'être traitées comme des données expérimentales.

A la disposition des chercheurs se trouvent certains types de graphes facilement accessibles et libres de droits, en nombre limité. Ce sont par exemple les graphes de citation d'articles scientifiques, les graphes décrivant la structure d'Internet, les graphes relatifs aux blogs... Mais leur format impose parfois d'aller jusqu'à récupérer les données à la main sur différents sites.

Pourtant, certains sites comme « MySpace », « Copains d'avant » sont les témoins privilégiés de la structure sociale de leurs utilisateurs. À ce titre, le site de Facebook est idéal : il est quasiment l'exacte traduction des rapports entre personnes, car il n'apporte pas d'autres services. En effet, on peut comparer l'information qu'apporte Facebook à celle apportée par les forums de discussion ; sur un forum, deux personnes peuvent discuter et se connaître parce qu'elles ont un intérêt commun véhiculé uniquement par le titre et le contenu du message, alors que sur Facebook, les gens se parlent parce qu'ils se connaissent.

C'est pourquoi il nous paraissait naturel de voir si nous pouvions avoir accès au graphe de Facebook, même si nos espoirs étaient faibles car nous pensions que outre les Conditions d'Utilisation, Facebook devait avoir une politique très stricte de protection des données. Nous fûmes alors surpris de découvrir que les informations qui n'étaient pas cachées par l'utilisateur étaient accessibles très facilement à partir du code source de la page.

1.2.1. Informations disponibles

Soit ident un identifiant Facebook (nombre ayant jusqu'à 7 chiffres). Alors, dans le code source de la page http ://www.facebook.com/friends/ ?id=ident on trouve la fonction

onloadRegister(function()Friends.initialize(ident,0,"everyone",0,50,[caractéristiques])

qui contient dans [caractéristiques] les identifiants de tous les amis de la personne.

Nous avons ainsi accès aux personnes du réseau France qui n'ont pas modifié les valeurs par défaut de Facebook, ainsi qu'à celles qui ont rendu leur profil totalement public.

Pour récupérer l'information, nous nous sommes servis d'une expression régulière dont le traitement était rendu possible par java.util.regex, qu'il s'agissait de retrouver sur la ligne qui décrit les paramètres de Friends.initialize. Sur cette ligne il est également possible de retrouver le nom, les réseaux, l'adresse de la photo de chaque utilisateur qui n'a pas masqué ces informations. Nous nous sommes contentés de récupérer les identifiants car notre but n'est pas de constituer une base de données à des fins commerciales - mais nous nous sommes rendus compte que cela n'était pas très difficile.

1.2.2. Programmation

1.2.2.1. Un grand nombre de données à traiter

Le site Facebook compte environ deux cents millions d'utilisateurs, le réseau France (sur lequel les gens ne prennent souvent pas la peine de masquer leurs amis) environ 3 millions d'utilisateurs. Chaque personne a en moyenne 200 amis, il fallait donc trouver un moyen efficace de stocker les données.

Nous avons choisi de coder les identifiants, puis de leur attribuer une fonction de hachage (reste modulo un grand nombre premier). Chaque utilisateur est ensuite inscrit avec la liste de ses amis dans un fichier qui porte le nombre de hachage de l'identifiant. L'objet java que nous avons utilisé est un RandomAccessFile car il permet d'écrire un fichier bit à bit, ce qui

permet de stocker les informations de la manière la plus compacte possible. Pour stocker quelqu'un, on écrit son identifiant puis les identifiants de ses 250 premiers amis, puis à nouveau son identifiant, puis les 250 amis suivants, etc. S'il reste moins de 250 amis, on complète la ligne avec des zéros. Le dossier contenant les informations (récupérées pendant 20 heures, 71 517 personnes) a ainsi une taille de 86,3 Megaoctets.

1.2.2.2. Un autre parcours de graphe

Pour explorer le graphe de Facebook, nous avons créé une pile qui contient les prochains identifiants à explorer (dont on a vérifié qu'ils n'avaient pas déjà été explorés) qui est en fait un HashSet<Integer>. Etant donné que chaque personne peut avoir beaucoup d'amis, nous avons voulu éviter que l'algorithme reste « piégé » dans une seule partie du graphe. C'est pourquoi nous avons choisi de créer également une zone de cache (LinkedList<Integer>) qui contient les sommets qui vont être traités immédiatement. Lorsqu'elle est vide, elle est remplie à partir de la pile en y choisissant aléatoirement 50 identifiants.

Cette méthode permet d'explorer tout le graphe. Nous l'avons préférée aux méthodes traditionnelles de parcours en largeur ou en profondeur car ces méthodes, tant qu'elles n'ont pas exploré un nombre significatif de sommets, ne constituent qu'un sous-graphe qui n'est pas très représentatif de la structure du graphe total (par exemple, le parcours en profondeur génère une structure arboresente qui s'exploite mal). Or nous voulions que nos données soient exploitables le plus vite possible car il nous paraissait tout de même difficile d'arriver à explorer la totalité du graphe de Facebook avant, par exemple, d'en être empêchés par le site.

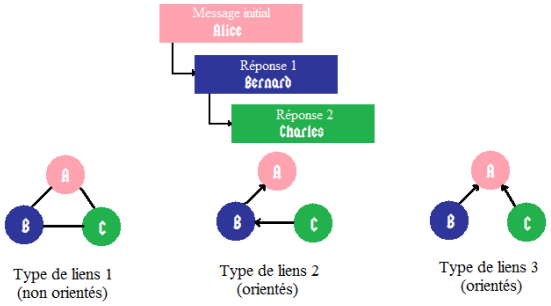
1.2.2.3. Sécurité de l'algorithme

Nous avons essayé autant que possible d'être discrets dans nos requêtes au site Facebook. Tout d'abord, nous avons distribué le programme, de manière à ce que quatre utilisateurs différents explorent Facebook avec des navigateurs distincts.

Nous utilisions les packages org.apache.http.client et org.apache.http.core pour pouvoir accéder à Internet à partir d'un programme Java. Ils nous ont permis d'interroger Facebook au nom d'un navigateur web usuel (dans notre cas Firefox) car le site bloque les requêtes des navigateurs qu'il ne connaît pas.

Ensuite, afin de ne pas assaillir le site de requêtes, chaque utilisateur attendait entre zéro et une seconde avant de lancer une nouvelle requête. Malheureusement, cela n'aura pas empêché Facebook de désactiver les comptes par lesquels nous avions effectué une requête (cf. figure 1.3a).

(a) Comptes désactivés



(b) Différents types de liens possibles

1.2.2.4. Exploitation

Les RandomAccessFile tels que nous les avons écrits ne peuvent pas être lus à partir des éditeurs de texte classiques. Nous avons donc programmé un petit package qui se charge d'afficher un fichier choisi sous la forme décrite plus haut : chaque ligne commence par l'identifiant de la personne considérée puis suivent 250 identifiants de ses amis, et ce sur plusieurs lignes, la ligne étant complétée par des zéros s'il n'y a plus d'amis.

Pour l'instant la récupération des données nous a occupés entièrement, mais il est possible d'implémenter certaines fonctions très basiques comme le degré moyen, le coefficient de clustering car ces opérations s'effectuent en $\mathcal{O}(n \cdot d)$ où n est le nombre de sommets du graphe et d le degré moyen.

Nous avons ainsi déterminé que sur les 71 517 personnes récupérées, 41 251 masquent leurs amis, les 30 266 restants ont en moyenne 282 amis.

Cependant, on comprend ici l'intérêt d'opérations sur les graphes qui aient une complexité minimale, comme l'algorithme de recherche de communautés par les chaînes de Markov ou encore l'algorithme de recherche du groupe de personnes le plus influent par une méthode de *hill-climbing*.

1.3. Données sur le réseau étudiant : «binets» et forums de discussion

Les binets sont des associations d'élèves qui regroupent chacun entre deux ou trois et plus d'une centaine de membres. À partir des listes des membres, il est naturel de former un graphe biparti binets-élèves, dont la projection sur l'ensemble des élèves donne une matrice d'adjacence entre élèves. Cette matrice, en tant que projection d'un graphe biparti, possède quelques particularités par rapport aux graphes sociaux en général et constitue notre premier terrain d'expérimentation.

Les données des forums de discussion, rangés par thèmes de conversation et sur lesquels les élèves échangent questions, constats, indignations, remerciements,... constituent une base de données idoine pour l'étude des graphes sociaux. En effet, elles permettent d'obtenir un graphe avec un nombre conséquent de sommets et d'arêtes (légèrement plus que 1000 pour les premiers, supérieur à 50000 pour les secondes), mais malgré tout suffisamment restreint pour pouvoir être manipulé par des algorithmes de complexité cubique. De plus, la frontière des individus à considérer est très nette, en ce sens qu'il n'y a nul besoin d'avoir recours à l'arbitraire pour la définir, et les individus impliqués sont actifs et peuvent être suivis sur une échelle de temps longue, de l'ordre de l'année.

La particularité de ce graphe est que, selon le type de liens considéré, il peut être vu comme un graphe orienté ou non orienté (Voir figure 1.3b). Bien entendu, il ne s'agit en aucun cas d'étudier les forums de discussion pour eux-mêmes, pas plus que le graphe des binets pour lui-même ; en revanche, tous deux constituent un champ d'expérimentation adéquat pour l'étude des graphes sociaux, et nombre des résultats auxquels ils donnent accès ou qu'ils permettent de tester peuvent être généralisés à d'autres types de graphes : citons ainsi la recherche de communautés, la recherche d'individus influents ou bien encore la catégorisation structurelle des discussions (pour les forums de discussion).

Les données étaient accessibles sous forme de dossiers, correspondant aux différents forums et composés de fichiers de code source. Il a fallu extraire les lignes intéressantes de ces derniers pour reconstituer l'arborescence des fils de discussions, filtrer en fonction de la date et obtenir les matrices d'adjacence selon le type de liens étudié. Les détails de l'algorithme sont fournis en annexe.

2. Structures statiques

2.1. Recherche de communautés

Bien que riche en informations, la topologie du graphe se révèle insuffisante pour rendre compte de l'existence de groupes d'individus fortement soudés, baptisés communautés. En effet, leur caractérisation par des cliques, ie des sous-graphes complets, est inadéquate, dans la mesure où les sommets d'une communauté sont rarement tous liés les uns aux autres, et il faut élaborer des méthodes de recherche de communautés plus fines.

2.1.1. Deux approches opposées

De manière générale, on distingue deux types d'approche [15]:

- d'un côté, les méthodes agglomératives partent de communautés à un élément, dont elles fusionnent les plus proches de manière itérative, ce qui conduit à un dendrogramme. Un critère d'arrêt permet de décider à quel niveau de fusion des communautés interrompre le processus.
- de l'autre côté, les méthodes par scission débutent avec une grande communauté regroupant tous les sommets et, à chaque étape, scindent une communauté en deux. Encore une fois, un critère d'arrêt permet l'interruption du processus avant l'éclatement du graphe en sommets isolés.

Dans chacun des cas, il faut définir une heuristique pour déterminer les communautés à fusionner ou à scinder à chaque étape.

2.1.2. Algorithmes classiques

La méthode Walktrap La méthode Walktrap, agglomérative, évalue une distance entre sommets en effectuant des marches aléatoires à partir de chacun d'entre eux, avec une probabilité de se déplacer vers un de ses voisins proportionnelle au poids de l'arête les reliant. Par la suite, on cherche à fusionner les communautés proches au sens de cette distance. L'idée qui motive cette méthode est qu'une marche aléatoire tend à rester piégée au sein d'une communauté.

Reste à déterminer la longueur de la marche aléatoire à effectuer. On peut déjà noter l'équivalence entre la simulation d'une telle marche aléatoire et l'exponentiation de la matrice stochastique $P = D^{-1}A$, où D est la matrice diagonale des degrés (ou des poids) des sommets, et A est la matrice d'adjacence du graphe, puisque $P^t(i,j)$ est la probabilité de passage de i à j en t étapes.

Pour peu que P soit fortement irréductible, par application du théorème ergodique, $(\mu P^t)_t$ tend vers une probabilité invariante π, quelle que soit la mesure initiale μ. Sans faire appel au théorème ergodique, ceci peut se démontrer de la façon suivante : $S = D^{\frac{1}{2}}PD^{-\frac{1}{2}} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ est symétrique et se diagonalise donc sur la base orthonormale (s_α) avec des valeurs propres λ_α réelles, de module inférieur à 1, car $\forall i, \|P(i,\bullet)\| \leq 1$. De plus, $PV = V$ se résout en : V est un vecteur uniforme. Ainsi, en revenant à P,

$P = \sum \lambda_\alpha (D^{-1/2} s_\alpha)({}^T s_\alpha) D^{1/2}$, puis, par orthonormalité, $P^t = \sum \lambda_\alpha^t (D^{-1/2} s_\alpha)({}^T s_\alpha) D^{1/2}$, avec tous les λ_α de module strictement inférieur à 1 sauf un, d'où le résultat.

Par conséquent, pour éviter la convergence vers la probabilité invariante, il ne faut pas prendre une longueur de marche aléatoire trop élevée. Mais il faut également laisser le temps à la marche de se propager dans toute la communauté, ce qui impose une longueur au moins de l'ordre de son diamètre moyen. La détermination d'une valeur adéquate de la longueur entre ces deux extrêmes demeure une question ouverte, mais il a été montré expérimentalement que la méthode est robuste à des variations de cette longueur sur la plage $[4, 10]$ (voir [13]).

Une alternative consiste à étudier, pour deux sommets i et j, la différence de leurs probabilités de présence $P^t(i,k)$ et $P^t(j,k)$ (pour tous les sommets k) après une marche aléatoire de longueur t et, après normalisation, d'en faire une distance. L'idée sous-jacente est que deux sommets d'une même communauté "voient" les sommets extérieurs de manière semblable.

Dans les deux cas, le critère d'arrêt est la maximisation d'une quantité Q appelée modularité et définie comme suit :$Q = \sum_{Communauté i} e_{ii} - a_i^2$, où e_{ij} est la fraction des arêtes du graphe allant de la communauté i à la communauté j, et $a_i = \sum_j e_{ij}$. En somme, il s'agit de maximiser la fraction de liens intracommunautaires (e_{ii}) par rapport aux liens sortants (a_i).

En règle générale, la maximisation de cette quantité sans heuristique est un problème NP-complet, puisqu'il s'agit de calculer cette valeur sur toutes les partitions en communautés envisageables. Certes, la modularité peut être actualisée en temps constant pour de petits changements, mais, même en supposant que l'on puisse parcourir toutes les possibilités en n'effectuant à chaque fois que de petites modifications, le nombre de partitions pour un graphe à N sommets est $B_N = \sum_{k=1}^{N} \sum_{i=1}^{k} \frac{(-1)^{k-i} i^n \binom{k}{i}}{k!} \geq 2^N$ (partitions en deux communautés). D'où la nécessité de choisir judicieusement les communautés à fusionner et l'introduction d'une distance. Un mauvais choix peut conduire à une baisse de la modularité ou à piéger l'algorithme autour d'un maximum très local de cette quantité.

Le nombre B_N est dit nombre de Bell. Pour obtenir la formule explicite ci-dessus, nous avons considéré le nombre de partitions en k communautés. Si l'on numérote les communautés et qu'on autorise les communautés vides, il vaut k^N, dans la mesure où pour chacun des N sommets on choisit une des k communautés. Pour se départir des communautés vides, on ôte les partitions où la communauté 1 est vide, au nombre de $(k-1)^N$. Pour prendre en compte que ce n'est pas forcément la communauté 1 qui est vide, on multiplie par $\binom{k}{1}$, mais on a alors supprimé deux fois les partitions à deux communautés vides, ce qui impose d'ajouter $(k-2)^N \binom{k}{2}$, et ainsi de suite... Enfin, pour éliminer la numérotation des communautés, on divise le tout par $k!$. Il suffit alors de sommer sur k allant de 1 à N pour retrouver la formule présentée.

Si l'on se contente d'une formule de récurrence, pour calculer B_{N+1}, on isole la communauté dans laquelle se trouve le premier sommet et qui en contient k autres. Reste alors à en placer $N-k$ et à tirer au hasard les k sommets parmi N (le premier sommet ne peut être choisi). On aboutit ainsi à $B_{N+1} = \sum_{k=1}^{N} \binom{N}{N-k} B_{N-k} = \sum_{k=1}^{N} \binom{N}{k} B_k$.

Comme le nombre de sommets que nos graphes contiennent demeure raisonnable, nous avons pu implémenter un algorithme qui cherche à chaque étape les individus ou communautés à fusionner parmi les voisins de chaque communauté, après un filtrage qui élimine les moins pertinents, et choisit la fusion qui maximise la modularité. Nous fournissons

les points clés du programme en annexe. L'évolution de la modularité dans le cas du graphe des binets est représentée sur la figure 2.1. Du point de vue de la modularité, il est avantageux au début de créer uniquement des communautés de couples de sommets très liés. Ce n'est qu'une fois ceux-ci épuisés que ces communautés peuvent s'élargir en phagocytant des membres isolés ou en fusionnant entre elles. On trouve finalement une grosse communauté de taille assez disproportionnée par rapport aux autres. On peut se demander si cela reflète une grande cohésion entre des individus de coeur très soudés, ou bien s'il s'agit d'un artéfact de l'algorithme.

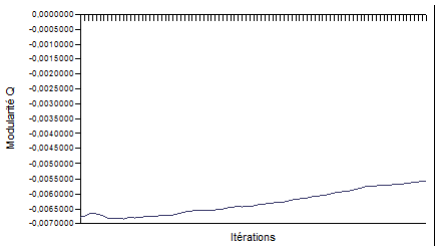


Figure 2.1.: Evolution de la modularité

Méthode fondée sur la centralité d'intermédiarité Mentionnons maintenant une méthode par scission. Elle est fondée sur la relative rareté des liens intercommunautaires, qui en fait des lieux de passage incontournables pour aller d'un sommet à un autre par le plus court chemin, pour peu qu'ils soient dans des communautés différentes. Le principe de l'algorithme consiste donc à éliminer les liens les plus fréquentés par les plus courts chemins, autrement dit de plus forte centralité d'intermédiarité. Pour déterminer ces derniers, nous avons utilisé la méthode de Roy et Warshall, qui introduit progressivement les sommets par lesquels ils peuvent passer et en les modifiant en conséquence après chaque introduction. Sa complexité temporelle est en n^3, avec n le nombre de sommets.

Pour tester ces résultats et cette méthode, nous avons construit une classe de graphes contraints, pour lesquels densité intracommunautaire et totale et longueur moyenne des plus courts chemins sont imposées. Tout d'abord, nous définissons le nombre et les tailles souhaités pour les communautés. Puis, pour ce qui est des deux premières valeurs, la démarche est simple : après décompte des arêtes intracommunautaires, d'une part, et intercommunautaires, d'autre part, nous insérons aléatoirement un tel nombre d'arêtes. Pour contrôler le point plus délicat du plus court chemin, nous faisons l'hypothèse qu'une longueur moyenne trop élevée est due à des communautés peu ou pas reliées entre elles, et non aux chemins intracommunautaires.

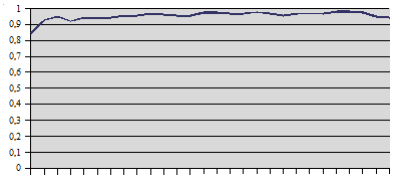


Figure 2.2.: Fraction de plus longs des plus courts chemins étant intercommunautaires en fonction du nombre d'itérations

Nos simulations confirment cette hypothèse, dans la mesure où les sommets reliés par les plus longs chemins, voire pas reliés, sont majoritairement - à 90% environ - situés

dans des communautés différentes (cf figure 2.2). À partir de là, l'idée est d'ajouter une arête en guise de «pont» entre ces sommets mal connectés – ce qui est également bénéfique pour leur voisinage, et, pour compenser, de supprimer une arête de même type, ie intracommunautaire ou intercommunautaire, de manière aléatoire.

Pour augmenter la valeur moyenne du plus court chemin, on procède de manière entièrement aléatoire, en remplacant des arêtes au hasard, ce qui est gênant, mais pas trop, car il s'agit le plus souvent de diminuer la valeur de l, et non de l'augmenter. À partir de là, on réalise un asservissement en imposant que les changements effectués pour raccourcir / allonger la longueur moyenne doivent être d'autant moins nombreux que l'écart entre la valeur actuelle et la valeur souhaitée est faible.

On se doute que les performances seront assez moyennes : «allonger» s'en remet au hasard, et « raccourcir » permet au mieux d'atteindre un minimum local. Cependant, pour peu que la valeur demandée ne soit pas trop éloignée de la valeur dans le graphe construit initialement, on obtient, après un temps variable, un graphe correspondant à nos exigences. Le graphe ci-dessous présente l'évolution de la longueur moyenne des plus courts chemins pendant une phase de raccourcissement. On observe que, passé un certain stade de déformation, elle augmente brusquement : l'algorithme est incapable de diminuer davantage sa valeur.

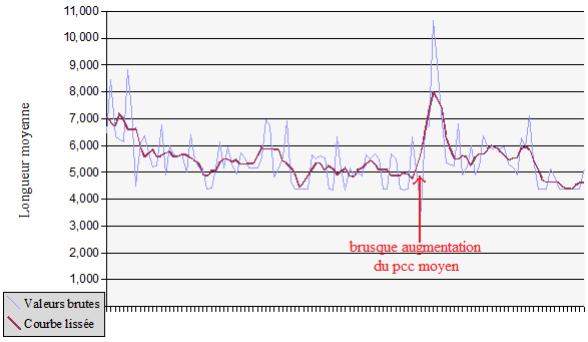


Figure 2.3.: Réduction de la longueur moyenne des plus courts chemins

Evaluation de la longueur moyenne des plus courts chemins au sein d'une communauté

On considère un sommet particulier de cette communauté et on note N(i) le nombre de sommets atteints par une marche de longueur inférieure ou égale à i à partir de ce sommet (ils seront dits *contaminés* par la suite).

- N(0) = 1
- Si l'on a N(i) sommets contaminés, considérer les N_{intra}(en moyenne) arêtes partant de ce sommet revient à choisir N_{intra} sommets pour chacun des N(i) contaminés. On les ajoute progressivement en prenant garde aux chevauchements. Après prise en considération du k-ième contaminé sur N(i), on a u_k contaminés au total, avec :
  - $u_0 = N(i)$
  - $u_k = u_k - 1 + N_{intra}(1 - \frac{u_{k-1}}{N_{total}})$, où $1 - \frac{u_{k-1}}{N_{total}}$ est la fraction de sommets pas encore contaminés

  ce qui donne la formule suivante pour u_k : $u_k = N_{total} - (N_{total} - N(i))(1 - \frac{N_{intra}}{N_{total}})^k$

donc $N(i+1) = u_{N(i)} = N_{total} - (N_{total} - N(i))(1 - \frac{N_{intra}}{N_{total}})^{N(i)}$

Comme il y a $N(i+1) - N(i)$ contaminés au i-ème tour, la longueur moyenne vaut : $\bar{l} = \frac{1}{N_{total}} \sum_i (i+1)[N(i+1) - N(i)]$

Avec $N_{total} = 300$ et $N_{intra} = [N_{total}.d_{intra}]$, en évaluant numériquement $\bar{l}$, on trouve les résultats suivants :

d_{intra}	$\bar{l}$ théorique	$\bar{l}$ expérimental
0,05	2,36	2,4
0,10	1,92	1,95
0,15	1,85	1,85
0,20	1,80	1,80
0,25	1,74	1,75
0,30	1,70	1,70
0,35	1,65	1,65

Table 2.1.: Valeurs théoriques et expérimentales (15 simulations) de $\bar{l}$. Ecart-type : 0,020

Test de validité

La méthode de recherche des communautés est-elle capable de retrouver les communautés prédéfinies dans les graphes contraints ? Pour quantifier cette capacité, nous introduisons un coefficient de qualité $q = \frac{1}{N} \max_{\sigma \in S_n} \sum_i C_i \cap C'_{\sigma(i)}$, où C_i est la communauté i prédéfinie et C'_i est la communauté i trouvée par l'algorithme. Si le nombre de communautés n'est pas identique, on associe toujours de manière univoque les communautés, en laissant certaines d'entre elles sans association.

Les résultats trouvés sont assez satisfaisants, dans la mesure où l'on trouve, avec des valeurs de densité qui peuvent être observées dans des graphes empiriques, des coefficients de qualité pouvant atteindre plus de 70%. La dépendance en $\frac{d_{intra}}{d}$ est représentée sur la figure 2.4.

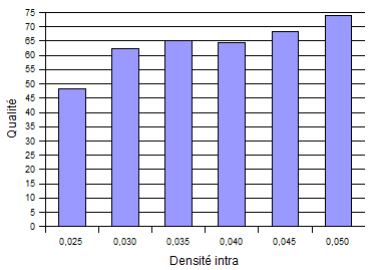


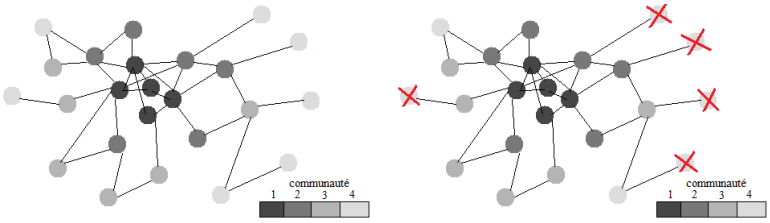
Figure 2.4.: Qualité en fonction de la densité intracommunautaire à $d = 0,01$

Communautés d'équivalence Contrairement aux autres communautés, les communautés par équivalence ne font pas ressortir des relations d'affinités, mais des similitudes de fonction. Ainsi, pour un hypothétique graphe des médecins et de leurs patients, les médecins se retrouveraient dans une communauté et les patients, dans l'autre, indépendamment de leur connaissance mutuelle.

En ce qui nous concerne, seront dans la même communauté les individus les plus à la périphérie des amas de sommets, tandis que ceux qui occupent une position centrale seront rangés dans un autre ensemble. Cette méthode était initialement conçue pour les graphes non orientés, mais son application à des graphes orientés dans le cadre des forums de discussion peut permettre de distinguer des profils d'individus : poseurs de questions, vers

lesquels pointent de nombreux liens *vs* fournisseurs de réponses, qui ont de nombreuses arêtes sortantes.

Pour implémenter cette méthode, nous éliminons à chaque étape les individus les plus vulnérables. Plus précisément, les sommets qui peuvent être déconnectés du reste du graphe par la suppression d'une arête sont supprimés à la première itération et rangés dans une même communauté "d'invulnérabilité 1,1". Cette suppression peut faire apparaître de nouveaux sommets détachables par le retrait d'une unique arête, sommets à nouveau éliminés. Une fois qu'on ne peut plus retirer de sommets, on passe aux communautés "d'invulnérabilité 2,1", qui peuvent être isolés par le retrait de deux arêtes, et ainsi de suite. Si le graphe est pondéré, ce n'est plus le nombre d'arêtes qui est considéré, mais la somme de leurs poids, ou plutôt d'une fonction de leurs poids (afin de ne pas donner trop d'importance aux arêtes très lourdes).



(a) Graphe initial (b) A la première itération

Figure 2.5.: Recherche des communautés d'équivalence

2.2. Vitesse de convergence des chaînes de Markov et recherche de communautés

Chaînes de Markov et graphes sont étroitement liés car une matrice de transition peut être vue comme la matrice d'adjacence d'un graphe. La chaîne de Markov associée à un graphe sera donc le processus qui à l'instant $t+1$ attribue au sommet i la moyenne des valeurs de ses voisins à l'instant t, pondérée par les coefficients des arêtes, en ayant renormalisé la matrice d'ajacence de manière à ce qu'elle soit stochastique (ie. la somme des coefficients sur chaque ligne vaut 1).

Il est alors naturel de chercher à exploiter cette similitude pour étudier les graphes. Ainsi, nous présentons ici des éléments mathématiques permettant de détecter des communautés au moyen d'une chaîne de Markov. Cette méthode de recherche de communauté peut être qualifiée de méthode des « vases communicants », car l'idée est d'attribuer des valeurs à chaque sommet et de laisser évoluer, l'intuition laissant à penser que les valeurs convergent plus vite à l'intérieur des communautés. Il s'agit de vérifier mathématiquement cette intuition.

2.2.1. Présentation de l'algorithme

Cette méthode est inspirée de celle proposée dans l'article de James Moody [3]. Voici ce en quoi elle consiste :

- attribuer à chaque sommet une valeur aléatoire
- attribuer à chaque sommet la moyenne de la valeur de ses voisins
- recommencer n fois

Le procédé à l'instant n ne dépend que des valeurs à l'instant n-1 : c'est une procédé de Markov. Ensuite, à la fin des n itérations, on calcule l'écart-type de la distribution des valeurs. L'utilisateur doit définir une finesse, c'est-à-dire une valeur qui permet de créer les communautés :

- choisir un sommet s_1 du graphe de valeur v_1
- pour tout autre sommet s, si sa valeur v_2 est telle que $|v_1 - v_2| \leq \sigma.finesse$, insérer s dans la communauté de s_1
- recommencer tant qu'il reste des sommets

L'idée de l'algorithme consiste à dire que même si la valeur de chaque sommet converge vers une seule valeur grâce au théorème ergodique, les convergences ne se font pas à la même vitesse au sein d'une communauté où les valeurs tendent à être homogènes plus vite. En quelque sorte, quand les individus sont très connectés, il existe un flot de grande valeur dans le sous-graphe qu'ils constituent, et ainsi l'information se transmet plus vite.

Les chaînes de Markov sont en effet reliées fortement aux questions d'existence de flot (quantité de liquide maximale pouvant s'écouler par les arêtes) : par exemple, la transience ou la récurrence d'une chaîne de Markov dans un arbre infini est liée à la capacité des flots pouvant circuler dans l'arbre [2].

Cependant, un algorithme de recherche de groupe où le flot est maximal n'est pas envisageable d'un point de vue de la complexité de l'algorithme : en effet une recherche naïve d'un groupe optimal est exponentielle en le nombre de sommet du graphe, et un algorithme de flot tel que l'algorithme de Ford et Fulkerson a une complexité en $\mathcal{O}(|\mathcal{A}| + |\mathcal{S}|)$ où $\mathcal{A}$ est l'ensemble des arêtes et $\mathcal{S}$ l'ensemble des sommets. On ne peut malheureusement pas envisager une méthode de type *hill climbing* car la fonction qui à un ensemble de sommets associe la valeur maximale du flot du sous-graphe n'est pas sub-modulaire (voir section 3.1). Si S et T sont deux ensembles de sommets, $S \subset T$ et v un sommet, la valeur du flot maximal de $\{S \cup \{v\}\}$ est potentiellement plus petite que celle de $\{T \cup \{v\}\}$.

C'est pourquoi il peut être intéressant d'accéder à un paramètre de ce flot grâce à une chaîne de Markov, facilement simulable. Ici, on effectue $2|\mathcal{S}||\mathcal{A}|$ calculs pour affecter des valeurs, puis dans le pire des cas $\frac{|\mathcal{S}|(|\mathcal{S}|-1)}{2}$ comparaisons (bien moins en moyenne plutôt de l'ordre de $|\mathcal{S}|$).

2.2.2. Justification mathématique des vitesses de convergence différentes

2.2.2.1. Chaîne de Markov et convergence

Le processus d'attribution de valeur à l'instant n ne dépend que des valeurs attribuées à chaque sommet à l'instant $n-1$. C'est pourquoi il peut être modélisé par une chaîne de Markov.

Commençons par quelques définitions :

Définition : La matrice de transition K d'une chaîne de Markov définie sur un espace d'état M est dite fortement irréductible si :

$$\forall (x, y) \in M \times M,\ \exists k \in \mathbb{N},\ K^k(x, y) > 0$$

Définition : Soit K la matrice de transition d'une chaîne de Markov définie sur un espace d'état M et μ une distribution de probabilité sur M. On note :

$$\mu K(y) = \sum_{x \in M} \mu(x) K(x, y)$$

On a le théorème suivant :

Théorème (ergodique) : Soit K la matrice de transition d'une chaîne de Markov. Si K est fortement irréductible alors K admet une probabilité invariante π et quelle que soit la distribution de probabilité initiale μ,

$$\lim_{n \to +\infty} \mu K^n = \pi$$

Ainsi la valeur de chaque sommet converge vers une même valeur finale. Dans le cas de la matrice d'un graphe social, la probabilité que la matrice de transition associée soit irréductible est très forte. On considèrera que c'est le cas. De plus, on remarque rapidement que si tous les sommets ont la même valeur, le processus décrit par l'algorithme ne modifie pas la répartition des valeurs :

$$\pi(x) = \frac{1}{|M|}$$

Il faut donc s'intéresser plus précisément à la vitesse de convergence des chaînes de Markov pour justifier que les valeurs des sommets sont plus proches pour des sommets appartenant à la même communauté.

2.2.2.2. Vitesse de convergence

Il existe quelques approximations canoniques de la vitesse de convergence des chaînes de Markov [4]. On a un produit scalaire dans $\mathbb{C}$ sur les fonctions définies sur l'espace des états M qui est tel que :

$$\langle f, g \rangle = \sum_{x \in M} \pi(x) \overline{f(x)} g(x)$$

Si l'on dispose d'un opérateur K, on définit son adjoint K^* comme l'opérateur tel que

$$\langle f, Kg \rangle = \langle K^* f, g \rangle$$

Définissons le trou spectral λ de K matrice de transition comme la plus petite valeur propre non nulle de $I - 1/2(K + K^*)$. On a alors le résultat suivant :

Théorème : Soit K telle que $K = K^*$, fortement irréductible, A une partie de M et μ une probabilité quelconque sur l'espace des états. Alors

$$|\mu K^n(x, y) - \pi(y)| \in \mathcal{O}(e^{-\lambda n})$$

On a alors une majoration de la vitesse de convergence de la chaîne de Markov. On comprend qu'il n'est pas possible d'obtenir une minoration de cette vitesse sans information supplémentaire sur la probabilité de départ, puisque par exemple cette vitesse est nulle lorsque la distribution d'origine est la probabilité stationnaire.

Cela nous donne tout de même une justification du bon fonctionnement de l'algorithme.

2.2.2.3. Application à la recherche de communautés

Il est difficile de rechercher les valeurs propres de très grands graphes comme ceux manipulés dans le cas des graphes sociaux. C'est pourquoi nous présenterons d'abord quelques exemples théoriques.

Modélisation d'un graphe social structuré en communautés Considérons des graphes formés à partir du graphe initial en fonction des communautés ; tout d'abord, il y a les sous-graphes de chaque communauté, dont on a vu qu'ils étaient par définition assez connectés. Ensuite, on peut également considérer le graphe projeté des communautés. Chaque sommet représente une communauté C_i. Il faut alors définir la probabilité de transition de C_i à C_j. Définissons

$$\phi(C_i, C_j) = \sum_{(x_i,x_j)\in C_i\times C_j} p(x_i, x_j)$$

où $p(x_i, x_j)$ est la probabilité de passage dans le graphe d'origine.
La probabilité de passage est alors :

$$\hat{p}(C_i, C_j) = \frac{\phi(C_i, C_j)}{\sum_{C_k} \phi(C_i, C_k)}$$

On remarque que l'intuition veut que $\hat{p}(C_i, C_i)$ soit grand (forte probabilité de rester dans la même communauté).

Calculons donc le trou spectral d'un graphe de communauté pour le comparer à celui du graphe projeté.

Le cas du graphe complet Supposons que chaque communauté soit un graphe complet, avec $p(x, x) = 0$ (la moyenne ne prend pas en compte la valeur du sommet), et que le graphe projeté soit un graphe complet lui aussi mais avec $\hat{p}(C, C)$ non nul, voire grand. On notera I la matrice identité et

$$J = \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$$

Alors la matrice d'une communauté de cardinal n sera du type $\frac{1}{n-1}(J - I)$, et celle du graphe projeté (s'il y a m communautés) du type $\beta J - (\beta m - 1)I$, ie.

$$K_{proj} = \begin{pmatrix} 1-\beta(m-1) & & \beta \\ & \ddots & \\ \beta & & 1-\beta(m-1) \end{pmatrix}$$

où β est un paramètre variant entre 0 et $1/(m-1)$ qui traduit la qualité de la transmission de l'information dans le graphe projeté ($\beta = 1/(m-1)$ représente la meilleure transmission possible). Alors :

$$\lambda_{inter} = \beta m$$

et

$$\lambda_{intra} = \frac{n}{n-1}$$

Si β est grand (bonne transmission de l'information) et le nombre de communautés plutôt petit, ce qui est le cas dans un tel graphe, la vitesse de convergence de la communauté est plus importante que celle du graphe projeté (bien entendu on a négligé l'influence de l'extérieur sur la communauté, ce qui rend le résultat encore plus intuitif).

Graphe « en anneau » On suppose maintenant que le graphe projeté est un cercle sur lequel sont disposées les communautés. La matrice correspondante est une matrice circulante, et symétrique. Un calcul classique de valeurs propres donne ainsi la valeur du trou spectral

$$\lambda_{inter} = 1 - \cos\left(\frac{\pi}{m}\right)$$

Sans surprise, plus l'anneau est grand, (ie. m grand), moins l'information se transmet car $\cos\left(\frac{\pi}{m}\right)$ est proche de 1.

Ainsi ces deux modèles montrent que l'algorithme peut fonctionner et déterminer des communautés. Reste à savoir si les différences effectives de vitesse de convergence sont assez importantes.

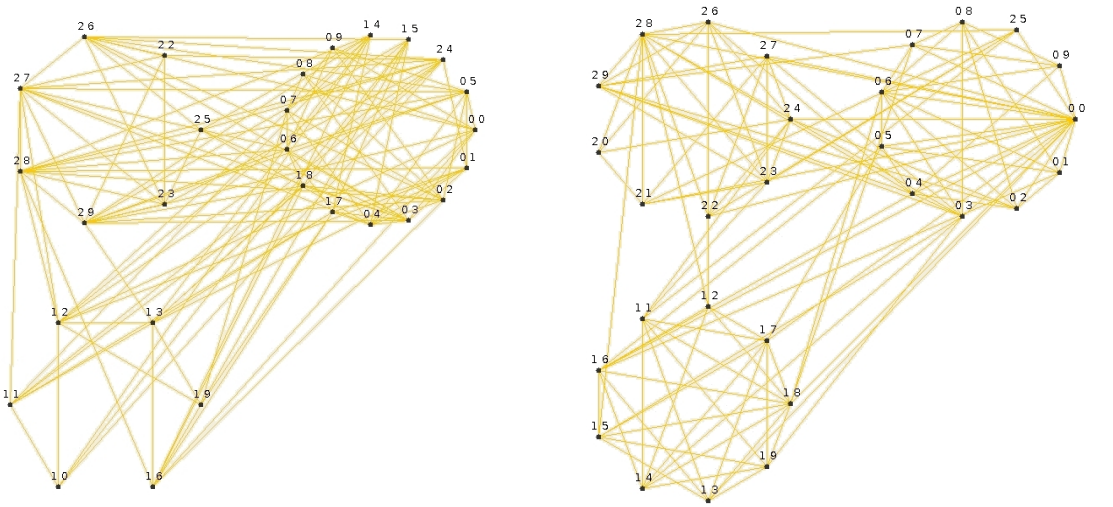
2.2.3. Expérimentation

L'expérimentation de l'algorithme s'est faite en deux temps. Tout d'abord, nous l'avons testé sur des graphes simulés, dans lesquels nous avons parfois recréé des communautés de manière artificielle. Ensuite nous avons appliqué cet algorithme aux différents graphes réels dont nous disposions.

2.2.3.1. Graphes de communauté simulés

Nous avons créé des graphes de la manière suivante : on donne un nombre de communautés, un nombre de membres par communauté, la probabilité p qu'un lien intracommunautaire existe et la probabilité q qu'un lien intercommunautaire existe (cette dernière étant naturellement choisie plus faible que la première). Nous définissons la pertinence d'un système de communautés comme le rapport entre les liens intracommunautaires et le nombre total de liens. Cela fait que l'on peut alors comparer la pertinence théorique (c'est-à-dire celle que l'on aurait si on avait retrouvé les communautés telles qu'elles avaient été initialisées) et la pertinence pratique du système de communautés de l'algorithme.

Les figures 2.6a, 2.6b montrent que le découpage est satisfaisant dans la plupart des cas, même si l'algorithme ne retrouve que très rarement les communautés à partir desquelles avait été construit le graphe.



(a) finesse 1 - pertinence théorique 0,42 - pertinence pratique 0,64

(b) finesse 0,9 -pertinence théorique 0,65 - pertinence pratique 0,69

FIG. 2.6.: 3 communautés de 10 membres - p=0,6 - q=0,3 - deux types de résultats

Il est alors normal de vouloir comparer la pertinence des communautés de départ à celle des communautés de l'algorithme (cf. figure 2.7). On remarque que l'algorithme est assez performant, même si tant que les communautés sont cloisonnées, il ne les retrouve pas complètement.

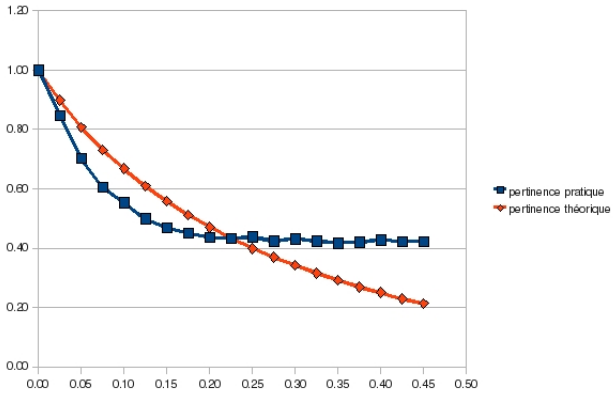


FIG. 2.7.: pertinence pour 10 communautés de 20 personnes en fonction de p (=1-q) (50 tests - en rouge la pertinence théorique, en bleu la pertinence pratique)

2.2.3.2. Graphes réels

Cependant, même en ayant distingué les communautés, il est difficile de les représenter de manière lisible sur un écran, ce qui est particulièrement visible avec des graphes réels. Pour le graphe des collaborateurs d'Erdös [6], l'algorithme ne donne qu'une seule communauté comportant quasiment tous les sommets. Il est difficile de savoir si le graphe comporte réellement un seul noyau très connecté (cf. figure 2.8). Cela est également vrai avec le graphe des associations projeté sur les élèves, d'autant plus illisible que chaque association crée un graphe complet et que par conséquent chaque sommet a un degré très élevé.

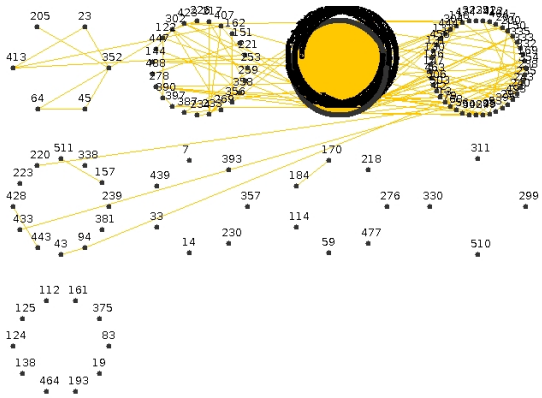


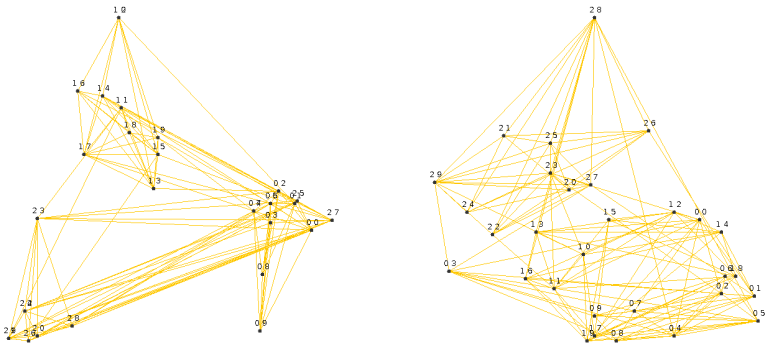
FIG. 2.8.: Graphe des collaborateurs d'Erdös - finesse 0,7

Nous avons donc pensé à une évolution de l'algorithme, qui avait d'abord pour but de représenter le graphe de manière plus agréable, puis qui a permis de retrouver de nouvelles communautés.

2.2.4. Une amélioration de l'algorithme

2.2.4.1. Un affichage plus intuitif...

Inspirée de la méthode Cosmoweb [5], cette évolution consiste à effectuer plusieurs recherches de communautés. À chaque étape, on attribue les mêmes coordonnées aux sommets d'une communauté. Les coordonnées finales sont ensuite les moyennes des coordonnées attribuées au cours des différentes itérations. La méthode Cosmoweb, quant à elle, consiste à placer les sommets au hasard, à leur attribuer une masse, puis à les rapprocher de la même manière que ce qui se passerait si les sommets étaient des points massiques attirés les uns les autres par l'attraction gravitationnelle.



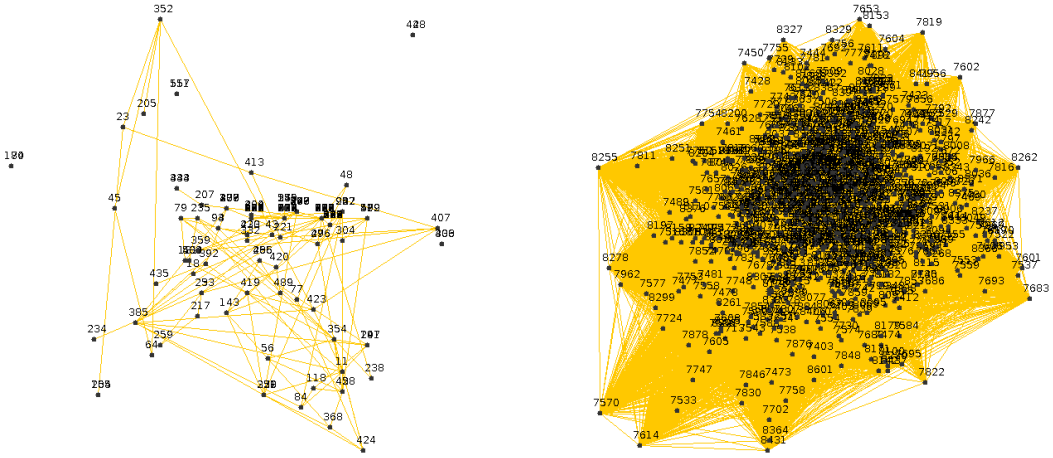
(a) Test réussi (b) Test moins probant

FIG. 2.9.: Test de l'algorithme barycentrique pour un graphe dont les communautés sont simulées (3 communautés de 10 membres, $p = 0,75$, $q = 0,2$)

Sur la figure 2.9a, on voit que les communautés ont été retrouvées de manière satisfaisante. Par exemple, le sommet 2 3 possède quatre liens intercommunautaires, et est donc excentré par rapport au reste de la communauté numéro 2. Cependant, les communautés

peuvent avoir tendance à être mélangées comme dans la figure 2.9b où la communauté 2 est séparée mais où les communautés 0 et 1 ont tendance à être mélées.

Que cette méthode permet-elle de voir sur les graphes réels ? Le graphe des collaborateurs d'Erdös de la figure 2.10a est ainsi beaucoup plus lisible que celui de la figure 2.8, bien que tous les noms de sommets ne soient pas lisibles ; cela signifie d'ailleurs qu'ils ont toujours été dans la même communauté. Cet algorithme paraît ainsi plus légitime.



(a) Graphe d'Erdös représenté grâce à l'algorithme barycentrique - Il y a 56 communautés, la plus grande ayant 290 membres (total 474) et la pertinence est de 0,82

(b) Graphe des associations de l'école projeté sur les élèves - On remarque qu'il y a un centre très connecté, et que chaque sommet a un degré très élevé, ce qui rend le graphe difficile à traiter.

FIG. 2.10.: L'algorithme barycentrique sur des graphes réels

2.2.4.2. ... Qui conduit à une nouvelle méthode de recherche de communautés

On remarque qu'à l'œil nu se dessinent déjà des communautés. Il vient donc naturellement à l'esprit de créer une partition des sommets en fonction de la disposition des sommets. Pour cela, on choisit un nombre de pixels de tolérance, puis un sommet, et on rajoute des sommets au fur et à mesure : un sommet est rajouté si son abscisse est supérieure à l'abscisse minimale moins la tolérance et inférieure à l'abscisse maximale plus la tolérance.

Nous avons effectué des tests afin de déterminer le nombre d'itérations qu'il valait mieux effectuer afin d'obtenir des résultats intéressants sur les communautés, car lorsqu'on travaille sur de grands graphes, il est préférable que les communautés apparaissent assez vite afin de réduire le temps de calcul.

itérations	médiane	troisième quartile	cardinal	max
10.00	2.95	12.3	126.95	63.70
20.00	2.60	12.15	93.55	124.25
30.00	5.10	13	63.05	180.35
40.00	7.00	17.4	40.40	250.05
50.00	4.25	15.3	24.40	340.05
60.00	3.00	17.75	17.10	379.90
70.00	2.55	18.85	9.60	407.05
80.00	2.00	3.35	5.85	457.45

TAB. 2.2.: Différentes caractéristiques des communautés du graphe des collaborateurs d'Erdös (moyennes sur 20 itérations - cardinal du graphe : 474 - itérations : nombre de calculs de communautés pour la recherche du barycentre, cardinal : nombre de communautés, max : taille de la plus grande communauté)

On remarque dans le tableau 2.2 qu'au bout de 80 itérations, on obtient une communauté géante contenant presque tous les sommets. Cela n'est pas très intéressant, mais traduit le compromis qu'il s'agit de trouver lorsqu'on recherche des communautés : le système de communautés qui maximise la pertinence est bien entendu celui où il n'y a qu'une communauté, mais l'information qu'il apporte est nulle... Le calcul de la médiane et du troisième quartile montrent également que la distribution est très inégalitaire : la plupart des communautés sont petites (moins de 10 membres en moyenne), et les trois quarts des communautés ont moins de 20 membres. Étant donné que le test a été répété un grand nombre de fois, on peut alors conclure de manière plus légitime que le graphe des collaborateurs d'Erdös comporte une grosse composante très connectée et plusieurs petites composantes.

2.2.5. Test de la validité de l'hypothèse mathématique

2.2.5.1. Une répartition des valeurs propres qui ne correspond pas aux attentes

Afin de tester la validité du modèle, nous avons voulu savoir comment varie la valeur du trou spectral d'un graphe d'Erdös en fonction de la probabilité d'existence des arêtes. Alors que nous nous serions attendus à avoir une courbe croissante pour p variant entre 0 et 1, nous avons obtenu la courbe figure 2.11.

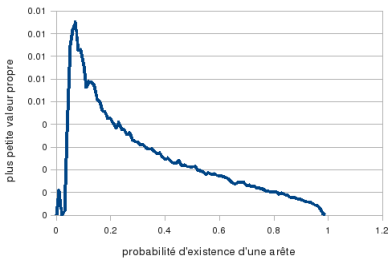
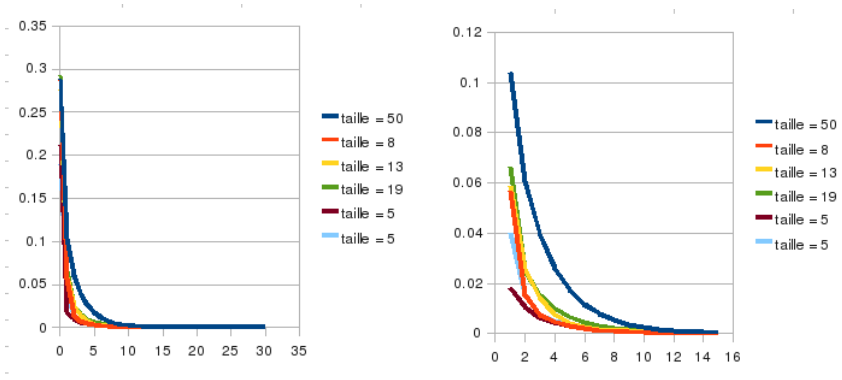


FIG. 2.11.: Trou spectral sur 1000 graphes d'Erdös de 100 sommets

Ainsi cela ne correspond pas du tout aux résultats attendus. On peut alors supposer qu'il existe une majoration de la vitesse de convergence bien plus fine et régie par d'autres paramètres.

2.2.5.2. Une décroissance exponentielle

Nous avons alors cherché à déterminer comment s'effectuait la convergence au sein des communautés. Pour cela, nous avons utilisé un graphe de 5 communautés de 10 personnes ($p = 0.8$, $q = 0.1$) dont nous avons trouvé les communautés en lançant une fois le premier algorithme de recherche de communautés, puis nous avons lançé à nouveau la première partie de l'algorithme qui attribue une valeur à chaque sommet. À chaque étape, nous avons mesuré l'écart-type sur le graphe entier puis au sein de chaque communauté. Les résultats sont reportés figure 2.12.



(a) Pour 30 itérations (b) Entre 1 et 15 itérations

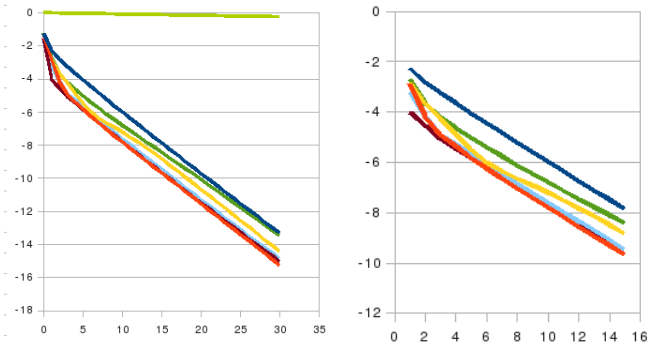
FIG. 2.12.: Ecart-type au sein de chaque communauté en fonction du nombre d'itérations (en bleu foncé : graphe entier ; en légende la taille de chaque communauté)

La décroissance est très rapide. De plus dans la justification précédente elle était exponentielle ce qui invite à tracer le logarithme des valeurs en fonction du nombre d'itérations, ce qui est fait en figure 2.13.

On remarque que la prévision portant sur la vitesse de convergence des chaînes de Markov est très mauvaise ; il y a donc dans le cas particulier de la chaîne de Markov que nous étudions des paramètres qui contrôlent de manière bien plus forte la décroissance exponentielle de l'écart-type. Il était intuitif que cette décroissance soit exponentielle car une chaîne de Markov à l'instant n ne dépend que de ce qu'elle était à l'instant $n-1$ et la loi exponentielle correspond à la solution de l'équation différentielle d'évolution sans vieillissement.

De plus, le coefficient directeur des droites obtenues dans la figure 2.13 est quasiment le même pour toutes les communautés, la différence se situe surtout dans l'ordonnée à l'origine. Cela signifie que les différences de vitesse de convergence s'estompent quand le nombre d'itérations est grand.

Malheureusement, nous n'avons pas réussi à déterminer quels paramètres étaient pertinents dans la mesure de cette vitesse de convergence.



(a) Pour 30 itérations - La courbe vert clair représente la prédiction de la section 2.2.2

(b) Entre 1 et 15 itérations

FIG. 2.13.: Logarithme de l'écart-type au sein de chaque communauté en fonction du nombre d'itérations (en bleu foncé : graphe entier ; les couleurs sont les mêmes que pour la figure 2.12)

2.2.5.3. Conclusion sur les paramètres qui font que l'algorithme donne des résultats satisfaisants

Si nous n'avons pas pu les déterminer, nous pouvons cependant nous en servir « en boîte noire », notamment en utilisant le fait que les différences ne sont plus significatives au bout de 30 itérations (cf. figure 2.12). Empiriquement, pour la plupart des simulations, nous avons constaté que 7 itérations permettaient de déterminer les communautés les plus satisfaisantes sur les graphes dont nous disposions. Dans l'objectif de traiter de grands graphes comme le graphe de Facebook, il est intéressant d'avoir peu d'itérations à faire, cependant, cela nécessite tout de même de déterminer le nombre d'itérations optimal pour obtenir la plus grande différence d'écart-type possible.

Il serait bien entendu idéal de pouvoir déterminer exactement en fonction de quoi la vitesse de convergence varie, mais on pourrait également étudier l'influence de divers paramètres sur cette vitesse, comme la distribution initiale des valeurs (dans notre programme elle est uniforme sur $[0, 1]$).

3. Structures dynamiques

3.1. Recherche des individus influents d'un réseau

Toute une classe de problèmes en sciences sociales se ramène à la question suivante : qui sont les personnages centraux du réseau ? Qui sont ceux qui détiennent le pouvoir ? Quels sont ceux qu'il faut immuniser en priorité pour éviter une épidémie, ceux qu'il faut informer en premier pour diffuser une information ou une idée, ceux qu'il faut cibler de préférence dans une campagne publicitaire ? On s'inspire ici des travaux de Kempe, Kleinberg et Tardos [7] qui se basent sur le problème-type suivant : une entreprise veut faire connaître un de ses produits en l'offrant à quelques personnes. Comment les choisir pour optimiser l'effet du bouche-à-oreille ? L'étude menée ici présuppose la connaissance d'un graphe pondéré représentant les liens entre les personnes. Comme souligné plus haut, la connaissance d'un tel graphe avec précision est hypothétique. Il existe néanmoins un endroit où de tels graphes sont accessibles, parfois librement, c'est sur la toile (voir paragraphe sur Facebook). Il serait tout à fait imaginable, quoique immoral et sans doute illégal, d'exploiter cette méthode afin de faire du marketing en ligne.

3.1.1. Formulation du problème

Prenons l'exemple d'une entreprise qui souhaite faire connaître son produit à un maximum de personnes dans une population N, en l'offrant à un échantillon S de personnes, de taille donnée K. Il s'agit pour elle de trouver l'échantillon S qui maximise la taille de l'ensemble des personnes finalement atteintes par le bouche-à-oreille.

Posons alors z la fonction qui à S associe le nombre de personnes finalement informées :

$$z : \begin{cases} \mathcal{P}(\mathcal{N}) & \to \mathbb{N} \\ S & \to \sharp S_\infty \end{cases}$$

Où S_t est l'ensemble des personnes convaincues à t.

Propriétés de la fonction z :
- $z(\emptyset)=0$: si l'on n'envoie d'échantillon à personne, personne n'est informé de l'existence du produit
- z est croissante ($S \subset T \Longrightarrow z(S) \leq z(T)$) : informer initialement des personnes en plus augmente nécessairement le nombre de personnes informées au final.

D'un point de vue mathématique, il s'agit de résoudre le problème :

$$(\mathcal{P}) \quad \max_{S \subset \mathcal{N}} \{z(S); \sharp S \leq K\}$$

Néanmoins, ce problème est NP-Complet[7]. Pour s'en convaincre, il suffit de voir que trouver le maximum de la fonction z nécessite au moins d'examiner les valeurs qu'elle prend sur tous les ensembles de cardinal K, soit au total $\binom{N}{K}$ possibilités.

C'est pourquoi il est naturel de chercher une méthode de recherche approchée du maximum, de complexité moindre.

3.1.2. Algorithme de *hill-climbing*

L'idée de l'algorithme est simple : suivre le sens de la pente. L'algorithme correspondant fonctionne comme un algorithme glouton. Partant de l'ensemble vide, on construit notre ensemble S_g approché en lui adjoignant à chaque étape le sommet permettant la plus grande augmentation de z. Le grand intérêt de cet algorithme réside dans sa complexité : il est linéaire (ou plutôt en $\mathcal{O}(KN)$).

```
hill-climbing(G)
S<-∅ ;
pour i de 1 à K faire
  trouver v* ∉ S tel que z(S ∪ {v*}) = max_{v∉S} z(S ∪ {v}) ;
  S<-S∪{v*} ;
retourner z(S) ;
```

Il est bien évident que cet algorithme ne nous garantit en rien de trouver une solution intéressante au problème. A priori, rien ne nous assure la proximité de la solution approchée à la solution optimale. Il faut pour celà disposer d'hypothèses supplémentaires sur la fonction z. Or, il est légitime de supposer que la fonction z est submodulaire.

Définition : Une fonction $z : \mathcal{P}(\mathcal{N}) \to \mathbb{R}$ est dite ***submodulaire***, si elle vérifie une des deux conditions équivalentes suivantes :

$$\forall S, T \subset \mathcal{N},\ z(S) + z(T) \geq z(S \cup T) + z(S \cap T)$$

$$\forall S \subset T, \forall v \notin T,\ z(S \cup \{v\}) - z(S) \geq z(T \cup \{v\}) - z(S)$$

La deuxième condition s'interprète comme une condition de rendements décroissants (chère aux économistes). Elle est assez intuitive. En effet, plus l'ensemble T est gros, plus le risque est important que l'ajout d'un nouveau sommet v n'apporte pas grand chose, puisque beaucoup des « amis » de v sont déjà dans T. En réalité, une démonstration rigoureuse nécessite auparavant de modéliser le processus de diffusion de l'information, les modèles les plus classiques étant les modèles à seuil et en cascade [7].

Cette nouvelle hypothèse, légitime et naturelle, permet d'obtenir le résultat fort suivant :

Théorème : Pour toute fonction $z : \mathcal{P}(\mathcal{N}) \to \mathbb{N}$ croissante, nulle sur l'ensemble vide, et ***submodulaire***, alors, notant Z la solution optimale au problème $(\mathcal{P})$ et Z_G la solution approchée obtenue par l'algorithme glouton, il vient :

$$\frac{Z_G}{Z} \geqslant 1 - \left(\frac{K-1}{K}\right)^K \geqslant 1 - \frac{1}{e} \simeq 0,63$$

Ainsi, au prix d'hypothèses assez légitimes, il est possible de trouver en un temps très court une bonne approximation de l'ensemble de K personnes le plus influent.

3.1.3. Eléments de démonstration

L'idée est d'exploiter les propriétés de submodularité de la fonction z pour exhiber des contraintes vérifiées par les différentes valeurs prises par la fonction $z(S)$ lors de l'exécution

de l'algorithme glouton. La borne inférieure du théorème s'obtiendra alors par résolution d'un problème d'optimisation sous contraintes. La démonstration présentée ici est une simplification d'une démonstration proposée par G.Nemhauser [8] qui ne suppose ni la croissance ni la nullité à l'origine de la fonction.

Notons par souci de simplification des écritures $\rho_v(S) = z(S \cup \{v\}) - z(S)$ (l'accroissement associé à l'ajout de v). La première condition suivante s'obtient par simple manipulation des inégalités de submodularité.

Proposition 1 : Pour z submodulaire, croissante, nulle sur l'ensemble vide :

$$\forall S, T \; z(T) \leq z(S) + \sum_{v \in T - S} \rho_v(S)$$

L'application de cette inégalité à des ensembles T et S bien choisis permet d'obtenir une majoration de Z, la solution optimale du problème (P). On note ρ_0,ρ_1...ρ_{K-1}les accroissements successifs obtenus lors de l'adjonction d'un nouveau sommet v^* dans l'algorithme glouton. Alors la solution approchée s'écrit $Z_G = \rho_0 + ... + \rho_{K-1}$.

Proposition 2 : Pour z submodulaire, croissante, nulle sur l'ensemble vide :

$$\forall t \in \{0, ..., K-1\} \; Z \leq \sum_{i=0}^{t-1} \rho_i + K\rho_t$$

Posant $x_t = \frac{\rho_t}{Z}$, ces conditions permettent de poser le problème d'optimisation suivant, dont la résolution donne une borne inférieure à $\frac{Z_G}{Z}$:

$$\begin{cases} \min\limits_{x_t} \sum\limits_{t=0}^{K-1} x_t \\ \forall t \in \{0, ..., K-1\} \; 1 \leq \sum\limits_{i=0}^{t-1} x_i + K x_t \end{cases}$$

dont la solution est bien $1 - \left(\frac{K-1}{K}\right)^K$ (la résolution de ce problème d'optimisation peut se faire par passage par le problème dual [9]).

3.1.4. Individus influents et invulnérabilité

Étudions maintenant la relation entre l'influence des individus et leur degré d'invulnérabilité, tel qu'il a été introduit dans le paragraphe sur les communautés d'équivalence. Intuitivement, des individus au cœur des amas de sommets du graphe seront plus influents du fait de leur position centrale. Cependant, il ne faut pas oublier que la recherche des individus influents aspire à déterminer l'ensemble de K individus le plus influent, et non les K individus les plus influents. Il se pourrait donc qu'à cause de sa différence de position, l'adjonction d'un sommet de périphérie soit plus utile que celle d'un autre sommet de cœur. Voyons les résultats des expériences sur le graphe des forums de discussion avec K=5.

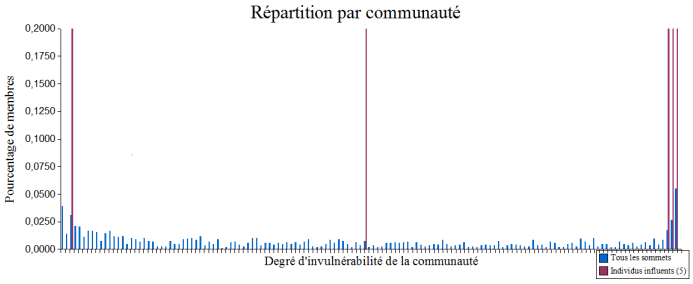


Figure 3.1.: Répartition des individus en fonction de l'invulnérabilité de leur communauté

Tous les individus	273,33
5 individus influents	418,6

Table 3.1.: Degré d'invulnérabilité moyen

Ainsi, cette expérience confirme l'intuition dans la mesure où les individus influents pour $K=5$ ont une forte propension à avoir un degré d'invulnérabilité élevé.

Conclusion Cet algorithme permet donc de trouver l'ensemble le plus influent de K personnes, même sur des graphes très conséquents. Il est à noter que les expériences prouvent que cette méthode est meilleure que toute heuristique simple que l'on pourrait imaginer - il serait par exemple intuitif de vouloir sélectionner les K personnes de degré le plus élevé. Cet algorithme peut être utilisé sur les très grands graphes des réseaux sociaux du web. Il est à espérer, pour le confort de l'internaute, que personne ne cherche à exploiter les éventuelles possiblités marketing ouvertes par ce genre de méthodes.

3.2. Prédiction des liens futurs

Dans un graphe social d'une communauté, il est toujours intéressant de pouvoir prévoir les liens qui vont se former dans le futur proche. Afin d'anticiper la création d'un lien entre deux personnes, la meilleure étude est celle de leurs amis communs. En effet, deux personnes qui ont de nombreux amis en commun ont de plus fortes chances d'être amenées à se connaître que deux personnes qui n'ont aucun contact en commun.

3.2.1. Programme de prédiction des liens futurs

Afin d'automatiser les prédictions, nous avons codé un programme permettant de calculer le score représentant la proximité entre deux sommets, et qui donne l'état futur d'un graphe en créant des liens selon différents calculs de score. Dans un premier temps, nous avons choisi de ne créer qu'un pourcentage de liens parmi ceux qui n'existent pas, en choisissant ceux qui ont le score le plus élevé. Une autre méthode pourrait être de choisir de créer tous les liens dont le score est supérieur à un seuil minimal donné.

3.2.1.1. Scores pour graphe non pondéré

Grâce aux travaux de Liben-Nowell et Kleinberg[12], nous avons pu étudier différentes façons d'aborder la proximité entre deux sommets d'un graphe non pondéré. En effet, ils proposent plusieurs manières de calculer le score existant entre ces deux sommets.

Si on note Γ(i) l'ensemble des voisins de i, sommet du graphe, alors plusieurs possibilités de score peuvent être envisagées en ne considérant que les voisins proches des deux sommets :

- Voisins communs : $score(i,j) = |\Gamma(i) \cap \Gamma(j)|$
- Coefficient de Jaccard :

$$score(i,j) = \frac{|\Gamma(i) \cap \Gamma(j)|}{|\Gamma(i) \cup \Gamma(j)|}$$

- Coefficient d'Adamic et Adar :

$$score(i,j) = \sum_{z \in \Gamma(i) \cap \Gamma(j)} \frac{1}{log\,|\Gamma(z)|}$$

- Attachement préférentiel :

$$score(i,j) = |(i)| \times |(j)|$$

3.2.1.2. Scores pour graphe pondéré

Les travaux de Liben-Nowell et Kleinberg permettent d'étudier les relations futures dans un graphe non pondéré, mais dans le but d'étudier des graphes réels, avec des liens dont l'intensité peut varier, nous avons pris en compte le fait que les graphes peuvent être pondérés. Par conséquent, nous avons dû réfléchir à différentes méthodes pour calculer les scores entre deux sommets :

$$score1(i,j) = \frac{\sum_{z \in \Gamma(i) \cap \Gamma(j)} \sqrt{Poids(i,z) \times Poids(j,z)}}{\sum_{z \in \Gamma(i)} Poids(i,z) + \sum_{z \in \Gamma(j)} Poids(j,z)}$$

$$score2(i,j) = \frac{\sum_{z \in \Gamma(i) \cap \Gamma(j)} Poids(i,z) \times Poids(j,z)}{\sum_{z \in \Gamma(i)} Poids(i,z) + \sum_{z \in \Gamma(j)} Poids(j,z)}$$

$$score3(i,j) = \sum_{z \in \Gamma(i) \cap \Gamma(j)} \sqrt{Poids(i,z) \times Poids(j,z)}$$

En effet, le numérateur a été choisi pour accorder plus d'importance à une relation future si les voisins communs de i et j leur sont fortement liés, et par des poids similaires (on aura un score plus élevé pour des poids de 5 et 5 que pour 1 et 9). Le dénominateur des deux premiers réduit le score si i ou j a beaucoup d'autres amis : il aura moins de temps à passer à faire de nouvelles connaissances. Il n'y a pas de dénominateur dans le troisième, car il peut également y avoir un phénomène tel qu'une personne très connectée se fera rapidement beaucoup d'autres amis.

3.2.2. Vérifications

Afin de vérifier la pertinence des résultats, nous avons testé les programmes sur des graphes obtenus à partir des forums de discussions. En choisissant un graphe initial (par exemple de la période Janvier 08-Mai 08), nous avons observé les différences de liens créés entre le graphe réel avec une période de 2 mois en plus, et le graphe prédit par le programme.

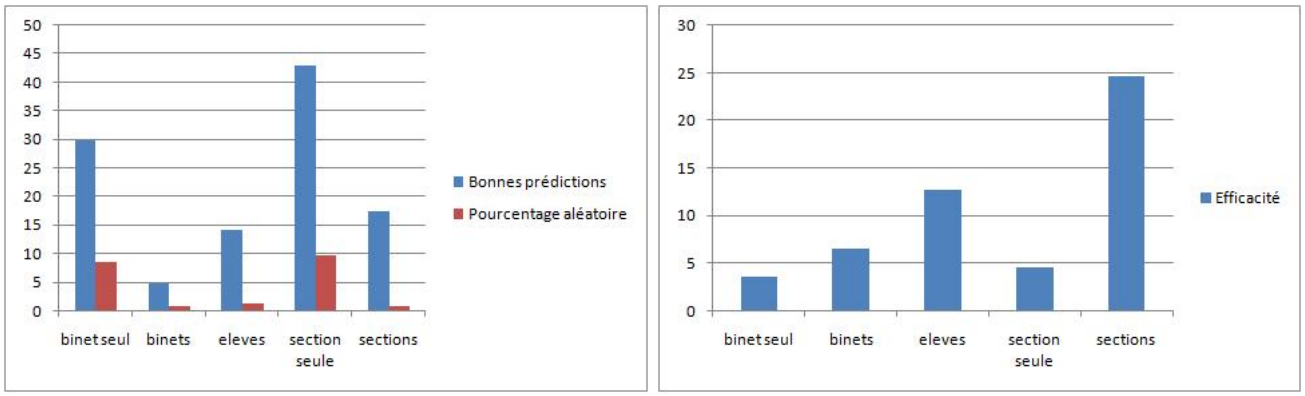
Nous avons ensuite comparé ces résultats à ceux obtenus en créant les liens de manière aléatoire. Pour cela, nous avons lancé une centaine de fois un calcul du graphe prédit, avec un score choisi aléatoirement entre 0 et 1, puis fait la moyenne des pourcentages de bonnes prédictions.

Pour simplifier les notations, nous noterons $P_{predit}(i)$ le pourcentage de bonnes prédictions avec la méthode i, P_{crees} le pourcentage de liens créés parmi ceux qui n'existaient pas, et P_{rand} la moyenne des pourcentages de bonnes prédictions obtenue avec la génération de graphes aléatoires. On notera enfin e l'efficacité, définie par : $e = \frac{P_{predit}(1)}{P_{rand}}$.

Nous avons lancé le programme sur différents types de forums de discussion (section, binet, eleves, etc.) pour effectuer des prédictions suivant différents types de relations que peuvent avoir les posteurs.

Les premiers tests, lancés sur des forums avec peu de posteurs pour pouvoir vérifier l'efficacité du programme, ont été faits sur des forums de binet particuliers. Les tests sur le br.eleves (forum de discussion destiné à tous les élèves, 700 posteurs et 18 000 messages), les forums de binet (65 000 messages parmi 1 100 posteurs) et les forums de section (35 000 messages pour 900 posteurs) n'ont utilisé qu'une seule méthode, pour minimiser les temps de calcul.

Forum	$P_{predit}(1)$	$P_{predit}(2)$	$P_{predit}(3)$	P_{crees}	P_{rand}	e
Binet quelconque	26,7%	28,7%	29,88%	8,28%	8,4%	3,56
Tous les binets	4,59%	Ø	Ø	0,71%	0,71%	6,46
Section quelconque	35,09%	37,39%	42,89%	9,76%	9,54%	4,5
Toutes les sections	17,45%	Ø	Ø	0,72%	0,71%	24,58
Eleves	14,08%	Ø	Ø	1,14%	1,11%	12,68



(a) Meilleur pourcentage de prédiction par type de forum

(b) Efficacité de la prédiction par type de forum

Figure 3.2.: Evaluation des méthodes de prédiction par type de forum

Si on se fie simplement au pourcentage de bonnes prédictions, alors les forums avec le moins d'utilisateurs sont les plus prédictibles (jusqu'à 43% de bonnes prédictions). Cependant, en comparant $P_{predit}(i)$ et P_{rand}, on a une bonne évaluation du rapport entre une prédiction faite avec des scores, et celle faite au hasard. De cette manière, nous pouvons observer un meilleur rapport sur le br.eleves (12,7 fois plus de bonnes prédictions) que sur les forums des binets (6,5 fois plus) et les forums avec peu d'utilisateurs (entre 3,5 fois et 4,5 fois plus). Le forum présentant le meilleur résultat selon ce critère est celui de tous les forums de sections réunis, car la méthode permet de prévoir 24,6 fois plus de créations de liens. En effet, les élèves des sections vivent en communauté et se côtoient plus souvent, donc créent de vraies relations qui se répercutent sur les forums de discussion.

Une autre observation intéressante est celle des pourcentages obtenus sur les forums de binet et de section particuliers : le pourcentage de bonnes prédictions était meilleur avec

le troisième score qu'avec le premier. Cela montre qu'une personne déjà très connectée ne sera pas handicapée par le temps passé avec ses contacts pour se faire de nouvelles connaissances.

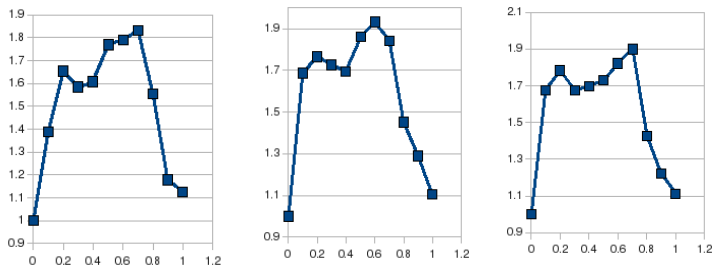
Les autres méthodes non pondérées données par Liben-Nowell et Kleinberg ont été appliquées au calcul des prédictions. Etonnemment, à part le score de Jaccard qui renvoyait 2 fois moins de bonnes prédictions qu'une prédiction aléatoire (ce qui montre à nouveau qu'une personne très connectée n'a pas nécessairement des scores plus faibles), tous les scores renvoyaient un nombre de bonnes prédictions du même ordre de grandeur.

3.3. Liens entre structures statiques et dynamiques

Il paraît naturel de vouloir étudier l'influence conjointe des structures statiques et dynamiques. C'est pourquoi nous avons testé sur des graphes simulés l'indépendance des liens latents créés et l'appartenance de ces liens à des communautés définies par la méthode des chaînes de Markov. Nous avons pour cela défini un test

$$\hat{p} = \frac{\sharp\{\text{liens intracommunautaires créés}\}}{\sharp\{\text{liens intracommunautaires manquants}\}} \cdot \frac{\sharp\{\text{liens intercommunautaires manquants}\}}{\sharp\{\text{liens intercommunautaires créés}\}}$$

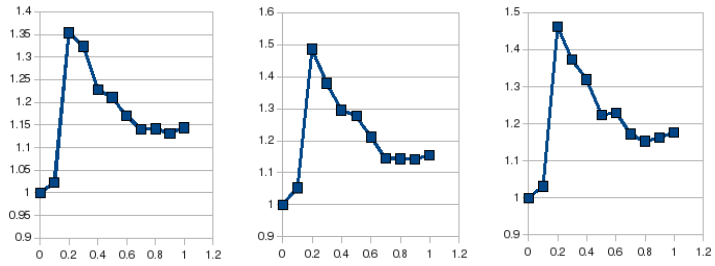
Si les deux évènements sont indépendants, alors $\hat{p} \to 1$. Nous avons donc effectué plusieurs tests ce qui donne les résultats figures 3.3 et 3.4.



(a) Méthode : voisins communs (b) Méthode : Jaccard (c) Méthode : Adamic et Adar

FIG. 3.3.: Test de $\hat{p}$ sur des graphes communautaires (6 communautés de 10 personnes) en fonction de la probabilité d'existence de liens intercommunautaires

Sur ces figures, l'indice est nettement supérieur à 1, même dans le cas surprenant des graphes communautaires inversés (probabilité intercommunautaire > probabilité intracommunautaire). Cela est dû au fait que d'autres communautés sont créées par rapport à celles qui étaient prévues. De plus, on remarque que la variation de l'indice dépend du graphe considéré mais pas de la méthode (profils de courbes semblables) : ces résultats ne permettent pas de distinguer par exemple le score d'Adamic et Adar des autres coefficients. En clair, la prévision de liens latents est nettement corrélée à la structure de communautés.



(a) Méthode : voisins communs (b) Méthode : Jaccard (c) Méthode : Adamic et Adar

FIG. 3.4.: Test de $\hat{p}$ sur des graphes type Watts et Strogatz (200 sommets, huit voisins) en fonction de la probabilité d'existence de liens intercommunautaires

3.4. Analyse de la structure locale des réseaux sociaux

3.4.1. Recherche de motifs

3.4.1.1. Introduction à la classification des graphes réels

Les graphes réels présentent des propriétés communes que nous avons déjà mentionnées (effet petit monde, clustering, distribution) et les différents raffinements portés sur les graphes aléatoires ont conduit à des modèles qui capturent plus ou moins bien ces paramètres. Cependant l'ensemble des graphes réels n'est pas une classe homogène, au contraire, plusieurs travaux ont permis de montrer que l'on pouvait regrouper ceux-ci en «super familles» aux propriétés bien spécifiques. Une classification intéressante découle de la recherche de motifs-réseaux [17]: il a été remarqué que des graphes provenant des disciplines différentes possédaient certains motifs dont la répartition s'éloignait de facon importante de l'un à l'autre, de même qu'ils étaient en nombre beaucoup plus important que ceux qu'on aurait pu attendre à partir des modèles aléatoires. L'intuition ici est que certains de ces motifs ont une importance cruciale dans le fonctionnement du réseau et capturent en quelque sorte l'essence de la mécanique interne du graphe, on pensera à l'exemple d'un réseau biologique où la production d'une protéine qui met en jeux des cascades de réactions chimiques bien déterminées.

Nous nous apprêtons donc dans cette partie à différencier nos forums de discussion à partir de la méthode de recherche de motifs-réseaux comme decrit dans [17] à partir de nos propres algorithmes.

3.4.1.2. Démarche et algorithmique

Motifs Il s'agit tout d'abord de décrire quels motifs intéresseront nos programmes. Nous avons vu que le coefficient de clustering faisait le rapport entre le nombre de triangles (que l'on appellera triades par la suite) entourant un nœud et le degré de celui-ci pour capturer la complexité du réseau. Ces triangles, qui sont de différents types, codent une information plus précise qui peut révéler les mécanismes sous-jacents du réseau :

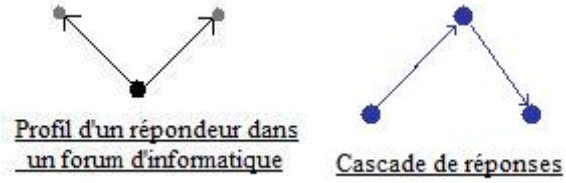


Figure 3.5.: Différents types de structures

Désormais quand on parlera de triades de type i on fera référence au sous-graphe orienté à trois sommets décrit dans la liste suivante :

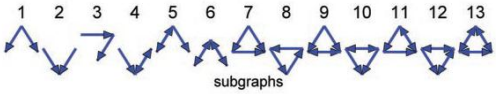


Figure 3.6.: Types de Triades

Le graphe étant pondéré on adoptera une méthode arbitraire de dénombrement inspirée de l'intuition suivante : nous prenons un crayon papier et nous faisons n traits légers si l'arête entre x et y si (x, y) est présente n fois. L'œil humain distinguera d'abord la configuration la plus foncée, ensuite il verra le dégradé qui laisse la deuxième forme la plus représentée, et ainsi de suite. Par exemple pour la configuration suivante nous comptons comme suit:

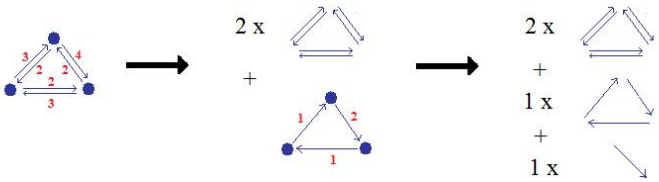


Figure 3.7.: Comment compter dans un graphe pondéré

Molloy et Reed Nous utiliserons pour le calcul du vecteur *Significance Profile* défini dans la section qui suit le modèle de Molloy et Reed. C'est un modèle introduit par les auteurs homonymes qui permet de créer un graphe avec une taille et une distribution de degrés données. Pour le construire nous procédons comme suit :

1. On construit le nombre de nœuds voulus.
2. On met sur chaque nœud des «demi-arêtes» en nombre fixé par la distribution du graphe réel (pour des graphes orientés il faudra prendre la séquence des degrés entrants et sortants).
3. On tire deux demi-arêtes au hasard que l'on rejoint donnant ainsi une arête.

Significance Profile Pour éviter des effets de degré et de taille nous définissons un coefficient pour chaque type de motif qui représentera son importance intrinsèque dans le graphe. Ce type de calcul est robuste à l'élimination et l'ajout d'arêtes au hasard.

Nous écrivons Z_i le nombre de motifs de type i retrouvés, $< Z_i >$et std_i la moyenne et l'écart-type d'un tirage de graphes Molloy et Reed de même degré et distribution que le graphe réel. Nous définissons donc le *Significance Profile* pour le motif de type i :

$$SP_i = \frac{Z_i - < Z_i >}{std_i}$$

3.4.1.3. Exploitation et résultats

Forums choisis Les forums de discussion choisis sont :

1. Bâtiment : forum utilisé par les personnes habitant dans un même bâtiment
2. Binet : forum très grand contenant les différents forums de chaque binet
3. Communauté : héberge les différentes communautés religieuses et culturelles à l'école
4. Enseignement : autour des questions sur les études
5. Informatique : pour les diverses questions sur le sujet
6. Élèves : forum qui héberge les sujets qui sont d'un intêret général pour l'ensemble des élèves de l'école
7. Section
8. PA : «petites annonces» (besoin d'une voiture, vente, achat, etc...)

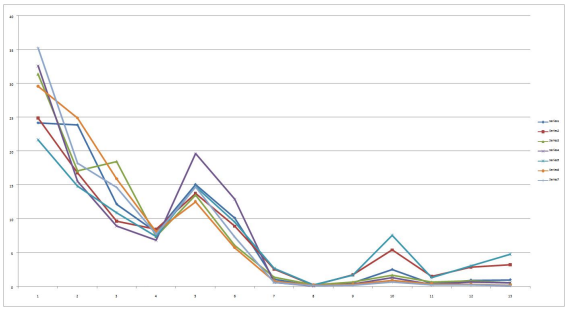
Certains de ces forums ont été choisis par leur taille (Binet), d'autres par les particularités de leur structure (PA).

Résultats

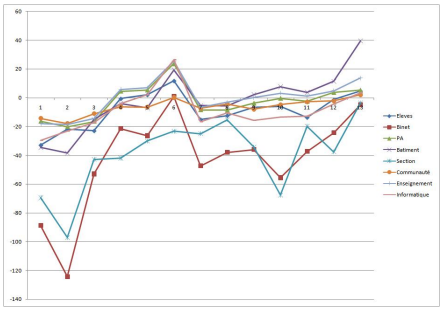
type	Binets	communauté	élèves	informatique	section	enseignement	pa	Bâtiment
1	59155	1135	22448	11850	14430	2866	3795	1627
2	58409	765	12216	5637	9857	2413	1961	977
3	29789	441	13227	3243	7236	1542	1580	835
4	19755	387	5286	2472	4912	785	831	421
5	36879	628	9658	7128	9835	1215	1583	252
6	24748	407	4337	4688	6324	556	788	682
7	2747	117	984	218	1785	80	65	301
8	201	9	180	15	140	21	14	39
9	1320	79	480	119	1078	32	19	114
10	6112	247	1214	460	5025	87	69	270
11	1125	70	482	99	867	39	27	113
12	2306	132	593	233	2000	31	29	107
13	2417	148	401	196	3164	20	13	62

Table 3.2.: Dénombrement des Triades

Cette table révèle la taille de chacun des binets. La croissance du nombre de motifs retrouvés augmente de manière importante avec la taille du forum. Une approche plus rapide consisterait à voir les pourcentages relatifs de présence des Triades :

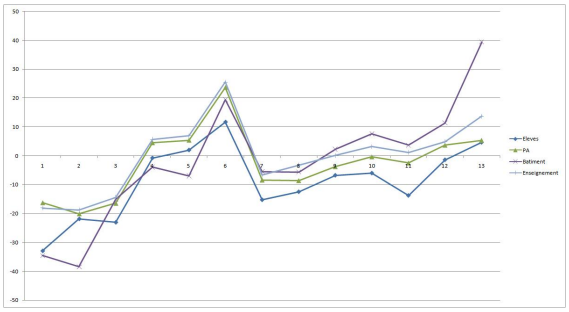


(a) Graphique des pourcentages de représentativité des Triades

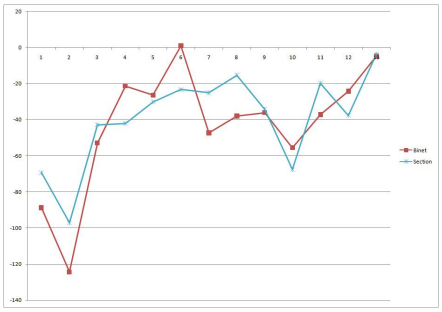


(b) *Significance profile*

Analyse Tout d'abord le graphique des *Significance Profile* (SP) nous permet de distinguer de premier abord deux grandes familles de forums, ceux qui sont bien centrés par rapport au tirage de Molloy et Reed (qui bien sûr dans notre graphique correspondrait à la droite $y = 0$) et des graphes qui sont bien en dessous de l'axe des abscisses. Voyons un peu plus clair au sein de ces familles :



(c) Première famille



(d) Deuxième famille

Nous n'avons pas retenu le forum des communautés, qui lui est beaucoup trop «plat» et ne suit donc pas les mêmes variations que cette famille. La ressemblance est remarquable entre PA et Enseignement et ce fait était déjà auguré par le graphique des pourcentages. Le forum des éleves est aussi assez proche de ces derniers, bien qu'un peu en-dessous, et on voit que le forum des bâtiments présente des pics assez importants pour le type 13 et le type 2. Un type 13 est révelateur d'une forte interaction au sein d'un même groupe, on pourrait donc parler d'une complicité qui est maximale pour les personnes qui vivent dans un même bâtiment. Comment expliquer la proximité du forum PA et Enseignement ? Une petite révision des conversations qui ont lieu dans ces deux serveurs permet de se rendre compte que souvent au forum d'enseignement on passe une petite annonce de besoin d'aide (contenu des partiels, date limite de depôt d'une candidature et pourquoi pas «qui à la réponse de *?»). On comprend donc des mécanismes de fond par des analyses globales.

Le rapprochement de ces deux forums n'est ni étonnant ni évident. On pourra se dire qu'aux binets comme dans les sections on partage une passion et que dans un binet se crée

une cohésion comme dans les sections. Cependant la multitude des binets et la présence de quelques binets inutiles et inactifs aurait pu créer une trop grande hétérogéneité pour se rapprocher des sections. Un autre fait intéressant est le manque de Triades de type inférieur à 5, celles-ci sont celles des échanges entre deux participants uniquement : on voit donc bien que les binets sont des espaces de partage, et que les sections sont intégrées dans leur généralité.

Pour finir, nous voudrions remarquer que les SP des forums ne sont pas fondamentalement différents, en fait cette méthode a le plus souvent été utilisée pour distinguer des familles plus globales (linguistiques, biologiques, web, sociales). Tout de même, l'outil nous a montré des différences qui au premier abord étaient difficiles à distinguer et a permis de lever un brouillard provoqué par les effets de taille et de degré qui font varier l'apparition de chaque type de Triade de façon indépendante.

3.4.2. Recherche d'une structure type

3.4.2.1. Idée

Nous avons déjà mentionné que certaines structures locales avaient un «sens». Cette fois-ci nous nous plongerons dans le contenu d'un message d'un certain type de motif donné, pour trouver une corrélation entre les deux ; les applications d'une telle méthode sont très intéressantes pour les réseaux actuels : détection de spams, classement des personnes qui coopèrent le plus dans un site, localisation de phénomènes dans un réseau biologique.

Une structure assez commune dans les fils des forums de discussion est une arborescence très fine et longue, souvent consituée d'un seul chemin, et qui dans la plupart des cas correspond à un remerciement des élèves envers d'autres élèves ou un binet ou un événement. Une explication rapide de ce phénomène consiste à dire que dans un remerciement on ne se soucie pas de ce que les autres ont écrit, on ne répond pas à une question et on ne s'adresse pas à quelqu'un, et donc on répond de la façon la plus facile qui est de répondre à la dernière personne qui a posté (bien sur avec quelques bifurcations du chemin unique un peu aléatoires).

Le coefficient de vraisemblance d'un fil donné avec le genre de structure que nous cherchons sera codé simplement par la formule suivante, qui vaut 1 si l'arbre a une seule branche :

$$M = p/n$$

où p est la profondeur de l'arbre et n est le nombre de nœuds qu'il possède. On remarquera qu'un M important impose une grande taille de l'arborescence.

3.4.2.2. Résultats et Exploitation

On n'a cherché les fils que dans 5 de nos forums de discussions (un remerciement n'a pas sa place au forum PA par exemple) ce qui a donné les résultas suivants :

	Bâtiment	Binet	Élèves	Section	Communauté
Nombre de messages	4	10	0	5	1
Anniversaire	3	0	0	2	0
Sport	0	0	0	2	0
Activité	0	10	0	1	1
Autre	1	0	0	0	0

Table 3.3.: Types de messages rencontrés pour M>0.9

Au forum des élèves on ne retrouve aucun de ces messages alors que c'était notre candidat numéro 1 pour les trouver : c'est au forum des élèves où nous avons vu le plus souvent ce genre de remerciements.

Nous voyons alors que deux types de sujets sont les plus récurrents, souhaiter un joyeux anniversaire et organiser une activité. En ayant vu de plus près ces messages nous retrouvons une structure familière à laquelle nous n'avions pas pensé au début : quand une activité est organisée, les élèves ont pour habitude de dire très rapidement s'ils sont pour ou contre en écrivant «+1», par exemple pour pratiquer leur sport collectif ou pour organiser un événement.

Avons nous été trop exigeants sur la largeur de l'arbre pour les remerciements ?

On a relancé notre algorithme avec un paramètre M dont le seuil inférieur était 0.8 pour le forum des sections. Nous trouvons 64 messages et une composition beaucoup plus hétérogène. Les remerciements aparaissent mais ne sont pas vraiment représentatifs, ne constituant que 10 % de l'échantillon. Pour les remerciements et l'assistance aux activités nous en trouvons beaucoup moins aussi, avec 10 % de chaque aussi.

Bien que notre première intuition se soit avérée fausse nous pouvons en conclure que nous avons trouvé une bonne démarche pour localiser les anniversaires et l'organisation d'activités. Cette dernière semble bien sûr plus pratique, car nous pouvons d'un point de vue sociologique voir facilement comment s'organisent les élèves, quels sont leurs intérêts, et avec quelle fréquence ils organisent des activités. Un autre point positif de ces résultats est le fait que nous ayons bien pu faire un lien entre structure et sémantique des messages, et peut être, quitte à affiner notre modèle, pourrons-nous trouver les remerciements dans les forums de discussion. Dans grand nombre d'articles les chercheurs ont pour but de révéler ce genre de rapports, qui permettent d'une part d'améliorer la compréhension de phénomènes que nous observons à une échelle plus macroscopique et d'autre part de créer des méthodes rapides pour la classification d'un certain réseau.

Troisième partie .

Travail en équipe

Une démarche de recherche

Nous pensons qu'il existe deux principales voies pour mener ce genre de projet collectif. La première consiste à se fixer dès le départ, en octobre, un objectif concret et à orienter le travail vers sa réalisation. Cette méthode comporte son lot de risque : car l'objectif peut être mal évalué et mal calibré pour le format PSC. Surtout, il se prête mal à un travail en mathématiques. Nous avons préféré opter pour une seconde voie : partir d'un intérêt et d'une envie commune pour travailler sur un sujet, en l'occurrence la théorie des graphes, pour nous spécialiser au fur et à mesure. Selon nous, la première démarche peut-être qualifiée de **démarche-projet**, tandis que notre choix s'est porté sur une **démarche-recherche**. L'idée est de se familiariser et d'approfondir ses connaissances dans un domaine avant d'identifier un ou plusieurs problèmes motivants où nous pouvons apporter notre contribution.

Ainsi, nous avons entrepris au départ d'approfondir légèrement quelques domaines où la théorie des graphes a des applications intéressantes. Un compte-rendu détaillé de ces diverses pistes se trouve dans le **dossier de présentation** remis en décembre. Dès novembre, notre choix s'était porté sur l'analyse des réseaux sociaux. Les mois d'hiver ont été consacrés en grande partie à un travail bibliographique pour se mettre à niveau dans ce domaine tout à fait nouveau pour nous qu'est l'analyse des réseaux sociaux. Le lecteur trouvera une trace des ces recherches dans le **livret intermédiaire** rendu début février. L'évolution des tâches menées par chacun depuis le départ reflète bien ce processus de convergence vers un sujet de plus en plus spécialisé.

Répartition des tâches

(e) 1ère phase : détermination du domaine

(f) 2ème phase : Mise au goût de l'état de l'art

(g) 3ème phase : Approfondissement

FIG. 3.8.: Evolution de la répartition des tâches dans le temps

Nous avons pu mesurer à quel point la répartition des tâches est une **étape difficile mais essentielle** du travail d'équipe. Entre chacune des phases présentées ci-dessus, notre projet a souffert d'une courte période de flottement, due principalement à l'imprécision de la définition de la mission de chacun. Dans ces moments, chacun a tendance à croire que le reste du groupe se chargera de faire avancer le projet.

Nous avons pu juger aussi combien il est important d'entretenir la **motivation** et l'engagement personnel de chacun dans le projet. En effet, dans une organisation hiérarchique, il incombe au cadre d'effectuer cette attribution des rôles. Dans un projet collectif comme

le nôtre, la répartition des tâches procède plutôt de l'initiative de chacun et de son engagement auprès des autres. Dans un groupe nourri par la **confiance mutuelle**, chacun s'occupe de prospecter les développements possibles, et, lors des réunion hebdomadaires, se fait force de proposition et s'engage à mener ses idées à terme.

Notre but n'est bien sûr pas de prétendre que l'organisation interne du groupe fut parfaite, et que nous avons su dès le départ trouver les recettes d'une collaboration harmonieuse. Nous avons tous en souvenir des réunions hebdomadaires infructueuses, ou des tensions contenues quand l'un ou l'autre n'effectue pas les tâches attendues. Ce projet, c'est donc avant tout l'**apprentissage par la pratique** du travail collectif.

Coordination et communication interne

Les conditions du Projet Scientifique Collectif nous ont imposé des contraintes bien particulières. En effet, aucun de nos cinq emplois du temps ne coincidait avec celui d'un autre. Y compris les soirs après les cours, il était très difficile de dégager un créneau de travail commun. Seul reste le créneau du Lundi matin, précieux mais malheureusement insuffisant. Surtout que le rythme de travail d'un tel projet n'est pas **linéaire**. En effet, malgré notre volonté de répartir le travail sur toute l'année, il existe toujours des périodes (examens notament) où le projet n'est plus au premier plan.

Dès lors, il est inévitable qu'une grande partie du travail soit accomplie individuellement hors de ces créneaux. Le principal défi réside alors dans la **coordination** des avancées de chacun. Nous nous sommes vite rendu compte qu'il était indispensable d'instaurer entre nous le **réflexe du compte-rendu**. Ainsi, toute lecture d'un article ou ouvrage scientifique donnait lieu à un résumé systématique aux autres, tout programme informatique réalisé était envoyé et expliqué en détail aux autres, toute expérimentation conclusive faisait l'objet d'un compte-rendu. Ces comptes-rendus ont d'ailleurs constitué des ingrédients précieux au moment de rédiger les rapports intermédiaires et finaux.

La coordination des travaux s'est jouée de façon cruciale dans l'élaboration des programmes informatiques. En effet, toutes les parties de notre travail ont donné lieu à des expérimentations numériques. Il était indispensable que ces programmes soient pensés conjointement et compatibles entre eux. En effet, les données recueillies par l'un sur le réseau sur les binets et les forums de discussion, ou par l'autre sur Facebook, doivent être utilisables par chaque membre du groupe pour ses applications. Et les résultats obtenus par une méthode doivent pouvoir être comparés à ceux donnés par une autre : c'est ainsi que l'on peut détecter, par exemple, si la prévision des liens futurs est corrélée ou non à l'existence de communautés dans le réseau. Pour assurer la **compatibilité** des programmes, un membre de groupe s'est chargé d'écrire sur Java quelques classes codant la structure de graphe social, accompagnées des méthodes essentielles, qui ont servi de base à tous. Néanmoins, cet objectif n'a été que partiellement atteint, puisque certains typages différents ont pu entraver le partage direct de nos programmes. Avec l'expérience que nous avons acquise et les solutions que nous avons rencontrées, nous savons désormais que nous pourrions utiliser une solution de type SVN qui permet une mise à jour commune et instantanée des programmes. Au final, notre projet, ce n'est pas moins de **8500 lignes de code**. Quand on sait à quel point la programmation et la simulation numérique sont des activités chronophages (écriture, débogage, ajustement des paramètres, temps de calcul qui se comptent parfois en heures), on mesure que tout ce travail constitue indéniablement la **partie immergée de l'iceberg**.

Conclusion

Il est l'heure maintenant de porter un regard rétrospectif sur ces huit mois de travail collectif. Quand nous avons choisi de nous lancer ensemble dans l'aventure, nous ne soupçonnions pas l'existence du domaine de recherche d'analyse mathématique des réseaux sociaux. Ce projet aura été l'occasion d'une **grande découverte** pour nous. Finalement, l'actualité nous a rattrapés, et l'utilisation des réseaux sociaux en epidémiologie met en lumière la qualité des méthodes développées par les chercheurs. Un point étonnant est que l'étude des réseaux n'est que **très récente** et qu'elle voit une explosion depuis les années 2000. Cette contemporanéité fut un atout pour une PSC, puisque le bagage nécessaire pour entrer dans le vif du sujet était à notre portée.

Si l'approche très bibliographique que nous avons adoptée initialement pouvait laisser un doute sur la touche personnelle que nous pourrions donner à notre projet, il ressort finalement que nous avons su peu à peu nous approprier certains aspects de ce domaine de recherche, et adapter des méthodes existantes à nos problèmes, voire en développer de nouvelles. Ce fut fait avec plus ou moins de bonheur, mais les résultats furent parfois inatendus. Ainsi, il était assez imprévisible que l'étude de vitesse de convergence des chaînes de Markov nous amène à développer notre propre méthode d'affichage des graphes. Notre travail fut donc à l'**image de la progression de la recherche** : il est impossible de savoir d'où vont surgir les prochains développements fondamentaux.

Domaine au confluent des sciences sociales, de la physique statistique, des mathématiques et de l'informatique, cette étude fut pour nous un véritable **projet multidisciplinaire à dominante mathématique**. Nous nous sommes ainsi enrichis d'une culture en sciences sociales, de méthodes d'analyse de problèmes transposables à d'autres domaines scientifiques, sans oublier l'expérience d'un travail de groupe de longue haleine.

Bibliographie

[1] Marc Barthélémy, *H1N1 : « Le partage des antiviraux atténuerait la pandémie »*, article du Monde, 30/04/09

[2] Russell Lyons, Yuval Peres *Probability on trees and networks,* 2008

[3] James Moody *Peer influence groups : identifying dense clusters in large networks,* Social Networks n°23, 2001

[4] Nicolas Boulanger, Vincent Vargas *Etude de la vitesse de convergence des chaînes de Markov*

[5] Lionel Tabourier *Analyse statistique de la périphérie des graphes de réseaux sociaux,* rapport de stage de M2 en Physique, 2006

[6] Jerry Grossman, Patrick Ion, Rodrigo de Castro *The Erdös Number Project* (http ://www.oakland.edu/enp/index.html)

[7] D.Kempe, J.Kleinberg, E.Tardos, *Maximizing the spread of influence through a social network,* Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, 2003

[8] G. Nemhauser, L. Wolsey, M. Fisher. *An analysis of the approximations for maximizing submodular set functions.* Mathematical Programming, 14(1978), 265.294.

[9] Michel Bierlaire, *Introduction à l'optimisation différentiable, presses polytechniques et universitaires romandes*, p.109-123

[10] Rick Durrett *Random Graph Dynamics, 2007*

[11] A. Barrat, M. Weigt *On the properties of small-world network models, The European Physical Journal, 2000*

[12] d.Liben-Nowell, Jon Kleinberg, *The link prediction problem for social network*, 2004

[13] P.Pons, *Détection de communautés dans les grands graphes de terrain*, thèse de doctorat de l'Université Paris VII-Diderot, juillet 2007

[14] J. Hopcroft, O. Khan, B. Kulis, B. Selman, *Natural communities in Large Linked Networks*, SIGKDD, 2003

[15] AK Jain, MN Murty, PJ Flynn, *Data clustering : A review*, ACM Computing Surveys, Vol.31,n°3, septembre 1999

[16] P. Pons, *Détection de structures de communautés dans les grands réseaux*, LIAFA

[17] Ron Milo, Shalev Itzkovitz, *Superfamilies of Evolved and Designed Networks*, Science, mars 2004

Annexe

Introduction aux graphes

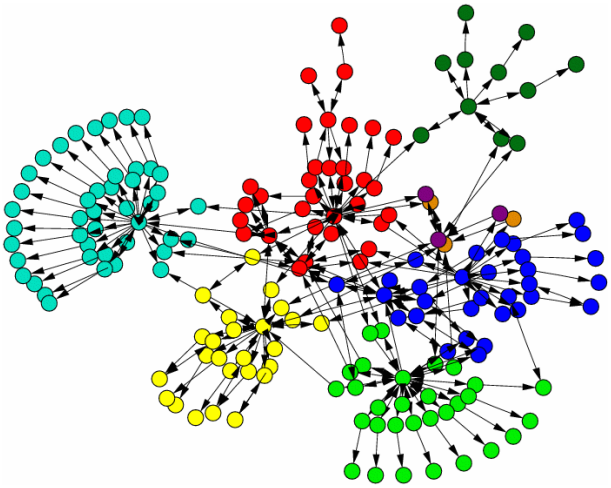


FIG. 1 – Un exemple de graphe orienté

Conceptuellement intuitifs, les graphes constituent un sujet riche qui nourrit la recherche moderne et un champ dont les applications couvrent une gamme croissante de disciplines. Notre volonté dans cette annexe n'est ni de mettre en place un formalisme inutile ni de donner une liste exhaustive des résultats de théorie des graphes que nous avons pu utiliser dans notre projet. Il s'agit avant tout de poser les définitions *ad hoc* aux applications que nous avons faites aux réseaux sociaux.

Définition

Définition Un graphe $G = (S, A)$ est un ensemble de sommets S dont certains sont reliés par une arête de A.

Remarque : Une arête reliant un sommet x à un sommet y est codée par le couple (x, y).

Orientation, pondération

Il existe plusieurs types de graphes pour représenter les différentes situations dans lesquelles on les utilise. Une premiére différence entre ceux-ci est l'orientation. On peut vouloir que le graphe contienne une information sur l' « ordre » selon lequel sont reliées les arêtes. Par exemple un GPS doit bien savoir qu'une route à sens unique ne devrait pas être prise en sens inverse. Par défaut dans notre construction un graphe est orienté c'est à dire qu'on lit (x, y) comme "arête allant de x vers y" et on la représente par une flèche. Ne pas se soucier du sens pour le conducteur revient à savoir que la route va dans les deux sens : c'est à dire $(x, y) \in A$ et $(y, x) \in A$

Définition Un graphe est dit non orienté si la relation A est symétrique c.a.d $(x, y) \in A \Rightarrow (y, x) \in A$

*Remarque*s :

a) On représente un graphe non orienté en dessinant une arête par un trait au lieu d'une flèche

b) Cette definition nous permet de coder les graphes par des matrices symétriques.

On peut vouloir associer un poids à chaque arête, dans un réseau routier l'arête peut porter la distance séparant deux villes.

Définition Un *graphe pondére* est un graphe $G = (S, A)$ qui à chaque arête associe un poids donné par une fonction $p : A \longrightarrow \mathbb{R}$.

Définition Un *graphe biparti* est un graphe $G = (S, A)$ pour lequel l'ensemble des sommets se partitionne en deux sous-ensembles S_1et S_2 sans arêtes internes : $(x, y) \in A \Rightarrow (x \in S_1 \, ET \, y \in S_2) \, OU \, (y \in S_1 ET \, x \in S_2)$

En reprenant notre exemple, nous aurions dans un sous-ensemble les conducteurs, et dans l'autre les modèles de voiture qu'ils conduisent.

Définition La *projection* d'un graphe biparti $G = (S_1 \cup S_2, A)$ sur S_1est le graphe (S_1, A_1) dont les arêtes relient les sommets de S_1liés à un même sommet de S_2 : $(x, y) \in A_1 \Leftrightarrow \exists s \in S_1 | \, (x, s), (y, s) \in A$

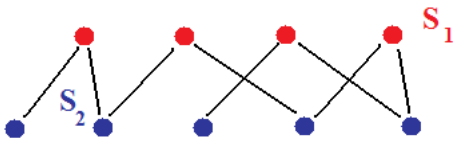


FIG. 2 – Graphe biparti

Projeter notre graphe sur l'ensemble des conducteurs correspond à relier les conducteurs possédant le même modèle de voiture.

Définition Une *clique* est un sous-graphe complet, ie un ensemble de sommets deux à deux connectés.

Obtention du graphe des forums de discussion

Les forums sont archivés sous forme de dossiers contenant des dossiers de sous-forums ou des fichiers codant les messages, ou bien les deux.

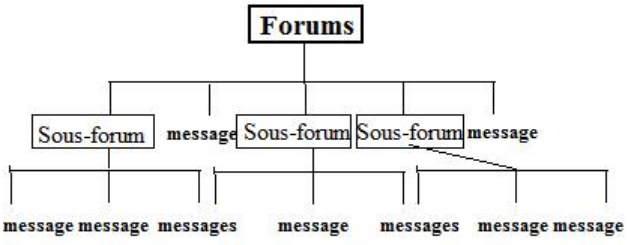


Figure 3: Arborescence des archives des forums

Les fichiers de message sont tous bâtis sur une structure similaire. Des mots-clés en tête de ligne indiquent le contenu de la ligne (expéditeur du message, date d'émission, identifiant du message, identifiant du fil de discussion...).

Pour reconstituer l'arborescence des messages, il a donc fallu parcourir tous ces messages, repérer les lignes intéressantes et établir un tableau de correspondance, codé sous forme de `HashMap`, entre les identifiants des messages et leurs émetteurs. Ensuite, selon le type de liens qu'on souhaite représenter, on stocke les identifiants des messages d'un fil de discussion dans une même liste (*type de liens 1* : tous les interlocuteurs intervenant dans un fil de discussion sont connectés), ou bien les couples messages-réponse (*type de liens 2* : une arête pointe de la personne qui répond vers celle à laquelle il répond), ou encore, pour chaque message inaugurant un nouveau fil de discussion, les personnes qui

prennent part à ce fil (*type de liens 3* : un lien pointe d'une personne qui répond vers la personne qui a ouvert le fil de discussion).

Algorithmes de recherche de communautés

Pour tous ces algorithmes, nous avons opéré avec une classe `Communauté`, contenant essentiellement la liste des identifiants de ses membres et les méthodes qui permettent de la manipuler (appartenance, recherche des voisins, ...) La partition en communautés est stockée dans une classe `Ensemble_communautes`, qui permet notamment d'ajouter ou fusionner des communautés ou bien encore de calculer le coefficient de qualité permettant de comparer deux partitions par identification des communautés les plus apparentées.

A chaque itération de la procédure, on choisit celle des trois possibilités suivantes qui maximise la modularité :

- Création d'une communauté à deux éléments, à partir des sommets isolés du graphe les plus liés
- Ajout d'un membre isolé à une communauté. Pour chaque communauté, tous les membres isolés liés à au moins une certaine fraction de la communauté sont testés. C'est ces nombreux tests qui sont responsables de la complexité temporelle de l'algorithme.

Actualisation de la modularité en temps constant

La maximisation de la modularité $Q = \sum_{Communauté i} e_{ii} - a_i^2$, où e_{ij} est la fraction des arêtes du graphe allant de la communauté i à la communauté j, et $a_i = \sum_j e_{ij}$, sert souvent de critère d'arrêt dans les méthodes de recherche de communautés. Il est donc utile, voire nécessaire, de pouvoir actualiser sa valeur à chaque étape de l'algorithme sans avoir à la recalculer *ab initio*. Pour ce faire, dans le cas d'une fusion entre les communautés 1 et 2 en une communauté 3, on remarque que : $e_{33} = e_{11} + e_{22} + e_{12}$ et $a_3 = a_1 + a_2 - e_{12}$. Par conséquent, $\delta Q_{fusion} = e_{12}[1 + 2(a_1 + a_2)] - e_{12}^2 - 2a_1a_2$. Bien entendu, les valeurs de $a_i$sont stockées pour chaque communauté.

Recherche des plus courts chemins

L'algorithme utilisé est celui de Roy et Warshall. La démarche est la suivante. On initialise le graphe $l^{(0)}$des longueurs des plus courts chemins en remplissant la case *(i,j)* par $l_{ij}^{(0)} = 1$ s'il existe un chemin de i à j, par $l_{ij}^{(0)} = \infty$ sinon. A la r-ième étape de l'algorithme, on n'autorise les plus courts chemins à ne passer que par les sommets entre *1* et r, et on actualise le graphe en remarquant que $l_{ij}^{(r+1)} = Min(l_{ij}^{(r)}, l_{ir}^{(r)} + l_{rj}^{(r)})$. Quand r=n, on a le graphe des longueurs des plus courts chemins. La complexité temporelle est donc en $O(n^3)$.