Thèse LaTeX

Y. Slimani

8 décembre 2018

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Farhat Abbas - Sétif 1 -

THESE

Présentée à la Faculté des Sciences

Département d'Informatique
Pour l'Obtention du Diplôme de

DOCTORAT EN SCIENCES

Option : Informatique

Thème

Extraction et analyse de connaissances à partir du Web

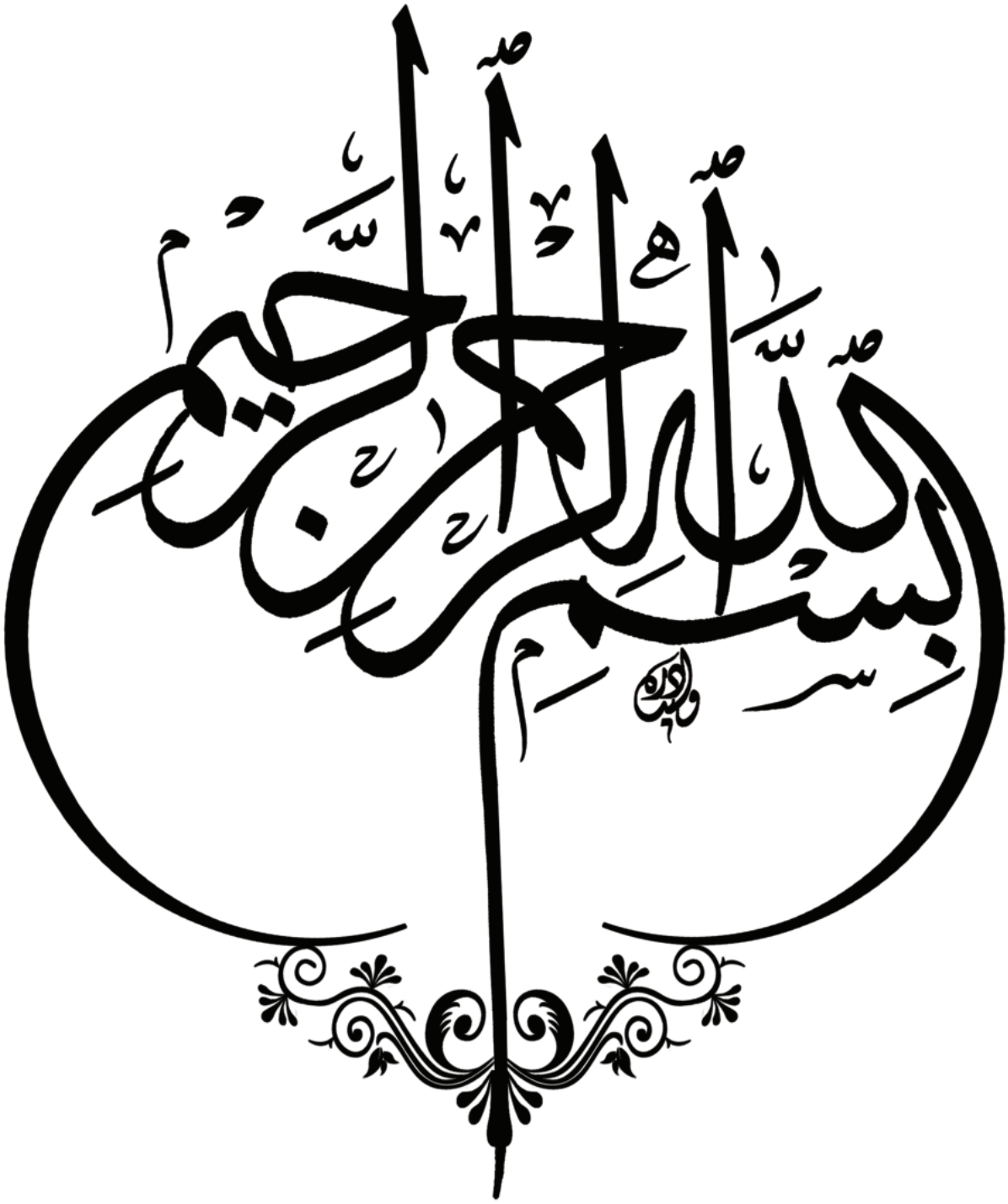
Présentée par

M. SLIMANI Yacine

Soutenu le : **08/12/2018**

Devant le jury composé de :

Dr SAIDI Mohamed	Université Farhat Abbas Sétif 1	Président
Pr MOUSSAOUI Abdelouahab	Université Farhat Abbas Sétif 1	Rapporteur
Dr LAMICHE Chaabane	Université M'SILA	Examinateur
Dr ZOUACHE Djaafar	Université BBA	Examinateur


بسم الله الرحمن الرحيم

Résumé La fouille du Web est l'application de fouille de données sur les données Web et la fouille d'usage du Web est une composante importante de la fouille du Web. Le but de la fouille de l'usage du Web est de comprendre le comportement des utilisateurs du site Web à travers le processus de fouille de données d'accès au Web. Les connaissances issues de la fouille de l'usage du Web peuvent être utilisées pour améliorer la conception du Web, introduire un service de personnalisation et offrir une navigation plus efficace. La dissertation propose trois contributions principales ; dans la première contribution, nous proposons un processus de fouille d'usage du Web. Le processus implique le prétraitement des données, l'intégration des données provenant de sources multiples et la transformation des données intégrées en une forme appropriée pour les opérations de fouille de données spécifiques. L'étape de prétraitement utilise un algorithme heuristique afin de réduire le temps de traitement et de regrouper les requêtes du Web en un certain nombre de sessions des utilisateurs, ce qui peut aider à déterminer le comportement de l'utilisateur d'une manière significative. Notre deuxième contribution modélise les communautés Web et analyse la corrélation entre les comportements des utilisateurs dans la phase de découverte de modèles. En fait, nous proposons une nouvelle approche probabiliste pour détecter les communautés Web. Notre méthode analyse chaque transaction d'utilisateur comme une probabilité de transitions et définit un coefficient de pertinence pour la quantification de comportements de l'utilisateur. Le modèle obtenu détecte les utilisateurs dans les communautés Web en utilisant les informations potentiellement utiles disponibles au cours des différentes sessions des utilisateurs. Dans notre troisième contribution et dans la phase de découverte, nous avons proposé une approche efficace dans laquelle nous utilisons une mesure de dissimilarité basée sur un processus stochastique qui converge vers une partition optimale du graphe du Web. Le modèle extrait montre une identification précise de la structure de la communauté et résulte d'une description significative des comportements des utilisateurs. Cette contribution apporte plus d'efficacité en terme de temps et de précision pour améliorer les tâches d'extraction du processus WUM.

Mots clés : Fouille des usagers du web, sessions de navigation, motifs fréquents, extraction de l'information, partitionnement du graphe, méthode de découverte de communautés, méthode de marche aléatoire, optimisation de la modularité, chaîne de Markov.

Abstract Web mining is the application of data mining on web data and web usage mining is an important component of web mining. The goal of web usage mining is to understand the behavior of web site users through the process of data mining of web access data. Knowledge obtained from web usage mining can be used to enhance web design, introduce personalization service and facilitate more effective browsing. The dissertation proposes three main contributions ; in the first contribution, we propose a Web usage mining process. The process involve pre-processing the original data, integrating data from multiple sources, and transforming the integrated data into a form suitable for input into specific data mining operations. The pre-processing step uses an heuristic algorithm in order to reduce time consumption and aggregates web requests into a number of user sessions, which can help to determine the user's behavior in a meaningful way. Our second contribution models the Web communities and analyzes the correlation between user behaviours in pattern discovery phase. In fact, we propose a novel probabilistic approach to detect Web communities. Our method analyzes each user transaction as a transition probability and defines a relevance coefficient for the quantification of user behaviors. The resulting pattern detects users within Web communities using the potentially useful information available during different user sessions. In our third contribution and in the discovery phase, we have proposed an efficient approach in which we use a dissimilarity measure based on a stochastic process that converges to optimal partition of the Web graph. The extracted model shows an accurate identification of the community structure and results from a meaningful description of users behaviors. This contribution provides more efficiency in terms of time and accuracy to improve the WUM process extraction tasks.

Keywords : Web Usage Mining (WUM), Web browsing sessions, frequent pattern mining, knowledge extraction, graph partition problem, community detection method, random walk method, modularity optimizatio, Markov chain.

Remerciements

Je remercie Dieu le tout Puissant qui m'a donné la force et la volonté pour réaliser ce modeste travail.

Je remercie mon directeur de thèse : le Professeur Abdelouahab Moussaoui, pour la confiance qu'il m'a accordée en acceptant d'encadrer ce travail doctoral, pour ses multiples conseils précieux et pour toutes ses orientations. Sa confiance et ses encouragements étaient un support infaillible pour arriver aux objectifs fixés au départ de ce travail de thèse.

J'adresse mes plus vifs remerciements à Monsieur Dr SAIDI Mohamed, Professeur à l'université de Sétif, d'avoir accepté de présider le jury de soutenance. J'exprime aussi tous mes remerciements à Dr C. LAMICHE et Dr D. ZOUACHE les membres de mon jury. Veuillez recevoir l'expression de ma profonde gratitude.

A la mémoire du Professeur Yves Lechevallier ; Ex-Directeur de Recherche à l'INRIA. Aucune reconnaissance ne saurait exprimer l'estime et le respect que j'ai toujours eu pour vous. Je n'oublie jamais les mois de stages durant lesquels Monsieur Lechevallier a consacré plusieurs journées pour me former en statistique et discuter avec moi certains points clés de mon analyse. De plus, les conseils qu'il m'a prodigué tout au long de mes recherches ont toujours été d'une grande aide pour mener à bien ma thèse. Pr Lechevallier a cru en mes capacités et m'a fourni d'excellentes conditions logistiques au laboratoire AxIS–I.N.R.I.A – Paris Rocquencourt, France.

Mes remerciements s'adressent également à ma collègue Dr Drif Ahlem, avec qui j'ai travaillé sur la thématique de détection de communautés, pour avoir crée un environnement d'études convivial et persévérant durant toute cette formation. Sa détermination, sa patience, son regard opérationnel critique m'ont été précieux. Je l'en remercie tout particulièrement.

Je remercie mes collègues de toute l'équipe de AxIS–I.N.R.I.A. Leur convivialité m'a fourni un cadre de travail particulièrement agréable et enrichissant.

Je tiens aussi à remercier Dr Rachid Hedjam et Pr Mohamed Cheriet, de l' ETS Montreal Canada, qui m'ont accueilli au sein de leur laboratoire Synchromedia. Je leur remercie aussi pour tous ces discussions fructueuses.

Je tiens à remercier mes amis et collègues du département de technologie pour leur collaboration durant tous ces années. Sans oublié les gérants et le personnel de MSI Yacer , SNC El-Rihane et ERUL Noor-Tech pour leur aides logistiques.

Enfin je remercie tous ceux qui ont contribué à l'aboutissement de ce travail.

Dédicaces

A mes parents,
A ma femme et mes enfants,
A la mémoire de mes grands parents,
A toute ma famille et mes amis

Table des matières

Table des figures

Liste des tableaux

Liste des Algorithmes

Chapitre 1

Introduction Générale

Le Webest devenu l'une des plates-formes les plus répandues pour la diffusion et la recherche d'information. Par conséquence, beaucoup d'opérateurs de sites Web sont incités à analyser l'usage de leurs sites afin d'améliorer leur réponse vis-à-vis des attentes des internautes. Or, la manière dont un site Web est visité peut changer en fonction de divers facteurs. Les modèles d'usage doivent ainsi être mis à jour continuellement afin de refléter fidèlement le comportement des visiteurs. De plus, la croissance exponentielle du domaine du Web tant dans le nombre de sites Web disponibles que dans le nombre d'utilisateurs de ces sites a généré de très grandes masses de données relatives aux traces d'usage du Web par les internautes, celles-ci enregistrées dans des fichiers logs Web.

Le Web Usage Mining consiste à analyser le comportement de l'utilisateur à travers l'analyse de son interaction avec le site Web. Les principales sources des données du Web Mining sont les pages web, les fichiers logs du serveur, les bases de données clients et les cookies qui permettent d'alimenter des data webhouses. Ces données sont classifiées en quatre types [SCDT00] :

- Données relatives au contenu : données contenues dans les pages Web (textes, graphes),
- Données relatives à la structure : données décrivant l'organisation du contenu (structure de la page, structure inter-page),
- Données relatives à l'usage : données sur l'usage telles que les adresses IP, la date et le temps des requêtes, fournies par les fichiers logs du serveur Web,
- Données relatives au profil de l'utilisateur : données fournissant des informations démographiques sur les utilisateurs du site Web.

En se basant sur ces types de données, les axes de développement actuels du Web sont :

- Le Web Content Mining (WCM) : consiste en une analyse textuelle avancée (traitement linguistiques, classification des pages, segmentation thématique...) intégrant les particularités du Web telles que la structure sémantique des pages.

- Le Web Usage Mining (WUM) : s'intéresse à l'analyse des comportements de navigation sur les sites Web notamment l'analyse du clickstream (l'ensemble des clics exécutés sur le site) afin de mesurer l'audience et la performance des sites Web (temps passé par page, nombre de visites, profil de l'utilisateur, horaires et fréquences des consultations,...) et d'enrichir les sources de données de l'entreprise et de l'organisation (bases de données clients, bases marketing,...).
- Le Web Structure Mining (WSM) : consiste à analyser l'architecture des sites Web et des liens entre les différents sites afin d'améliorer leur ergonomie par la suppression ou l'ajout de nouveaux liens entre les pages.

En fait, le Web Usage Mining (WUM)correspond justement au processus d'extraction des connaissances à partir des données (ECD), ou Knowledge Discovery in Databases (KDD), en anglais, appliqué aux données d'usage sur le Web. Ce processus, décrit dans la figure 1, englobe trois étapes principales [CS00] : le prétraitement des données, la découverte des schémas et l'analyse (ou l'interprétation) des résultats. Un processus WUM extrait des patrons de comportement à partir des données d'usage et, éventuellement, à partir d'informations sur le site (structure et contenu) et sur les utilisateurs du site (profils). La quantité des données d'usage à analyser ainsi que leur faible qualité (en particulier l'absence de structuration) sont les principaux problèmes en WUM. Les algorithmes classiques de fouille de données appliqués sur ces données donnent généralement des résultats décevants en termes de pratiques des internautes (par exemple des patrons séquentiels évidents, dénués d'intérêt). Dans cette thèse, nous apportons des contributions importantes pour le processus WUM. Tout d'abord,nous proposons une méthodologie générale de prétraitement des logs Web et les résultats de cette analyse sont exploités par la suite pour comprendre les comportements de navigation sur un site Web. Ensuite, nous proposons de nouvelles approches intégrant la notion de communauté Web pour la phase de découverte des patterns d'usage du Web. Les patterns obtenus identifient et caractérisent minutieusement le comportement des usagers du Web.

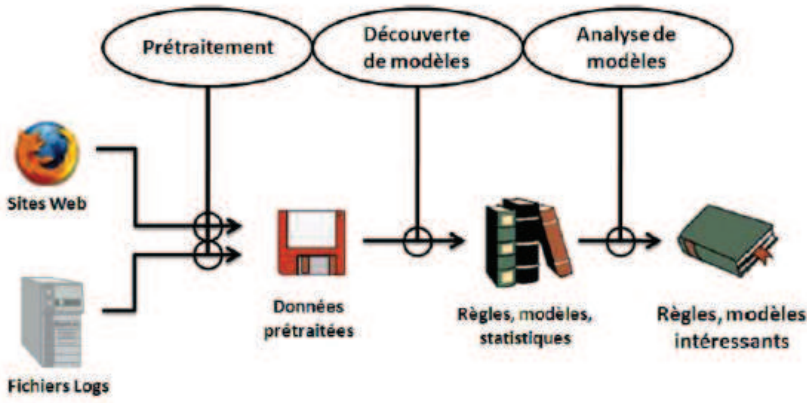


FIGURE 1.1 – Processus du Web Mining

Objectifs et motivations

Le Web permet de mieux répondre au besoin toujours plus grandissant d'information et de connaissances. Le WUM peut encore apporter des avantages à d'autres domaines, comme par exemple, l'ajout dynamique de liens dans des pages Web [YJGMD96], la recommandation de produits [PPPS03], [Mob04], la caractérisation de groupes d'utilisateurs, l'amélioration de politiques comme le caching et le prefetching anticipés [SOBB03], etc. Mais en apportant une solution, il apporte par le fait même de nouveaux problèmes aussi bien pour les internautes que pour les concepteurs des sites Web. Parmi ces problèmes :

- Trouver l'information recherchée : les internautes, à la recherche d'une information spécifique, utilisent des moteurs de recherche qui retournent un ensemble de pages assez similaires à la requête de l'utilisateur. Cependant, la performance des algorithmes utilisés par ces moteurs n'est pas assez satisfaisante à cause de la faiblesse de leur précision et leur rappel. La faiblesse de la précision est due au nombre élevé de documents non pertinents retrouvés pour une requête donnée alors que la faiblesse du rappel est due à la difficulté d'indexer toute l'information disponible sur le Web [Cha00].
- Créer des nouvelles connaissances à partir de l'information présente sur le Web : alors que le problème précédent est orienté recherche d'information, le second est orienté data mining. Il suppose la présence d'une collection des données Web à partir desquelles de la connaissance devrait être extraite. Plusieurs travaux ont été menés dans ce sens [MBNL99] , [CMP+98] afin d'utiliser le Web comme une base des connaissances pour l'aide à la décision.
- Comprendre le comportement des consommateurs ou visiteurs des sites Web et personnaliser l'information : ce problème concerne les concepteurs et les gestionnaires des sites Web. Il consiste à identifier les visiteurs des sites, leurs préférences, leurs motifs de visite et apporter des modifications aux sites pour répondre à leurs attentes.

Ce dernier problème a fait l'objet de plusieurs travaux de recherches. Dans le travail [CCB02], les auteurs ont proposé l'approche "SurfMiner" reposant sur l'analyse des fichiers Logs afin de découvrir les usages d'un site associés à des descriptions d'utilisateurs. Cette approche repose sur l'hypothèse qu'il existe une certaine corrélation entre les pratiques différentes des utilisateurs et leurs caractéristiques personnelles. Elle consiste à extraire des motifs fréquents de navigation des utilisateurs de référence et découvrir des relations entre les motifs découverts et des traits d'utilisateurs. Srivastava et al [SCDT00] cherchent dans les données extraites des fichiers logs deux types de classes : classes d'usagers et classes de pages. Par contre, la classification des utilisateurs a pour objectif d'établir des groupes d'internautes ayant des comportements de navigation similaires. L'examen de ces groupes permet d'associer un profil à chaque classe d'utilisateurs. La classification des pages Web consultées par les internautes permet de découvrir des groupes de pages ce qui facilite la tâche des navigateurs et des robots. Dans ce cas, la classification des pages est basée sur les résultats de la classification des utilisateurs. En conséquence, la majorité des travaux

associent la classification des pages à la classification des utilisateurs construisent les classes des pages à partir des classes des utilisateurs ou à partir des visites des internautes (pages visitées ensemble par plusieurs visiteurs). En d'autres termes, la classification des utilisateurs guide la classification des pages Web [KV09].

Dans le travail [LHY05], un algorithme de classification simple, DBSCAN est appliqué aux données issues des fichiers logs pour découvrir des classes de pages puis les classes des utilisateurs sont construits à partir des classes des pages. Ainsi, la liaison entre l'usage et le contenu est effectuée à travers les données sur l'usage. Le contenu textuel ou multimédia des pages n'est pas utilisé dans la classification des pages. D'autres efforts ont été concentrés sur l'analyse du contenu des pages Web. Charrad et al [CLSA08] ont intégré le contenu textuel des pages dans l'analyse de l'usage étant donné que le comportement des internautes sur un site dépend fortement du contenu proposé dans les pages. Ce qui permet d'évaluer la conception du site par la comparaison entre la structure sémantique et la structure logique et confronter cette structure logique à la perception des utilisateurs représentée par les traces de navigation enregistrées dans les fichiers logs. Dans le travail [YLK+16], Yu et al ont proposé une approche pour déterminer les pages Web préférées en générant des pages Web volumineuses et des modèles émergents de simple graphe liés à des pages. Cette approche identifie les pages Web préférées de chaque utilisateur en éliminant le bruit dû aux pages populaires globales et en regroupant les utilisateurs Web selon les motifs émergents générés.

Dans les dernières années, la recherche sur l'analyse de réseaux sociaux dans le Web est devenue un axe nouvellement actif en raison de l'avènement des nouvelles technologies [XZL10]. De plus, pour faire face à la complexité des caractéristique des données Web, les méthodes de détection de communautés se sont imposée comme un nouveau moyen efficace de gestion des données Web pour modéliser les objets Web et les communautés Web . Contrairement à la gestion de base de données conventionnelle, dans laquelle les modèles de données et le schéma sont bien définis, la communauté Web qui est un ensemble d'objets Web a sa propre structure logique. Ces considérations nous ont motivé à analyser le comportement des utilisateurs dans le processus de fouille de données d'usage du Web (Web Usage Mining) en capturant les caractéristiques sociales et sociétales de la structure du réseau à travers le Web afin d'améliorer l'efficacité des tâches d'extraction du processus WUM. Il s'agit d'apporter des contributions au niveau de l'extraction et l'analyse des nouveaux patterns en proposant des méthodes efficace d'identification des communautés Web. De ce fait, les approches que nous avons proposées découvre des patterns servant à mieux comprendre le comportement général des visiteurs du site Web, et ainsi offrir de nouvelles connaissances utiles. Ce qui mène à restructurer et personnaliser le design des sites Web. Les apports du présent travail sont illustrés dans la Fig. 1.2

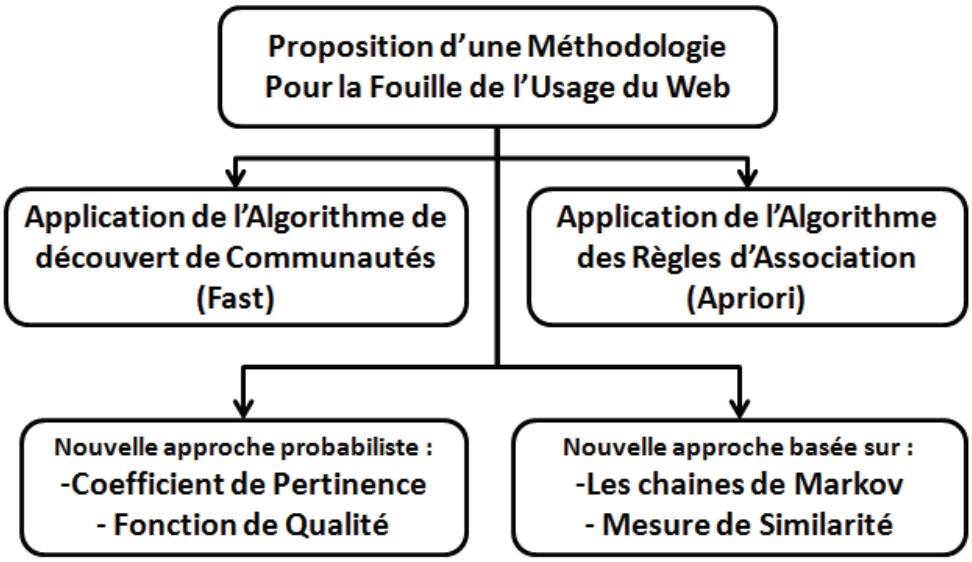


FIGURE 1.2 – Nos travaux de recherche.

Problématique et contributions

La caractérisation des internautes fréquentant un site Web est un problème incontournable pour assister l'internaute et prédire son comportement. Afin de contribuer à l'avance de la recherche scientifique dans la communauté d'analyse des données, cette thèse est axée sur l'investigation de l'efficacité des stratégies de classification non supervisée les plus adaptées à la découverte des communautés Web et présente quartes contributions majeures :

- Premièrement, nous proposons une méthodologie de fouille d'usage du Web partant des comportements de navigation durant les sessions des utilisateurs pour faciliter davantage la manière dont cette information est présentée au sein du site. Dans la phase du prétraitement de données, les données , stockées dans des fichiers logs des serveurs Web, sont nettoyées en enlevant l'information et le bruit non pertinents. Ensuite, les données restantes sont arrangées d'une manière cohérente afin d'identifier d'une façon précise les sessions des utilisateurs. En effet, une fois les sessions des utilisateurs ont été identifiées, on les utilise pour extraire le degré d'intérêt des utilisateurs pour chaque ressource Web. Après l'identification des sessions des utilisateurs, on applique l'algorithme Apriori affin d'extraire des règles d'associations qui définissent le comportement des usagers du site Web étudié ; et nous permettent de personnaliser ce dernier pour s'adapter aux besoins de ses utilisateurs. Notre méthodologie du processu WUM a permet non seulement de réduire considérablement la taille des fichiers Logs, mais également à regrouper les requêtes Web dans un certain nombre de sessions des utilisateur ce qui peut aider à déterminer le comportement de l'utilisateur d'une manière significative. Cette partie de la thèse a fait l'objet des communications et publications suivantes :

1. **Journal : Yacine Slimani**, Abdelouahab Moussaoui. La fouille des usagers du Web par application de l'algorithme Apriori sur les fichiers logs. Revue d'information Scientifique et Technique. Vol 18(2010)N 1, Editeur : CERIST, ISSN : 1111-0015 [SM10].
2. **Conference : Yacine Slimani**, Abdelouahab Moussaoui, Abdelmalek Guessoum. Extraction des règles d'associations depuis les fichiers logs dans un processus de fouille des usagers du Web. Conférence Internationale des technologies de l'information et de la Communication, CITIC'2009 – Sétif (Algérie) : 4–5 Mai 2009 [SMG09a].
3. **Conference : Yacine Slimani**, Abdelouahab Moussaoui, Abdelmalek Guessoum. Prétraitement des fichiers journaux d'accès dans le processus de la fouille des usagers du Web. Les 1ères Journées Scientifiques sur l'Informatique et ses Applications, JSIA09– Guelma (Algérie) : 3 – 4 Mars 2009 [SMG09b].
4. **Conference :** Nourddine Mekroud, Abdelouahab Moussaoui,**Yacine Slimani**. Intégration des techniques du Datamining et des bases de données avancées dans le processus de gestion des connaissances : proposition d'un processus hybride basé sur le raisonnement à partir de cas, CITIC'2009 – Sétif (Algérie) :4–5 Mai 2009. [MMS09]

- En second lieu, nous avons abordé le domaine des réseaux complexes qui a eu un essor important dans les dernières années, depuis des aspects très fondamentaux jusqu'à des applications interdisciplinaires (sciences sociales, informatique, épidémiologie, ...ect). Les réseaux complexes : en font partie les réseaux sociaux, le graphe du Web, les réseaux neuronaux, ainsi que de nombreuses classes de graphes du monde réel. En plus de leur taille considérable, les réseaux complexes partagent des propriétés structurelles assez surprenantes tel que celle de la structure communautaires. Tout d'abord, la notion générale de structure communautaire dans les réseaux complexes a été introduite pour la première fois dans la littérature de physique par Girvan et Newman [GN02], et se réfère au fait que les nœuds dans de nombreux réseaux réels semblent se regrouper en sous-graphes dont les sommets sont plus liés entre eux qu'avec le reste du réseau. Ensuite, la structure de communautés a été identifiée dans de nombreux réseaux technologiques, biologiques et sociaux réels [GA05], [HHJ03] et son émergence semble être au coeur du processus de formation des réseaux du monde réel [GDDG$^+$03]. Dans notre travail, nous visons à comprendre une telle propriété et son émergence dans les graphes Web. A cet effet, nous avons étudié les méthodes de détection communautaires existantes ce qui nous a permis de proposer de nouvelles approches de découverte de communautés et les appliquer aux divers contexte à savoir dans le domaine des réseaux mobiles technologiques. Nos publications académiques liées aux approches de la découverte des communautés dans les réseaux complexes sont les suivantes :

1. **Rapport technique** : Drif Ahlem, Boukerram Abdellah, **Slimani Yacine**, Moussaoui Abdelwaheb. (2016). Découverte de communautés dans les réseaux complexes. Hal.archives-01389844 [DBSM16].
2. **Journal :**Ahlem Drif, Abdellah Boukerram, **Yacine Slimani**, Silvia Giordano. Discovering Interest Based Mobile Communities, Mobile Networks and Applications, Print ISSN : 1383-469X, Online ISSN : 1572-8153, Vol. : 21 Issues : 110, P1-12 DOI : 10.1007/s11036-017-0811-3 (Springer Feb. 2017) [DBSG17].
3. **Journal :** Ahlem Drif, Abdallah Boukerram, **Yacine Slimani**. Community Discovery Topology Construction for Ad Hoc Networks, Wireless Internet : 8th International Conference, WICON 2014, Lisbon (Portugal), Nov. 13-14, 2014, ISSN 1867-8211 Volume :146, P197-208, DOI 10.1007/978-3-319-18802-7-28 [DBS14].

- Après avoir mis en lumière le concept de structure de communautés et souligné certaines limitations des approches de découverte de communautés existantes, nous avons adapté l'algorithme de Newman et al dans le contexte de l'extraction d'un patterns d'accès des utilisateurs du Web. Nous avons commencé par définir la notion de "communauté Web" et par modéliser son schéma d'extraction afin de l' adapter à nos besoins d'analyse. De ce fait, nous présentons deux approches pour l'extraction des données basées sur la fonction de modularité des graphes, et cela après la modélisation des sessions de navigations des internautes en graphes. La première approche permet la découverte des communautés existantes dans un graphe non pondéré quand à la deuxième approche, qui utilise un graphe pondéré, permet de mieux distinguer ces communautés. L'application de ces techniques sur le site Web de l'université de Ferhat Abbes de Sétif, a permis de mieux découvrir le centre d'intérêt des internautes et ainsi pouvoir restructurer le site selon leurs besoins. Nos travaux là-dessus ont fait l'objet de plusieurs publications et conférences internationales :

  1. **Conférence : Yacine Slimani**, Abdelouahab Moussaoui. Analyse des sessions de navigations du site Web de l'U.F.A.S par les algorithmes de réseaux sociaux. Colloque sur l'Optimisation et les Systèmes d'Information, COSI'2012 –Tlemcen (Algérie), 12-15 Mai 2012 [SM12b].
  2. **Journal : Yacine Slimani**, Abdelouahab Moussaoui, Yves Lechevalier, Ahlem Drif. Identification de communautés d'usage du Web depuis un graphe issus des fichiers d'accès, 12ème Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances, EGC2012 – Bordeaux (France) : 31 janvier–3 février 2012. Revue des Nouvelles Technologies et d'Information (RNTI). Vol E.23 (2012) Editeur : Hermann, isbn 978 2 7056 8310 8 [SMLD12].
  3. **Conférence :** Ahlem Drif, Abdallah Boukerram, **Yacine Slimani**. Découverte d'une structure de communauté des usagers du Web, 4ème Conférence sur les Modèles et l'Analyse des Réseaux : Approches Mathématiques et Informatiques, marami2013, Saint-Etienne (France), 16 au 18 octobre 2013 [DBS13].

4. **Conférence : Yacine Slimani**, Abdelouahab Moussaoui, Yves Lechevalier, Ahlem Drif. A community detection algorithm for Web Usage Mining Systems, The 4th IEEE International Symposium on Innovation in Information and Communication Technology, ISIICT2011–Philadelphia University Amman (Jordan), Nov. 29–Dec. 1 2011. (IEEE 2011) isbn 978 1 61284 675 0 [SMLD11].
5. **Conférence : Yacine Slimani**, Abdelouahab MOUSSAOUI . A social network algorithm for detecting communities from weighted graph in Web Usage Mining system, The International Conference on Information and Communication Technology (ICICT'2012), Ramallah, Palestine 26th, June 2012 [SM12a].

- Ensuite, nous avons proposé une nouvelle approche probabiliste pour la découverte des communautés et l'adapter au contexte de Web mining afin de profiter du pouvoir classificatoire des méthodes de détection de communautés. Notre idée est basé sur le fait qu'un noeud a une forte probabilité d'être membre à une communauté s'il présente un nombre d'occurrence significative au sein d'un même groupe. L'analyse des traces des internautes ainsi que l'extraction des comportements minoritaires passibles d'avoir lieu pendant de courtes périodes de temps restent ainsi inaperçus par les méthodes classiques. La prise en compte de la pertinence des intérêts des visiteurs s'avère donc nécessaire pour l'analyse de ce type de données. En conséquence, nous avons définit un coefficient de pertinence qui quantifie l' information utile véhiculé par les séquences d'accès de l'utilisateur et reflète la probabilité d'appartenance d'un utilisateur à une communauté. Ainsi,nous avons réécrit la modularité et définit un algorithme agglomératif pour le partitionnement du graphe. L'apport de ce travail a fait l'objet des publications suivantes :

  1. **Conférence : Yacine Slimani**, Abdelouahab Moussaoui, Ahlem Drif. Using weighted graph for discovery of usage patterns from Web data. In The First International Symposium on Informatics and its Applications, ISIA 2014, M'sila (Algeria), February 25-26, 2014 [SMD14].
  2. **Journal : Yacine Slimani**, Abdelouahab Moussaoui, Ahlem Drif. Discovery and Analysis of Usage Patterns for Web Personalization, International Journal on Recent and Innovation Trends in Computing and Communication ISSN : 2321-8169 Volume :3 Issue :2 P578-582, DOI : 10.17762/ijritcc2321-8169.150232 [SMD15].

- Notre dernière contribution présente une nouvelle méthode pour détecter les communautés Web en utilisant les chaînes de Markov. Lorsqu'un système est modélisé par une équation différentielle son avenir est uniquement déterminé par sa situation présente. Les chaînes de Markov , au contraire, font l'hypothèse qu'il y a plusieurs évolutions possibles à partir de la situation présente, chacune d'elles ayant une certaine probabilité de se réaliser. Pour un système possédant plusieurs avenirs possibles à partir

de son état présent, il se pourrait que la probabilité que l'un ou l'autre de ces avenirs se réalise dépende non seulement de son état présent mais aussi de son histoire récente. La propriété de Markov $P(X_{t+1} = x_j/X_t = x_i, X_{t-1} = x_k, X_{t-2} = x_l, ...) = P(X_{t+1} = x_j/X_t = x_i)$ prend en compte ce type de dynamique aléatoire. A cet effet, nous avons modéliser le comportement des utilisateurs durant leurs sessions de navigation par les chaînes de Markov. Notre idée consiste à analyser la probabilité d'occurrence de toutes les séquences fréquentes au cours de différentes sessions d'utilisateur afin d'extraire les communautés Web décrivant le comportement des utilisateurs. Nous envisageons aussi à réécrire la modularité pour avoir des partitions optimales. En fait, la modularité est une mesure de qualité qui permet de comparer deux partitions d'un même graphe [NG04]. Il a été spéculé que la maximisation de la modularité [New06] est un problème NP-difficile en raison de la similarité avec le problème MAX-CUT [FDPP06, FB07, MRC05]. Divers approches d'optimisation ont été proposées ; tel que les algorithmes gloutons [CNM04, New04a], les algorithmes de partition spectrale [New06, WS05], les algorithmes basé sur le recuit simulé [GSPA04b, RB06] et l' optimisation extrémale [DA05], tous ces algorithmes d'optimisation résultent d'un ensemble de partitions sous-optimales sur de nombreuses instances. En conséquence, il est justifié d'utiliser des algorithmes d'approximation et des heuristiques pour faire face au problème d'optimisation de la modularité. Ainsi, notre approche d'identification des communautés Web basée sur les chaînes de Markov permet d'extraire des communautés plus pertinentes et d'obtenir des partition sous optimales. Ce travail a été publié dans le journal IJAIP :

1. **Journal : Yacine Slimani**, Abdelouaheb Moussaoui, Yves Lechevallier, Ahlem Drif. Discovering Communities for Web Usage Mining Systems. International Journal of Advanced Intelligence Paradigms, ISSN online : 1755-0394, ISSN print : 1755-0386, DOI : 10.1504/IJAIP.2018.10017353 [SMDL18]

Organisation du document

Ce mémoire est composé de deux grandes parties. La première partie est composée de trois chapitres présentant un état de l'art sur le Data mining, le Web Mining,et les algorithmes de découvertes de communautés dans les réseaux complexes. La deuxième partie, composée également de quatre chapitres, présente les contributions de cette thèse. Les différents chapitres de la thèse sont organisés comme suit :

Le chapitre 2 : présente le cadre général de nos travaux de recherche. Ils se placent plus globalement dans le domaine de la fouille des données. Ce chapitre définit la data mining, et en décrit les principales applications et les phases d'un projets de data mining.

Le chapitre 3 : présente une brève revue de littérature sur la détection de communautés. Comme il existe de nombreuses approches proposées, nous allons retenir celles ayant le plus d'intérêts de la part de la communauté scientifique. Ces approches illustrent aussi la diversité

de méthodologies et donnent une vue d'ensemble des techniques proposées en décrivant leurs critères et leurs aspects liés à la structure communautaire.

Le chapitre 4 : est consacré à la présentation détaillée de notre méthodologie du processus Web usage mining. Celle-ci se compose des phases suivantes : le pré-traitement des données, la modélisation des données d'usage, des analyses exploratoires de classification guidée par les algorithmes de fouille de données, et finalement l'interprétation des résultats obtenus. Au début on a appliqué les règles d'associations comme outils de fouilles de données. Dans une seconde partie ; on a adapté les méthodes de découvertes de communautés aux besoins d'extraction de connaissance et d'analyse du processus WUM. La découverte des communautés Web dans le graphe issu de la structuration des navigations des usagers en sessions a permet de les identifier par leur sujet d'intérêt et donc d'extraire des connaissances pertinentes. En considérant cette approche, les expérimentations obtenues ont alors illustré qu'on peut améliorer le design du site Web en fonction du comportement des usagers du Web et les informations le concernant.

Le chapitre 5 : décrit notre méthode agglomérative d'identification de communautés adapté au contexte de WUM. En vue de définir une mesure plus performante pour la quantification de comportements de l'utilisateur, nous proposons un coefficient de pertinence calculé en fonction de l'occurrence des intérêts des utilisateurs. Notre approche est testé sur des données réelle afin d'extraire un patterns des usagers d'un site Web. Les expérimentations ont ainsi montré qu'une amélioration considérable de performances peut être réalisée.

Le chapitre 6 : est consacré à la présentation détaillée de notre approche CDMMC (Community Detection Method based on Markov Chain). La modélisation de comportement de l'utilisateur avec la chaîne de Markov en définissant une nouvelle mesure de modularité dans un contexte de WUM est l'une des particularités de notre travail. Ainsi, nous montrons comment obtenir une classification des comportement des usages du Web partant des propriétés des chaînes de Markov. Ce chapitre illustre nos expérimentations suivi par l'analyse des résultats obtenus à partir de l'application de notre approche sur différents jeux de données d'usage réelles.

Enfin, en guise de conclusions, nous indiquerons quelques remarques sur ce travail et nous exposerons les perspectives pour de futurs travaux.

Première partie

Etat de l'art

Chapitre 2

La Fouille de Données

Sommaire

2.1 Introduction

Durant ces dernières années, on a assisté à une forte augmentation dans le volume et dans les types des informations mémorisées par des bases de données scientifiques, médicales, économiques, financières, administratives, etc. Les techniques usuelles analysant ces données sont insuffisantes d'où le besoin d'une nouvelle génération d'outils et de théories pour aider à extraire les informations utiles (les connaissances) à partir de ces volumes de données numériques qui croissent rapidement.

L'extraction des connaissances à partir de ces bases de données est un domaine de l'informatique qui regroupe toutes les technologies et les outils qui réalise cette tache et dont le coeur est la fouille de données (datamining). Le data mining, dans sa forme et compréhension actuelle, à la fois comme champ scientifique et industriel, est apparu au début des années 90. On peut voir le data mining comme une nécessité imposée par le besoin des entreprises de valoriser les données qu'elles accumulent dans leurs bases.

Une confusion subsiste encore entre data mining, que nous appelons en français " fouille de données ", et " extraction des connaissances à partir des données" (ECD). Le data mining est l'un des maillons de la chaîne de traitement pour la découverte des connaissances à partir des données. Sous forme imagée, nous pourrions dire que l'ECD est un véhicule dont le data mining est le moteur.

La classification et la segmentation sont parmi des taches importantes du datamining, elles sont largement étudiée avec ses nombreuses extensions et sont probablement les techniques les plus répandues. Dans ce chapitre, nous allons réaliser une synthèse de la documentation étudiées dans laquelle nous allons introduire les notions de bases essentielles dans le domaine de la fouille de données.

2.2 L'Extraction des Connaissances à partir des Données [ECD]

2.2.1 Définition

L'Extraction de Connaissances à partir de Données (E.C.D en français) ou Knowledge Discovery in Databases (K.D.D en Anglais), désigne l'ensemble des étapes du processus interactif et itératif d'extraction de connaissances utiles à partir des données [FPSS96], c'est donc un processus d'identification de propriétés valides, nouvelles et potentiellement utiles dans les données larges et complexes , Définit autrement le processus de l'ECD est l'analyse de bases de données (souvent très grandes) afin de découvrir des relations souvent insoupçonnées et de résumer les données d'une manière à la fois compréhensible et utile. Ce processus vise à transformer ces données (volumineuses, multiformes) en informations dont on va extraire des connaissances intelligibles à l'usager et qui doivent être utiles, intéressantes dans le contexte, et favoriseront une meilleure décision [HKP00].

2.2.2 Problème de nomenclature

Dans la littérature, et en particulier dans la documentation française. En trouve beaucoup d'ambiguïté lors de la traduction de quelques termes de l'anglais au français. Le tableau 3.1 liste les erreurs les plus fréquentes :

TABLE 2.1 – Terminologie.

Anglo-saxons	**Francophones**	**Certains auteurs**
K.D.D	E.C.D	Fouille de données
Clustering	Classification	Segmentation
Classification	Classement	Classification
Decision trees	Arbres de décision	Segmentation

2.2.3 Notions de base

Le processus d'E.C.D réside dans l'extraction des connaissances à partir des informations fournies par les données, ces trois notions sont définies comme suit :

Donnée

Une donnée décrit des exemples ou des événements précis. Elle peut être recueillie de manière automatique ou par écrit. La notion de donnée est donc perçue comme la couche de plus bas niveau dans la hiérarchie conceptuelle du savoir. C'est sur elle que s'élaborent les notions d'information et de connaissance, Son exactitude peut être vérifiée par référence au monde réel. Dans le concept de donnée, il n'y a aucune notion d'interprétation ni de contexte, elle a une valeur quantitative ou qualitative, se présente sous la forme d'un nombre, d'un extrait électronique de texte ou d'une image, etc.

L'information

Un élément de connaissance susceptible d'être codé ou représenté à l'aide de conventions pour être conservé, traité ou communiqué , quand on donne un sens à une donnée à travers un cadre interprétatif, elle devient information [Tsu93], les informations sont des données organisées présentées en contexte La notion d'information présente donc un degré conceptuel plus élevé que la notion de donnée dans la valeur et la signification qu'elle occupe dans l'application où elle est utilisée, elle est toujours associée à la possibilité de traitement informatique c'est-à-dire son stockage sous forme exploitable pour les applications. Pour ces raisons, une information doit donc être associée à une représentation formelle permettant sa traduction informatique et conceptuelle.

La connaissance

Les connaissances sont définies soit comme des informations affinées, synthétisées, systématisées, soit comme des informations associées à un contexte d'utilisation [ECB+96], la connaissance est l'information en contexte, associée à une compréhension de son mode d'utilisation. [Bro99] Ainsi, le principe est, idéalement, à partir de données dont on ne sait rien et sur lesquelles on ne fait aucune hypothèse, d'obtenir des informations pertinentes, et à partir de celle ci de découvrir de la connaissance qui est donc le résultat d'une information traitée, compréhensible et assimilable par un être humain. Ce qui est typique de la connaissance, c'est qu'elle résulte d'un processus complexe.

2.3 Les phases du processus de E.C.D

Le processus de l'ECD consiste en une séquence itérative des étapes suivantes :

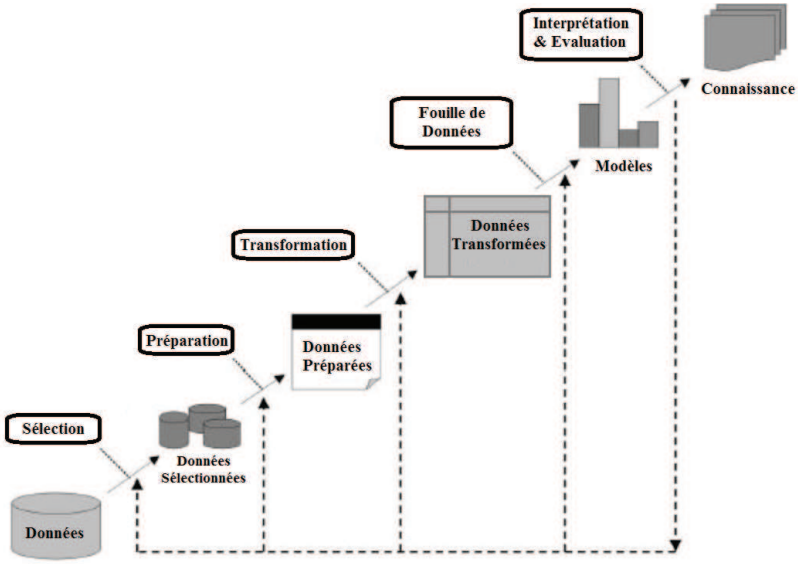


FIGURE 2.1 – Le processus de l'E.C.D.[Bro99].

2.3.1 Phase d'acquisition des données

Les données peuvent être localisées sur des sites différents de celui où s'effectue l'ECD, elles peuvent être stockées selon des architectures variées (des bases de données relationnelles, des entrepôts de données, le web ou des banques de données spécialisées) , comme elles peuvent être structurées ou non selon différents types : données tabulaires ou textuelles, images, sons ou séquences vidéo La phase d'acquisition nécessite le recours à des moteurs de recherche de données, elle peut passer par les moteurs de requêtes des bases de données comme le langage SQL ou à travers des outils de requêtes plus spécifiques aux données non structurées comme les données textuelles, les images ou le web (des moteurs de recherche

par le contenu) , et sert généralement à nettoyer les données qui sont rapatriées. On peut également e limiter le nombre d'enregistrements que l'on souhaite traiter. A l'issue de la phase d'acquisition, un stock de données contenant potentiellement l'information ou la connaissance recherchée est obtenu.

2.3.2 Phase du prétraitement des données

Les données issues des entrepôts ne sont pas nécessairement toutes exploitables par des techniques de fouille de donnée, ses données acquises peuvent être de types différents (trouver des textes de longueur variables, des images, des enregistrements quantitatifs ou des séquences vidéo).

La préparation consiste à homogénéiser les données et à les disposer en tableau lignes/colonne, chaque ligne/colonne peut être considérée comme un objet vecteur ayant un nombre fixe de composantes. Ce vecteur ligne/colonne sera vu comme un objet mathématique que l'on pourra manipuler selon qu'il possède ou non certaines propriétés. Les principales opérations de préparation peuvent être listées comme suit :

- **Sélection de ligne/colonne :** L'objectif est soit de réduire le nombre de données soit de sélectionner les lignes ou colonnes les plus pertinentes par rapport aux préoccupations de l'utilisateur.
- **Le traitement des données manquantes ou aberrantes :** Il faut définir des règles pour gérer ou pour remplacer ces données manquantes.
- **Les transformations d'attributs :** Il s'agit de transformer un attribut A en une autre variable A' qui serait, selon les objectifs de l'étude, plus appropriée.
- **La construction d'agrégats :** Un agrégat d'attribut est un nouvel attribut obtenu selon une transformation précise. Par exemple, le prix au mètre-carré d'un appartement, défini par le rapport entre le prix de l'appartement et la surface totale de l'appartement, fournit une indication assez pertinente pour comparer les appartements ou les quartiers dans les bases de données spatiales.
- **Le traitement des données complexes :** Toutes les méthodes de prétraitement citées précédemment opèrent sur des tableaux de données lignes/colonnes. Or il arrive que nous travaillions sur des données non structurées sous forme de tableaux. Par exemple, en fouille de texte, nous disposons d'un ensemble de textes de longueurs variées qu'il convient de ramener à une forme tabulaire

2.3.3 Phase de fouille de données

La fouille de données (data mining) est au coeur du processus d'ECD. Cette phase fait appel à de multiples méthodes issues de la statistique, de l'apprentissage automatique, de la reconnaissance de formes ou de la visualisation. Les méthodes du Data Mining permettent de découvrir ce que contiennent les données comme informations ou modèles utiles. Si nous

essayons de classifier les méthodes de fouille de données utilisées, trois catégories se distinguent :

- Les méthodes de visualisation et de description.
- Les méthodes de classification et de structuration.
- Les méthodes d'explication et de prédiction.

Chacune de ces familles de méthodes comporte plusieurs techniques appropriées aux différents types de tableaux de données. Certaines sont mieux adaptées à des données numériques continues alors que d'autres sont plus généralement dédiées aux traitements de tableaux de données qualitatives. Nous allons donner un aperçu général sur les principales méthodes dans la suite du travail.

2.3.4 Phase de validation et de mise en forme

Les modèles extraits, notamment par les méthodes d'apprentissage supervisé, ne peuvent être utilisés directement en toute fiabilité. Nous devons les évaluer, c'est-à-dire les soumettre à l'épreuve de la réalité et apprécier leur justesse. Le procédé habituel consiste à estimer au mieux le taux d'erreur du modèle. Ainsi, l'utilisateur décidera d'appliquer ou non le modèle de prédiction en connaissance des risques qu'il prend. Le taux d'erreur est généralement calculé à partir de la

matrice de confusion. Celle-ci donne le pourcentage d'affectation dans les différentes classes en fonction des classes d'origine. [ZR02]

2.4 Principales taches de fouille de données

On dispose de données structurées. Les objets sont représentes par des enregistrements (ou descriptions) qui sont constitues d'un ensemble de champs (ou attributs) prenant leurs valeurs dans un domaine. De nombreuses taches peuvent être associées au Data Mining, parmi elles nous pouvons citer :

2.4.1 La classification

Elle consiste a examiner les caractéristiques d'un objet et lui attribuer une classe, la classe est un champ particulier a valeurs discrètes. Des exemples de tache de classification sont :

- attribuer ou non un prêt a un client,
- etablir un diagnostic,
- accepter ou refuser un retrait dans un distributeur,
- attribuer un sujet principal a un article de presse,
- etc.

2.4.2 L'estimation

Elle consiste a estimer la valeur d'un champ a partir des caractéristiques d'un objet. Le champ a estimer est un champ a valeurs continues. L'estimation peut être utilisée dans un but de classification. Il suffit d'attribuer une classe particulière pour un intervalle de valeurs du champ estime. Comme par exemple , estimer les revenus d'un client.

2.4.3 La prédiction

Cela consiste a estimer une valeur future. En général, les valeurs connues sont historiés. On cherche a prédire la valeur future d'un champ. Cette tache est proche des précédentes. Les méthodes de classification et d'estimation peuvent être utilisées en prédiction. Des exemples de taches de prédiction sont :

- prédire les valeurs futures d'actions,
- prédire, au vu de leurs actions passées, les départs de clients.

2.4.4 L'analyse des clusters

Le clustering [Lar14] (ou la segmentation) est le regroupement d'enregistrements ou des observations en classes d'objets similaires, des groupes (ou clusters). Un cluster est une collection d'enregistrements similaires l'un à l'autre, et différents à ceux existants sur les autres clusters, et que deux groupes différents contiennent certainement des objets suffisamment différents. Etant donné que chaque instance doit être semblable aux autres instances du même groupe, et dissemblable avec les instances des autres groupes, il est habituel que les méthodes de clustering fassent usage d'une mesure de similarité, afin d'identifier les groupes. Cette mesure de similarité, appelée aussi une fonction de distance, est indispensable pour effectuer le regroupement, mais peut être assez difficile à définir, en particulier en présence de types de données complexes. Le Clustering diffère de la classification que dans clustering n'existe pas des variables sortantes, n'existe pas de données étiquetées, ce qui signifie que c'est une opération sans supervision [Jul11]. La tâche de clustering ne classifie pas, n'estime pas, ne prévoit pas la valeur d'une variable sortantes. Au lieu de cela, les algorithmes de clustering cherchent à segmenter les données en sous-groupes ou des groupes relativement homogènes, où la similarité des

documents au sein du cluster est maximisée et la similitude de registres à l'extérieur de la grappe est minimisée [Lar14].

2.4.5 La description

Parfois le but du Data Mining est simplement de décrire ce qui se passe sur une base de données compliquée en expliquant les relations existantes dans les données pour en premier lieu comprendre le mieux possible les individu, les produits et les processus présents sue cette

base. Une bonne description d'un comportement implique souvent une bonne explication de celui-ci [LB11, AK05] .

2.5 Principales méthodes de fouille de données

2.5.1 Apprentissage supervisé

En sciences cognitives, l'apprentissage supervisé est une technique d'apprentissage automatique — plus connu sous le terme anglais de machinelearning — qui permet à une machine d'apprendre à réaliser des tâches à partir d'une base d'apprentissage contenant des exemples déjà traités. Chaque élément<(item) de l'ensemble d'apprentissage (training set) étant un couple entrée-sortie. De part sa nature, l'apprentissage supervisé concerne essentiellement les méthodes de classification de données (on connaît l'entrée et l'on veut déterminer la sortie) et de régression (on connaît la sortie et l'on veut retrouver l'entrée).

2.5.1.1 Les arbres de décision

Un arbre de décision est, comme son nom le suggère, un outil d'aide à la décision qui permet de répartir une population d'individus en groupes homogènes selon des attributs discriminants en fonction d'un objectif fixés et connu. Il permet d'émettre des prédictions à partir des données connues sur le problème par réduction, niveau par niveau, du domaine des solutions. Chaque noeud interne d'un arbre de décision porte sur un attribut discriminant des éléments à classifier qui permet de répartir ces éléments de façon homogène entre les différents fils de ce noeud. Les branches liant un noeud à ses fils représentent les valeurs discriminantes de l'attribut du noeud. Et enfin, les feuilles d'un arbre de décision sont ses prédictions concernant les données à classifier. C'est une méthode qui a l'avantage d'être lisible pour les analystes et permet de déterminer les couples <attribut, valeur> discriminants à partir d'un très grand nombre d'attributs et de valeurs.

L'algorithme **ID3 de Quinlan (1986)** , construit un arbre de décision de façon récursive en choisissant l'attribut qui maxime le gain (2.2) d'information selon l'entropie de Shannon (2.1). Cet algorithme fonctionne exclusivement avec des attributs catégoriques et un noeud est créé pour chaque valeur des attributs sélectionnés. ID3 est un algorithme basique facile à implémenter dont la première fonction est de remplacer les experts dans la construction d'un arbre de décision. Cependant, les arbres de décisions ne sont ni robustes, ni compacts ce qui les rends inadaptés aux grosses bases de données. Entropie de Shannon :

$$E(S) = -\sum_{j=1}^{|S|} p(j) log_2 p(j) \tag{2.1}$$

où $p(j)$ est la probabilité d'avoir un élément de caractéristique j dans l'ensemble S.

$$Gain(S, A) = E(S) - \sum^{v} (\frac{|S_v|}{|S|} * E(S_v)) \tag{2.2}$$

où :

S est un ensemble d'entraînement ;

A est l'attribut cible ;

S_v le sous-ensemble des éléments dont la valeur de l'attribut A est v

Algorithme 1 : L'algorithme ID3

```
Données : Exemples, attributCible, AttributsNonCible
si estVide(Exemples) alors
    retourner un noeud Erreur
sinon
    si estVide(AttributsNonCible) alors
        retourner un noeud ayant la valeur la plus représenté pour attributCible
    sinon
        si tous les éléments de Exemples ont la même valeur pour attributCible alors
            retourner un noeud ayant cette valeur
        sinon
            AttributSélectionné = attribut maximisant le gain d'information parmi
             AttributsNonCible
            AttributsNonCibleRestants = AttributsNonCible -AttributSélectionné
            nouveauNoeud = noeud étiqueté avec AttributSélectionné
            pour chaque valeur de AttributSélectionné faire
                ExemplesFiltrés= exempleAyantValeurPourAttribut(Exemples,
                 AttributSélectionné, valeur)
                nouveauNoeud.fils(valeur)=ID3(ExemplesFiltrés, AttributSélectionné,
                 AttributsNonCibleRestants)
            retourner nouveauNoeud
```

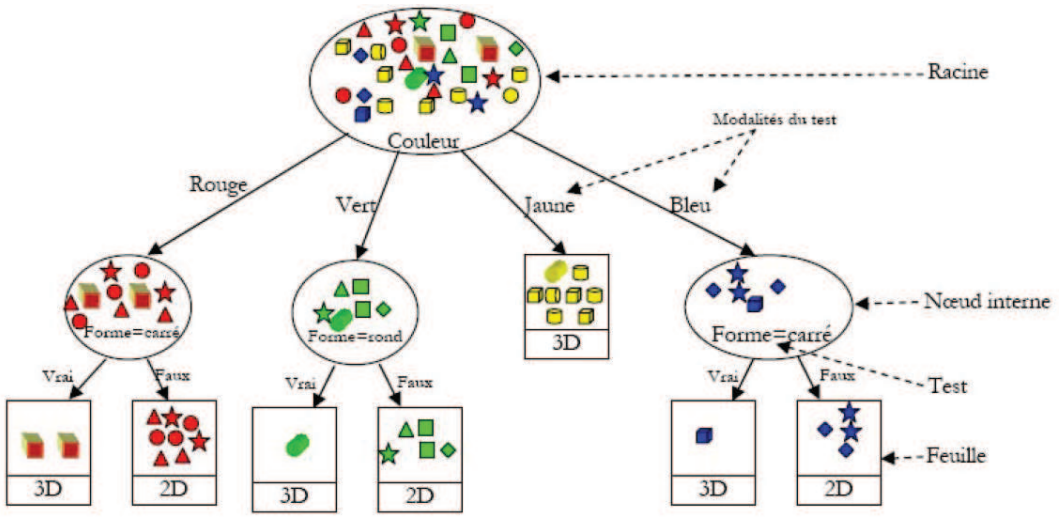


FIGURE 2.2 – Exemple d'arbre de décision

2.5.1.2 K plus proche voisins

La méthode K-NN est une méthode d'apprentissage supervisé. En abrégé k-NN ou KNN, de l'anglais k-nearest Neighbors. L'algorithme des k-plus proches voisins est un des algorithmes de classification les plus simples. Le seul outil dont on a besoin est une distance entre les éléments que l'on veut classifier. Dans ce cadre, on dispose d'une base de données d'apprentissage constituée de N couples « entrée-sortie ». Pour estimer la sortie associée à une nouvelle entrée x, la méthode des k plus proches voisins consiste à prendre en compte (de façon identique) les k échantillons d'apprentissage dont l'entrée est la plus proche de la nouvelle entrée x, selon une distance à définir. Dans un problème de classification, on retiendra la classe la plus représentée parmi les k sorties associées aux k entrées les plus proches de la nouvelle entrée x. [Jam98]

Algorithme 2 : Algorithme des K-plus proches voisins

Données : k //nombre de K-ppv
$X^{train} = (x_1^{train}, x_2^{train}, ..., x_n^{train})$//Données d'apprentissage
$Z^{train} = (z_1^{train}, z_2^{train}, ..., z_n^{train})$//Classes des données d'apprentissage
$X^{test} = (x_1^{test}, x_2^{test}, ..., x_m^{test})$//Données de test
pour $i \leftarrow 1$ **à** m **faire**
  pour $j \leftarrow 1$ **à** n **faire**
    $d_{i,j} \leftarrow dist(x_i^{test}, x_j^{train})$;
    $d_j \leftarrow d_{i,j}$;
  $IndVoisins$ est les indices des voisins
  Z_{voisin} sont les classes des voisins
  Tries d_j par ordre croissant
  pour $k \leftarrow 1$ **à** K **faire**
    $C_k \leftarrow 0$;
  pour $k \leftarrow 1$ **à** K **faire**
    $h \leftarrow z_{IndVoisin_k}^{train}$;
    $C_h \leftarrow z_h + 1$;
  $z_i^{test} = argmax_{k=1}^{K} C_k$
return $Z^{test} = (z_1^{test}, z_2^{test}, ..., z_n^{test})$//*Classes des données de test*

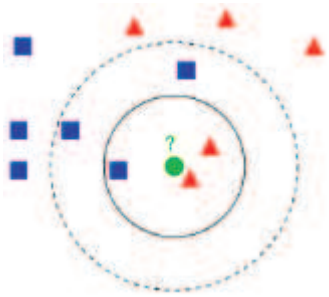


FIGURE 2.3 – Exemple de l'Algorithme K plus proche voisins.

2.5.1.3 Les réseaux de neurones

Un réseau de neurones est un modèle de calcul dont le fonctionnement schématique est inspiré du fonctionnement des neurones biologique. Chaque neurone fait une somme pondérée de ses entrées (ou synapses) et retourne une valeur en fonction de sa fonction d'activation. Cette valeur peut être utilisée soit comme une des entrées d'une nouvelle couche de neurones, soit comme un résultat qu'il appartient à l'utilisateur d'interpréter (classe, résultat d'un calcul, etc.).

La phase d'apprentissage d'un réseau de neurones permet de régler le poids associé à chaque synapse d'entrée (on parle également de coefficient synaptique). C'est un processus long qui doit être réitéré à chaque modification structurelle de la base de données traitée.

Un réseau de neurones artificiels est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement des neurones biologiques. [Jam98] Un réseau de neurones est composé de plusieurs neurones inter-connectés. Un poids est associé à chaque arc. A chaque neurone on associe une valeur. L'objectif est d'obtenir un ensemble de poids permettant de classer correctement (presque tous) les objets du jeu d'apprentissage.

Deux grandes classes de réseaux sont définies :

- Réseaux à propagation directe : c'est le modèle des réseaux PMC perceptron multi-couches (Multi Layer Perceptron, MLP). Les connexions entre les neurones des différentes couches partent dans un seul sens. Nous avons d'abord une couche d'entrée avec seulement des entrées du réseau (pas de neurone), ensuite les couches cachées contenant des neurones cachés, et en dernier la couche de sortie englobant les neurones visibles.
- Réseaux récurrents : exemple réseau de Hopfield ou la machine de Boltzmann. Dans ce type de réseaux les neurones (cachés ou visibles) sont connectés dans n'importe quel sens d'une façon cyclique.

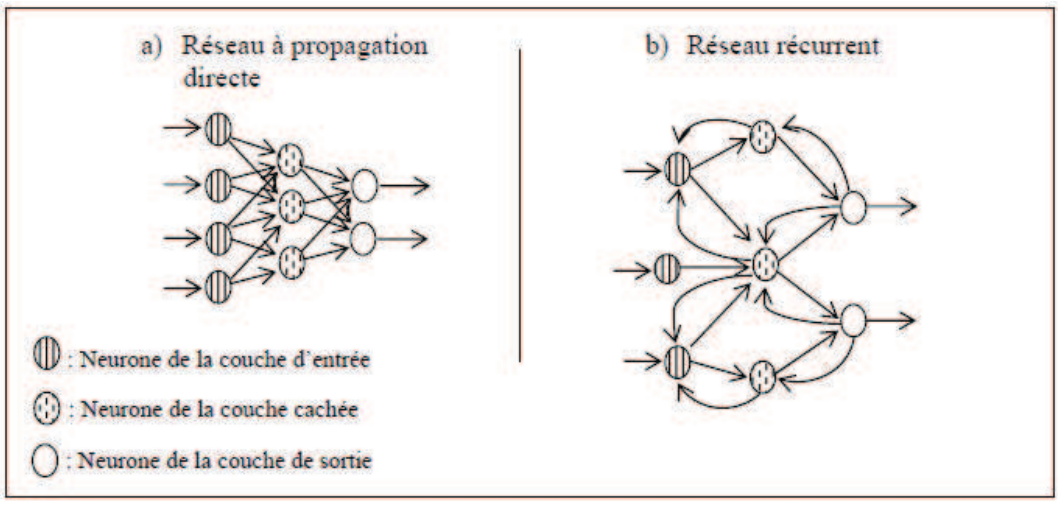


FIGURE 2.4 – Architectures d'un réseau de neurones

Apprentissage des réseaux de neurones artificiels La base d'apprentissage d'un réseau de neurone va contenir l'ensemble des exemples d'étiquetage (x,v), où x représente le vecteur d'entrée réceptionné au niveau de la couche d'entrée, et v c'est le vecteur de sortie désiré sortant de la couche de sortie. L'objectif de l'apprentissage soit supervisé ou incrémental , est d'estimer les poids de connexion du réseau afin de rapprocher la sortie y = f(x) du réseau de la sortie désiré v pour chaque exemple d'apprentissage.

2.5.2 Apprentissage non supervisé

On parle d'apprentissage non supervisé lorsque l'on cherche à extraire des informations nouvelles et originales d'un ensemble de données dont aucun attribut n'est plus important qu'un autre. Le résultat des algorithmes de data mining non supervisé doit être analysé afin d'être retenu pour un usage ou tout simplement rejeté.

2.5.2.1 Clustering

Le clustering est une méthode statistique d'analyse de données qui a pour but de regrouper un ensemble de données en différents groupes homogènes. Chaque sous-ensemble regroupe des éléments ayant des caractéristiques communes qui correspondent à des critères de proximité. Le but des algorithmes de clustering est donc de minimiser la distance intra-classe (grappes d'éléments homogènes) et de maximiser la distance inter-classe afin d'obtenir des sous-ensembles le plus distincts possible. La mesure des distances est un élément prépondérant pour la qualité de l'algorithme de clustering. Cette classe d'algorithmes ne sera pas traitée dans le présent document.

2.5.2.2 Algorithmes des C-moyennes ("Hard C-Means" HCM)

L'algorithme des C-Moyennes a pour objectif de définir un partitionnement de l'ensemble de données $X = (X_1, ..., X_N)$ en C classes. Une donnée X_j est représentée par un vecteur d'attributs $\{x_{j1}, x_{j2}, ..., x_{jn}\}$ dans un espace d'attribut de dimension n.

Une classe W_i est caractérisée par un centre P_i, qui va permettre de calculer le degré d'appartenance u_{ij} de l'objet X_j à cette classe. En finalité, chaque donnée X_j sera attribuée à une et une seule classe W_i parmi les C classes proposées.

Pour cela, à partir des centres des classes, l'algorithme va minimiser la fonction nommée WGSS (Within Group Sum of Squared errors) [3] suivante :

$$J(P, U, X) = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij} d^2(X_j, P_i) \tag{2.3}$$

- $P = (P_1, P_2, ..., P_C)$: l'ensemble des centres de classes, avec

$$P_i = \frac{\sum_{j=1}^{N} u_{ij} X_j}{\sum_{j=1}^{N} u_{ij}} \tag{2.4}$$

- La distance euclidienne entre la donnée X_j et le centre de la classe W_i :

$$d^2(X_j, P_i) = \sum_{k=1}^{N} (x_{jk} - P_{ik})^2 \tag{2.5}$$

- $U = [u_{ij}]$: la partition recherchée de l'ensemble X, c'est la matrice des degrés d'appartenance du modèle de classification. Nous avons :

$$u_{ij} = \begin{cases} 1 & \text{si } d^2(X_j, P_i) < d^2(X_j, P_k) \forall k \neq i \\ 0 & \text{sinon} \end{cases} \tag{2.6}$$

L'algorithme des C-moyennes est illustré comme suit :

Algorithme 3 : Algorithme HCM

Données : C, X

Résultat : P, W

Initialisation : Définir partition U_0 ;

répéter

a. Evaluation des regroupements :

Calculer les centres Pi ;

b. Adaptation des prototypes :

calculer les nouveaux degrés $u_{ij}^{(t)}$ et mettre à jour la partition U^t ;

jusqu'à $\Delta U < \epsilon$;

2.5.2.3 Algorithme C-moyennes floues ("Fuzzy C-Means" FCM)

Developpe par Dunn en 1973 et approuve par Bezdek en 1981, c'est une méthode de classification non supervisée qui permet d'associer une donnée a une ou plusieurs classes en même temps. L'appartenance d'une donnée à une classe est définie par un degré d'appartenance qui représente l'élément de base d'une partition U définie en (section 4.1). L'objectif de FCM est de construire une matrice de C-partition flou U : CxN, ayant des éléments numériques dans l'intervalle [0,1], ces éléments représentent les degrés d'appartenance u_{ij} des données X_j aux classes W_i.

L'objectif de l'algorithme FCM est défini par la minimisation de la somme pondérée des carres des distances entre les données a regrouper et les centres de classes :

$$J(P, U, X) = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^m d^2(X_j, P_i) \tag{2.7}$$

Ou :

- $m \in [1, \infty]$: est l'indice du flou (fuzzy index) qui détermine le degré du flou de la partition obtenu. Le choix de l'indice m est souvent compris entre 1.1 et 3, autrement, le théorème de Tuker suggère de prendre $m \geq N/(N-2)$, N étant le nombre de données. Tant que m tend vers ∞ la partition est de plus en plus flou.

- Dans cet algorithme la distance euclidienne entre la donnée et le centre de classe est définie par :

$$d^2(X_j, P_i) = (X_j - P_i)^T A(X_j - P_i) \tag{2.8}$$

Avec :

$$P_i = \frac{\sum_{j=1}^{N} (u_{ij})^m X_j}{\sum_{j=1}^{N} (u_{ij})^m} \tag{2.9}$$

et $A(NxN)$: est une matrice définie positive.

- Les degrés d'appartenances sont calculés par la manière suivante :

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{d^2(X_k, P_i)}{d^2(X_j, P_i)} \right)^{\frac{1}{(m-1)}}} \tag{2.10}$$

L'algorithme des C-moyennes floues est illustre comme suit :

Algorithme 4 : Algorithme FCM

Données : C, m, X

Résultat : P, W

Initialisation : Définir partition U_0 ;

répéter

a. Evaluation des regroupements :

Calculer les C centres Pi ;

b. Adaptation des prototypes :

calculer les nouveaux degrés $u_{ij}^{(t)}$ et mettre à jour la partition U^t ;

jusqu'à $\Delta U < \epsilon$;

2.5.2.4 Algorithmes de C-moyennes possibilistes ("Possibilistic C-means" PCM)

Cet algorithme est une modélisation possibiliste de l'algorithme C-moyennes introduite par Krishnapuram et Keller pour améliorer les performances de classification en présence du bruit. Il est basé sut la théorie de possibilité pour définir la partition U des données X sur l'ensemble des classes W représentées par leur centres P. En effet, les degrés d'appartenances u utilisés en FCM sont traduits en PCM autant que degrés de vérité relatifs, décrivant l'appartenance d'une donnée à chacune des classes possibles. La nouvelle formulation de la fonction objective à minimiser est décrite par :

$$J(P, U, X) = \sum_{i=1}^{C} \sum_{j=1}^{N} (u_{ij})^m d^2(X_j, P_i) + \sum_{i=1}^{C} n_i \sum_{j=1}^{N} (1 - u_{ij})^m \tag{2.11}$$

Où :

- n_i : est le carré de la distance entre P_i (centre de la classe W_i) et l'ensemble des données X_j ayant leur degrés d'appartenance (à la classe W_i) $uij = 0.5$; C'est un paramètre positif utilisé pour évaluation des degrés d'appartenance possibles aux classes. Il peut être

défini de plusieurs manières, soit estimer à l'initialisation de l'algorithme ou recalculer pendant l'itération de l'algorithme par la formule suivante :

$$n_i = \frac{\sum_{j=1}^{N} (u_{ij})^m d^2(X_j, P_i)}{\sum_{j=1}^{N} (u_{ij})^m} \tag{2.12}$$

- Les centres de classes Pi sont calculés da la même manière dans FCM ;
- Contrairement au FCM, où la somme des degrés d'appartenance doit égale à 1, en PCM, la somme des possibilités d'appartenance de données aux classes n'est pas forcément égale à 1.
  La mise à jour des valeurs de possibilité d'appartenance est comme suit :

$$u_{ij} = \frac{1}{1 + \left(\frac{d^2(X_j, P_i)}{n_i}\right)^{\frac{1}{(m-1)}}} \tag{2.13}$$

- Dans [45], le bon choix de l'indice m est conseillé par Krishnapuram et Keller à ce qu'il soit $= 1.5$, contrairement à 2 dans le FCM ;

Le premier terme de la nouvelle fonction PCM correspond au critère du FCM. Le deuxième terme représente la nouveauté apporté par le PCM dont sa minimisation va contribuer pour avoir des degrés d'appartenance de plus

2.5.2.5 Les règles associatives

Les règles associatives sont des règles qui sont extraites d'une base de données transactionnelles (itemset) et qui décrivent des associations entre certains éléments. Cette technique permet de faire ressortir les associations entre les produits de base (les produits essentiels, ceux pour lesquels le client se déplace) et les produits complémentaires, ce qui permet de mettre en place des stratégies commerciales visant à accroître les profits en favorisant, par exemple, les ventes complémentaires. Ces algorithmes permettent de résoudre des problèmes dits de Frequent Set Counting (FSC).

2.5.2.6 Sequence mining

Le sequence mining concerne la détection de motifs dans les flux de données dont les valeurs sont délivrées par séquences. Cette technique est particulièrement utilisé en biologie pour l'analyse de gènes et des protéines mais également afin de faire du text mining, les phrases étant considérées comme des séquences ordonnées de mots.

2.5.3 Apprentissage incrémental

L'apprentissage incrémental permet à une machine d'apprendre par ajout successif d'informations. Pour être considéré comme tel, un système d'apprentissage doit :

- être capable t'apprendre de nouvelles informations à partir de nouvelles données ;
- être capable de se passer des données d'origine pour entraîner le nouveau classifieur ;
- préserver le savoir précédemment acquis ;
- être capable de reconnaître de nouvelles classes introduites dans les nouvelles données.

2.6 Les mesures de qualité d'un Clustering

2.6.1 La similarité

- La mesure de dissimilarité DM : plus la mesure est faible plus les points sont similaires (distance)
- La mesure de similarité SM : plus la mesure est grande, plus les points sont similaires

2.6.2 La distance

On appelle **distance** sur un ensemble E toute application d définie sur le produit $E^2 = E \times E$; et à des valeurs réels positifs dans l'ensemble $\Re^+$: $d : E \times E \to \Re^+$

Et qui vérifie les propriétés suivantes :

- symétrie : $\forall (a, b) \in E^2, d(a, b) = d(b, a)$
- séparation : $\forall (a, b) \in E^2, d(a, b) = 0 \Leftrightarrow a = b$
- inégalité triangulaire : $\forall (a, b, c) \in E^3,\ d(a, c) \leq d(a, b) + d(b, c)$

1. Mesure de la distance $d(x, y)$ entre 2 points x et y dans un espace vectoriel E dans $\Re^n$:
Plusieurs mesures de distance peuvent être utilisé dans ce cas telles que :

- La distance Euclidienne : $\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
- La distance de Manhattan : $\sum_{i=1}^{n} |x_i - y_i|$
- La distance de Minkoswski : $\sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}$
- La distance de Mahalanobis : $\sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\sigma_i^2}}$

2. Mesure de la distance $d(x_1, x_2)$ entre 2 points x_1 et x_2 à valeurs discrètes :
On peut utiliser :

- Une matrice de contingence.
- La distance de Hamming.
- Distance de Tanimoto,

3. Mesure de la distance $d(C_1, C_2))$ entre 2 classes C_1 et C_2 :
Peut être calculé a l'aide de :

- La méthode du plus proche voisin : $Min\{d(x_i, x_j), x_i \in C_1, x_j \in C_2\}$
- La distance des centres de gravité des classes : $d_2(G_{C_1}, G_{C_2})$
- La distance de Ward : $\frac{P_{C_1} * P_{C_2}}{P_{C_1} + P_{C_2}} d_2(G_{C_1}, G_{C_2})^2$

2.7 Comparatif des Méthodes par type

Nous récapitulons dans le Tableau 2.2 les avantages et les inconvénients des méthodes les plus courantes de fouille de données :

TABLE 2.2 – Classification des Méthodes par type

Type	**Famille**	**Sous-Famille**	**Algorithme**
méthodes descriptives	modèles géométriques	analyse factoriel (projection et visualisation dans un espace de dimension inférieure)	Analyse en Composantes principales ACP (variables continues)
			Analyse Factorielle des Correspondances AFC (variables quantitatives et binaires)
			Analyse des Correspondances Multiples ACM (variables quantitatives et binaires)
		analyse typologique (regroupement dans tout l'espace en classes homogènes)	Méthode de partitionnement (centres mobiles, k-means,k-medoids...)
			Méthodes hiérarchiques (ascendantes, descendantes)
		analyse typologique + réduction de dimension	Classification neuronale (réseaux de kohonen)
	Méthode combinatoires		Classification par agrégation des similarités (variables qualitatives)
	Modèle à base de règles logiques	Détection des liens	Recherche d'associations
			Recherche de séquences similaires
méthodes prédictives	Modèle à base de règles logiques	arbres de décisions	arbres de décisions(variable à expliquer continue ou qualitative)
	Modèles à base de fonctions mathématiques	Réseaux de neurones	Réseaux à apprentissage supervisé (perceptron, réseau à fonction radiale de base ...)
		Modèles paramétriques ou semi paramétriques	régression linéaires, ANOVA, MANOVA, ANCOVA, MANCOVA (variable à expliquer continue)
			Analyse discriminante de Fisher, régression logistique (variable à expliquer qualitative)
			Modèle log-linéaire (variable à expliquer qualitatives)
			modèle linéaire généralisé GLZ (variable à expliquer continue, discrète, comptage ou qualitative)
	Prédiction sans modèle	analyse probabiliste	k-plus proche voisins (k-nn)

2.8 Conclusion

La fouille de données est l'exploration et l'analyse de grandes quantités de données afin d'y découvrir de l'information implicite et par la suite d'extraire des connaissances représentant un support d'aide à la prise de décision. Avec le web, l'explosion des volumes des données ainsi que leurs natures a pausé un sérieux problème au système de fouille de données traditionnels existants, et par conséquent l'imagination de nouvelles technologies et méthodes répondant aux besoins et permettant l'exploitation de ces volumes de données est indispensable.

Chapitre 3

Méthodes de découverte de communautés dans les réseaux complexes

Sommaire

3.1 Introduction

Beaucoup de systèmes complexes du monde réel peuvent être représentés et étudiés en tant que réseaux. Les réseaux complexes recouvrent ainsi des réseaux aussi divers que le réseau Internet, les réseaux des contacts sociaux entre individus [Sco12], les réseaux des réactions chimiques entre protéines dans le métabolisme d'un être vivant [HHJ03, JTA$^+$00], les réseaux des pages web [AJAL99] qui contiennent plusieurs millions de noeuds, les réseaux trophiques [WM00], les réseaux de dictionnaires [BS02] ... et bien d'autres.

Les études menées sur la signification physique et les propriétés mathématiques des réseaux complexes ont constaté que ces réseaux partagent des propriétés macroscopiques. Parmi ces propriétés, on cite des propriétés prototypes telle que l'effet petit-monde [WS98] et l'échelle-libre [BA99], des propriétés dynamiques tel que la diffusion [BP01, ESMS03] et des propriétés structurelle comme la structure de communauté [Moo01, FLGC02, GN02, LN04, KFM$^+$03, GA05, PDFV05]. La propriété de structure de communautés paraît être commune à beaucoup de réseaux complexes et permet de comprendre la relation entre un simple noeud dans la microscopie et des groupes dans la macroscopie. Par conséquent, la découverte de structure de communautés a fait l'objet de plusieurs récents efforts. Il s'agit d'une problématique proche des problématiques classiques de clustering de données et de partitionnement de graphe.

Les méthodes de découverte de communautés supposent que le réseau se divise naturellement en un ensemble de sous-groupes et visent la détection de ces groupes (communautés). Les critères utilisés pour détecter correctement la structure de communautés sont très cruciaux et divers ce qui justifie le nombre important des méthodes de découverte de communautés proposées. Ce chapitre est un état de l'art des méthodes existant pour la détection de communautés [DBSM16].

3.2 Contexte et motivations

3.2.1 La théorie des graphes classique est-elle appropriée aux réseaux du monde réel ?

L'étude des réseaux sous la forme de théorie des graphes est l'un des piliers fondamentaux des mathématiques discrètes. La résolution d'Euler, en 1735, du problème de ponts de Königsberg est considérée comme le premier théorème de la théorie des graphes. Au 20 ème siècle la théorie des graphes s'est développée en tant que domaine substantiel de la connaissance et les graphes sont également devenus extrêmement utiles comme représentation d'une grande variété de systèmes dans différents secteurs tels que les réseaux biologiques, sociaux, technologiques, et de l'information. Ainsi, l'analyse des graphes est devenue cruciale pour comprendre ces réseaux du monde réel. Les réseaux ont été également étudiés intensivement dans les sciences sociales en se basant sur l'usage des graphes dont les sommets représentent les individus ou les organisations sociales et les liens désignent les interactions sociales entre eux. Des études sont, par exemple, menées sur les propriétés de centralité et de connectivité.

Ces dernières années, vue la disponibilité croissante des données à grande échelle, l'étude des réseaux a été changé de l'analyse des simples graphes et des propriétés des sommets individuels à l'analyse des propriétés statistiques des graphes complexes. La théorie des graphes classique a été concernée par des problèmes des réseaux réels mais son approche qui est orientée vers la conception n'est pas approprié aux réseaux surgissant dans le monde réel. Plusieurs questions qui ont été précédemment posées dans les études de petits réseaux ne peuvent pas être utile dans des grands réseaux, par exemple, l'analyste d'un réseau social pourrait demandé : "quel noeud affecte-il la connectivité du réseau s'il est retiré ?", mais une telle question a peu de signification dans des réseaux qui contiennent des millions de sommets (car dans des tels réseaux la suppression d'un seul sommet n'aura aucun effet). Désormais, la question qui devrait être posée : " Quel est le pourcentage des sommets a enlevé pour affecter considérablement la connectivité du réseau ?" et ce type de questions statistiques a une concrète signification dans les réseaux du monde réel.

C'est ainsi qu'un groupe divers de scientifiques, y compris des mathématiciens, physiciens, informaticiens, sociologues, et biologistes, avaient activement poursuivi ces questions et avaient fondé le nouveau champ de la théorie des réseaux, ou la "science des réseaux" [AL02, Buc03, Wat04]. Une littérature significative s'est déjà accumulée dans ce nouveau domaine interdisciplinaire qui se penche sur l'étude et la découverte des propriétés que partagent un grand nombre de grands réseaux complexes [WS98].

3.2.2 Quelle opportunité y a-t-il pour une nouvelle science interdisciplinaire des réseaux ?

Cette science se distingue des travaux précédents sur les réseaux de trois manières importantes :

- Elle se focalise sur les propriétés des réseaux du monde réel telles que la longueur des chemins, le degré de distribution, et le comportement du système pour proposer des mesures appropriés à ces propriétés.
- Elle vise l'extraction des modèles qui permettent la compréhension approfondie des propriétés des réseaux du monde réel.
- Elle étudie la prédiction de la dynamique de comportement des systèmes en considérant les propriétés du réseaux complexes influant les différents acteurs des réseaux.

3.2.3 Quelles sont les propriétés que partagent un grand nombre de réseaux complexes ?

Les réseaux complexes que l'on peut rencontrer dans les différentes disciplines n'ont, à première vue, pas de raison de se ressembler. Cependant, plusieurs études ont révélé l'existence de caractéristiques communes et significatives [WS98, Str01, AB02, New03b, DM02]. Nous citons brièvement les propriétés communes les plus étudiées dans les réseaux complexes :

3.2.3.1 L'effet petit monde "The small-world effect"

L'effet petit monde tient son nom de l'expression populaire "le monde est petit" désignant la surprise de constater que deux connaissances d'un même individu, a priori sans rapport, se connaissent entre elles. La notion petit monde est définie, dans certains articles [WS98] comme la combinaison d'un fort coefficient de clustering et d'un petit diamètre. Cette propriété étudiée par le psychologue Milgram [Mil67] est vérifiée par le modèle de graphes aléatoires d'Erdös-Rényi [ER59]. Pour pallier aux limites de modèle d'Erdös-Rényi, plusieurs travaux ont été publiés [Bol98].

3.2.3.2 Clustering "Transitivity or clustering"

Une des propriétés essentielles des réseaux complexe est l'existence d'une forte densité locale qui s'oppose à la faible densité globale du graphe. Cette densité est souvent mesuré par le coefficient de clustering [WS98]. Il s'agit de la moyenne, sur tous les nœuds u, du ratio du nombre de voisins de u qui sont reliés entre eux sur le nombre total de liens qui pourraient potentiellement exister entre ces voisins (probabilité que deux voisins de u soient reliés). Cette propriété illustre la tendance des acteurs à se regrouper en modules ou communautés.

3.2.3.3 Distribution des degrés "Degree distributions "

Le modèle usuel utilisé au départ pour ce domaine d'étude était un réseau aléatoire uniforme, sur lequel on observe un effet de seuil pour la transmission d'un virus, c'est-à-dire qu'en dessous d'une fraction d'individus infectés, le virus cesse de se répandre. Mais une étude similaire menée sur un modèle présentant une distribution de degrés en loi de puissance a donné des résultats différents, en particulier l'effet de seuil disparaît. En 1999, Faloutsos et

al [FFF99] ont observé que le réseau Internet présentait cette propriété. Une telle observation a donc remis en cause les mécanismes mis en place pour freiner la propagation des virus et les modèles utilisés jusqu'alors. Des modèles suivant une loi de puissance sont donc utilisés, car c'est cette distribution qui est retrouvée la plupart du temps dans tous les réseaux réels [New03b]. Deux distributions de degrés sont connues : une distribution homogène des degrés des noeuds (selon une loi de Poisson), et une distribution hétérogène des degrés des noeuds (selon une loi de Puissance). La distribution selon une loi de puissance est donnée comme suit : le nombre P_k de sommet de degré k est proportionnelle à $k^{-\alpha}$, pour une constante $\alpha > 0$ sur un intervalle de plusieurs ordres de grandeur (par exemple entre $k = 10$ et $k = 10^6$).

3.2.3.4 Résilience des réseaux "Network resilience"

La propriété de résilience des réseaux est liée à la distribution de degrés. Quand il y a une suppression de sommets, la longueur des chemins augmentera, en conséquence, des paires de sommets devenus déconnectées et la communication entre elles deviendra impossible. Le niveau de résilience de réseau se varié selon la connectivité des sommets. Un intérêt particulier pour l'étude de la résilience de réseau a été soulevé par le travail d'Albert et al [AJAL00].

3.2.3.5 Mixing patterns

Dans la plupart des réseaux il existe des types différents de sommets et la probabilité qu'il existe un lien entre une paire de sommets différentes dépend souvent du type de la relation. Autrement dit, mixing patterns se réfère aux tendances systématiques d'un type de nœuds dans un réseau pour se connecter à un autre type par exemple Maslov et al [MSZ04] ont étudié l'existence de trois type de nœud dans le réseau Internet : les fournisseurs qui ont une forte connectivité, les consommateurs qui sont les utilisateurs finaux, et les providers de services Internet qui jouent le rôle de relais entre les deux types de nœuds précédents.

3.2.3.6 Degré de corrélation "correlation degree"

La corrélation de degré peut fournir des détails intéressants sur la structure du réseau. En fait, cette propriété nous permet de savoir si les sommets ayant un degré élevé sont de préférence connectés à d'autres sommets avec un degré élevé, ou sont plutôt connectées à des sommets ayant un faible degré. Plusieurs études ont été proposées pour quantifier le degré de corrélation à l'exemple des travaux de Maslov et al [MSZ04, MS02].

3.2.3.7 Navigabilité "Network navigation"

Kleinberg [Kle00] a proposé le premier modèle de petit monde présentant la propriété de navigabilité, c'est-à-dire le premier modèle de graphe dont le diamètre est polylogarithmique en nombre de noeuds et dont des chemins polylogarithmiques peuvent être découverts par un algorithme décentralisé entre tout couple de sommets. A l'exemple de la navigation à travers

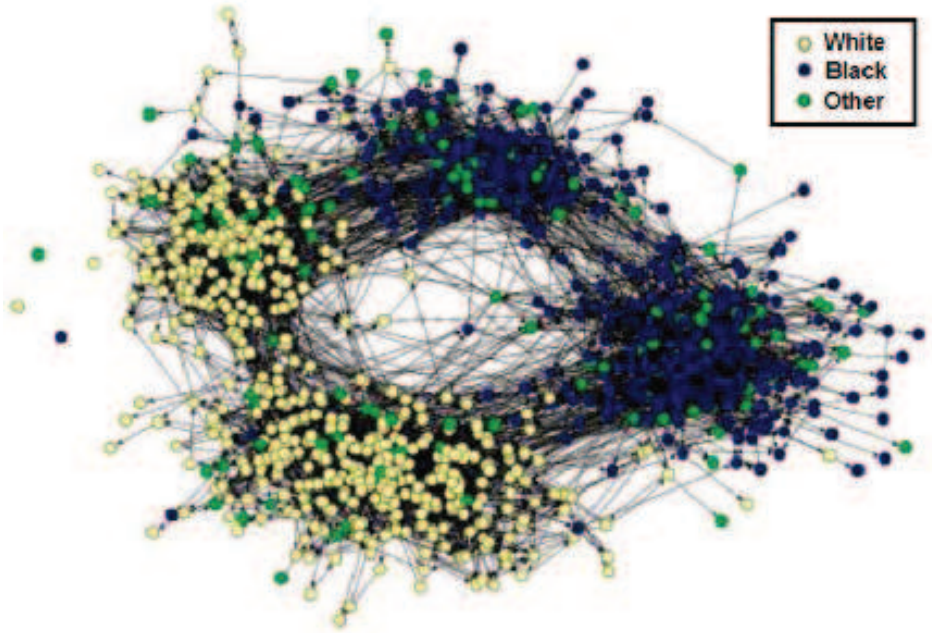


FIGURE 3.1 – Réseau d'amitié des enfants dans une école aux USA. Les liens d'amitiés sont déterminées par les participants, et par conséquent ils sont orientés. Les sommets sont colorés selon la race.

le réseau des pages Web qui se faisait d'une page à l'autre sans connaître la carte globale du réseau [AJAL99, Kai99]. Par ailleurs, la découverte des chemins de façon décentralisée est très rentable pour les réseaux d'interactions qui contiennent un très grand nombre de noeuds car une recherche classique des plus courts chemins est très coûteuse en temps.

3.2.3.8 Structure de communautés "Community structure"

Dans les réseaux complexes, la présence de groupes de sommets fortement liés entre eux et faiblement liés avec l'extérieur fonde la propriété de structure de communautés. Les communautés sont des groupes de sommets qui partagent probablement des propriétés communes ou des rôles semblables dans le réseau complexe. Ainsi, les communautés peuvent correspondre aux groupes de pages Web traitant le même sujet [FLGC02], aux modules fonctionnels dans les réseaux métaboliques [GA05, PDFV05], aux groupes d'individus dans les réseaux sociaux [GN02, LN04], et aux subdivisions dans les chaînes alimentaires [Pim79, KFM$^+$03], ...ect. La figure 3.1 montre une visualisation du réseau d'amitié des enfants dans une école au USA selon l'étude de Moody. Moody [Moo01] a coloré les sommets selon la race de chaque individu. Ce réseau semble avoir une forte structure de communauté qui en résulte principalement à cause de la propriété de race des individus.

3.2.4 Quels sont les objectifs du processus de découverte de communautés dans les réseaux complexes ?

Les principaux objectifs qui ont motivés les études sur les méthodes de découverte de communautés dans les réseaux du monde réel sont les suivants :

- La détection de communautés est un outil important pour la compréhension des structures et des fonctionnements des systèmes complexes.
- Les communautés donnent un point de vue macroscopique sur la structure des graphes. Elles permettent par exemple de regrouper et d'identifier les sommets qui jouent des rôles similaires. Par exemple, la détection de communautés dans le graphe du Web est une piste envisagée pour améliorer les moteurs de recherche [FLGC02].
- La détection de communautés peut aussi être utilisée pour la visualisation des graphes complexes [ACJ+03].
- Les méthodes de détection de communautés sont utilisées pour le partitionnement du graphe afin d'effectuer des calculs séparés moins coûteux sur chaque communauté. Ce procédé de parallélisme permet d'envisager des gains en complexité du temps pour les grands graphes.
- La détection de communautés permet une classification des sommets, selon leur position topologique dans le graphe. Ainsi, les sommets de position centrale dans leur clusters partagent un grand nombre de liens avec les autres groupes, en conséquence, ils peuvent avoir une importante fonction de contrôle et de stabilité au niveau du groupe. Quand aux sommets de frontière, ils jouent un rôle important de relais entre les différentes communautés. Une telle classification est très utiles dans les réseaux sociaux et les réseaux métaboliques [Gra73, Bur76, Fre77].
- Les communautés peuvent être utilisées pour améliorer les méthodes de compression de graphes à l'exemple du travail [MKNG06].

3.3 Description des communautés

Une définition naturelle des communautés stipule qu'une communauté est dense, c'est-à-dire que ses membres sont fortement connectés entre eux et que, dans le même temps, ils sont peu liés à des membres en dehors de la communauté. Le problème de la détection de communautés est donc naturellement formalisé en la recherche d'une partition d'un graphe en sous-groupes denses peu connectés entre eux. Définir les communautés comme les parties d'une partition est fréquemment admis, mais cela implique qu'un nœud n'appartient qu'à une et une seule communauté. Il existe des définitions de communautés où un nœud peut appartenir à diverses communautés. On appelle de telles communautés des communautés recouvrantes. Néanmoins, aucune définition ne fait aujourd'hui consensus, il existe peu d'algorithmes pour les détecter

3.3.1 Définition des communautés

En dépit de la grande quantité d'étude dans ce domaine, un consensus sur ce qui est la définition d'une communauté n'a pas été atteint. Conceptuellement, les définitions de communauté se basent sur la notion de sous graphe et peuvent être séparées en deux catégories : les définition comparatives et les définition de référence individuel. Dans ce qui suit, nous en citons quelques exemples :

3.3.1.1 Définitions comparatives

La comparaison est effectué le plus souvent en terme de liens internes et externes dans chaque communauté et parfois des auteurs comparent des critères de similarités pour pouvoir détecter la structure de communautés.

Définition 1 : [WF94]

Une communauté peut être décrite comme collection de sommets dans un graphe qui sont fortement reliés entre eux-mêmes mais faiblement relié du reste du graphe.

Définition 2 : [RCC+04]

Soit A_{ij} la matrice d'adjacence du graphe G ; Le degré k_i d'un noeud $i \in G$ est :

$$k_i = \sum_{j \in G} A_{ij} \tag{3.1}$$

Soit un sous graphe $V \subset G$ et $i \in V$, le degré total est donné par :

$$k_i(V) = k_i^{in}(V) + k_i^{out}(V) \tag{3.2}$$

Tel que :

$k_i^{in}(V) = \sum_{j \in V} A_{ij}$: est le nombre de liens reliant le noeuds i à d'autres noeuds appartenant à V ;

$k_i^{out}(V) = \sum_{j \notin V} A_{ij}$: est le nombre de liens vers les noeuds qui n'appartiennent pas à V (le reste du réseau).

Définition d'une communauté au sens fort :

Le sous-graphe V est une communauté au sens fort si :

$$k_i^{in}(V) > k_i^{out}(V), \forall i \in V \tag{3.3}$$

Une communauté est définie en tant qu'un ensemble de noeuds dans lequel chaque noeud a plus de connexions au sein de cette communauté qu'avec le reste du réseau.

Définition d'une communauté au sens faible :

Le sous graphe V est une communauté au sens faible si :

$$sum_{i \in V} k_i^{in}(V) > \sum_{i \in V} k_i^{out}(V) \tag{3.4}$$

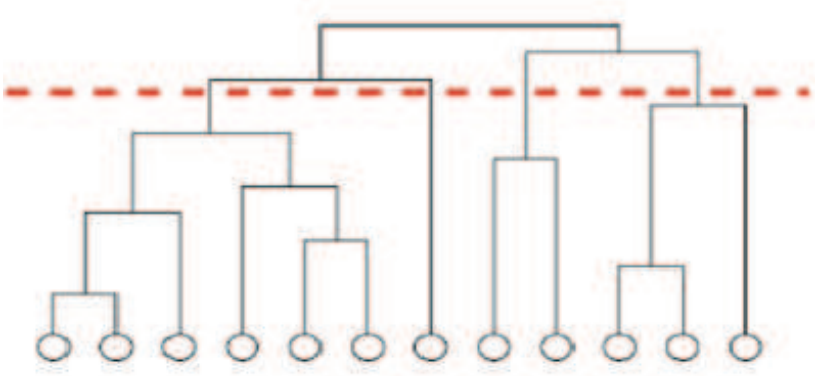


FIGURE 3.2 – Dendrogramme d'un algorithme de détection de communautés

Une communauté est définie comme un ensemble de noeuds dont le nombre total de liens internes est supérieur au nombre total des liens vers l'extérieur.

Définition 3 :

Les communautés sont des groupes de sommets qui sont similaires les uns aux autres. Un critère est choisi pour l'évaluation de la similarité.

3.3.1.2 Définitions de référence individuelle

Définition 1 : La communauté est une clique, définie en tant que sous-groupe d'un graphe contenant plus de deux noeuds où tous les noeuds sont reliés entre eux au moyen de liens dans les deux directions (c'est un sous graphe entièrement connecté). Les triangles sont les cliques les plus simples, et sont fréquentes dans les réseaux du monde réel mais les plus grandes cliques sont rares, ainsi elles ne sont pas de bons modèles de communautés. En outre, l'utilisation de l'algorithme de Bron-Kerbosch [BK73] pour trouver les cliques résulte d'un coût de calcul élevé (complexité exponentielle).

Définition 2 : [New06]

Une communauté est un sous graphe indivisible.

3.3.2 Représentation graphique des communautés

La théorie des graphes est employée pour représenter le réseau et même les communautés dans le réseau en question. Les Dendrogrammes sont aussi souvent utilisés pour illustrer la progression entière de l'algorithme de découverte de communautés et le regroupement des sommets depuis le graphe initial au graphe résultant partitionné en communauté comme le montre la figure 3.2. Les coupes horizontales à travers l'arbre hiérarchique représentent clairement toutes les divisions possibles en communautés à chaque niveau.

3.3.3 Mesures de la qualité de partition d'un réseau en communauté

Comment savoir si les communautés détectées sont bonnes ou non et comment évaluer une telle partition ? Quelle est la meilleure partition pour le réseau en question ? A quel niveau on coupe le dendrogramme pour obtenir la partition adéquate du réseau ou bien le nombre de communautés appropriées ? Pour répondre à ces questions, Newman et al [NG04] [New03a]. ont introduit une mesure de la qualité de partition du réseau appelé "modularité".

Supposons une partition particulière d'un réseau en k communautés. Soit e une matrice symétrique $k \times k$, ses éléments e_{ij} représente la fraction de tous les liens dans le réseau qui relient les sommets de la communauté i aux sommets de la communauté j.

La trace de la matrice e : $Tr_e = \sum_i e_{ii}$ représente la fraction de tous les liens qui relient les sommets dans les mêmes communautés. Une valeur élevée de la trace indique une bonne partition en communautés.

La somme de n'importe quelle ligne (ou colonne) de e : $a = \sum_j e_{ij}$ correspond à la fraction de tous les liens reliés aux sommets de la communauté i.

Si le réseau ne possède pas la propriété de structure de communauté, la valeur prévue des fractions des liens dans une partition peut être estimée. C'est la probabilité qu'un sommet d'extrémité d'un lien soit dans la communauté i, donc a_i, multiplier par la fraction des liens qui se termine par un sommet dans la communauté i, donc a_i . On peut alors écrire : $e_{ij} = a_i.a_i$, ce qui représente le nombre des liens intra-communautés prévus.

Ainsi, la mesure de modularité est définit comme suit :

$$Q = \sum_i (e_{ii} - a_i^2) = Tr_e - \|e^2\| \tag{3.5}$$

La modularité permet de comparer deux partitions d'un même graphe mais pas vraiment des partitions de graphes différents. Elle n'est pas une mesure absolue de qualité, dans le sens où la meilleure partition pour un graphe n'aura pas la même modularité que la meilleure partition pour un autre graphe. Cependant, il est possible que les partitions de meilleures modularités ne correspondent pas aux partitions en communautés les plus pertinentes [FB07].

3.4 Domaines d'application des méthodes de découverte de communautés

La nature interdisciplinaire de la nouvelle théorie de réseaux vient de la diversité des réseaux du monde réel. Ces réseaux complexes possèdent des propriétés communes et soulèvent des problématiques similaires. Une de ces problématiques est la découverte d'une structure significative de communautés qui constitue un backbone fondamental pour bien comprendre les interactions des réseaux complexes. Dans cette section, nous citons quelques exemples des réseaux complexes qui sont caractérisés par une structure de communautés.

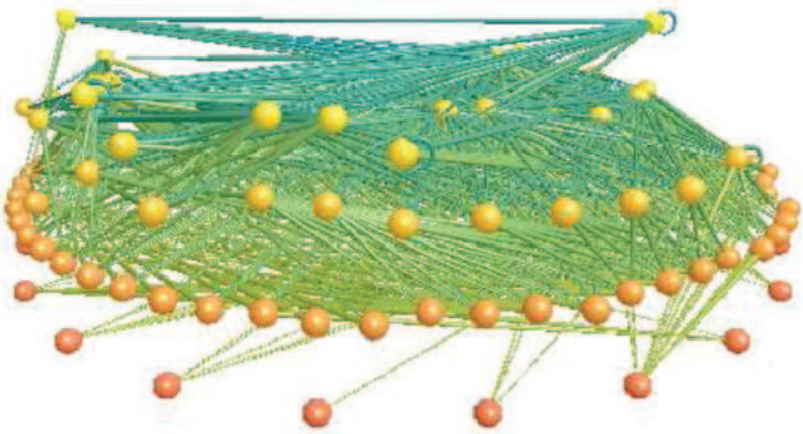


FIGURE 3.3 – Une chaîne alimentaire des interactions de prédateur-proie entre les espèce dans un lac [Mar91].

3.4.1 Réseaux sociaux

Les réseaux sociaux constituent un champ d'application ancien et important [WF94] dans lequel les acteurs sont des individus ou entités sociales (associations, entreprises, pays,...etc)et les liens entre eux peuvent être de différentes natures. Il existe plusieurs types de réseaux sociaux : les réseaux de connaissance (deux individus sont reliés s'ils se connaissent), les réseaux de collaboration (deux individus sont reliés s'ils ont travaillé ensemble) , en particulier, de nombreux travaux ont étudié les collaborations scientifiques [New01]), les réseaux d'appels téléphoniques [Res00] (deux individus ou numéros de téléphones sont reliés s'il y a eu un appel entre eux), les réseaux d'échanges (deux entités sont reliées si elles ont échangé un fichier [GLLB04] ou un courrier électronique [EMB02] par exemple), ...etc.

3.4.2 Réseaux biologiques

Les réseaux biologiques sont assez divers parmi lesquels il existe les réseaux métaboliques [JTA+00] (les sommets sont des gènes ou des protéines qui sont liés selon leurs interactions chimiques), les réseaux de neurones (chaque neurone est connecté à plusieurs autres neurones) ou les réseaux trophiques [WM00] (les espèces d'un écosystème sont reliées pour représenter les chaînes alimentaire). Un exemple d'une chaîne alimentaire [Mar91] est illustré sur la figure 3.3.

3.4.3 Réseaux d'information

L'exemple classique du réseau d'information est le réseau de citation des travaux scientifiques. La plupart des articles citent les travaux précédents des auteurs sur le même sujet (voir **??**). Ces citations forment un réseau dont les sommets sont des articles ; un lien orienté de l'article A vers l'article B indique que A cite B. La structure d'un réseau de citation

reflète la structure de l'information stockée dans ses sommets. Un autre exemple très important du réseau d'information est le réseau World Wide Web, dont les pages Web contenant l'information, reliée ensemble par les lien hypertextes d'une page vers l'autre [ER90].

3.4.4 Réseaux technologiques

Les réseaux technologiques sont des réseaux synthétiques conçus typiquement pour la distribution d'un certain produit ou ressource, telle que l'électricité. La grille d'énergie électrique est un bon exemple des réseaux technologiques. Plusieurs études statistiques ont été menées sur la grille d'électricité par Watts et Strogatz [WS98, Wat99] et Amaral et al [ASBS00]. Nous pouvons aussi citer d'autres réseaux de distribution tels que le réseau des itinéraires de ligne aérienne [ASBS00], les réseaux des routes [KSM03], réseau de chemins de fer [SDC+03, LM02], et les réseaux sans fil mobiles [DBS14]. Les réseaux de fleuve sont aussi considérés comme des réseaux de distribution [DR00].

3.4.5 Réseaux linguistiques

Ces réseaux relient les mots d'un langage donné, à l'exemple des réseaux de synonymes (deux mots sont reliés s'ils sont synonymes), des réseaux de co-occurrences [RS01] (deux mots sont reliés s'ils apparaissent dans une même phrase d'un ouvrage) ou encore des réseaux de dictionnaires [BS02] (deux mots sont liés si l'un est utilisé dans la définition de l'autre).

3.5 Classification des méthodes de découverte de communautés

La méthode proposée par Girvan et Newman [GN02] a marqué le début d'une nouvelle ère dans le domaine de la découverte de communautés dans les réseaux complexes. Depuis ce travail de référence, le sujet a reçu une extraordinaire attention de la part de la communauté scientifique et de très nombreuses nouvelles approches ont été sans cesse proposées. La détection de communautés s'approche des deux thématiques classiques en informatique qui sont le partitionnement de graphe et le clustering de données. Le problème de détection de communautés peut être vu comme un problème de clustering de données pour lequel il faut choisir une distance adéquate. Cependant, les graphes considérés par les applications de clustering usuelles ne possèdent pas les caractéristiques spécifiques des graphes complexes. Par conséquent de nombreuses approches classiques de clustering de données sont inadaptées pour la détection de communautés. Dans cette section, nous décrivons les approches de détection de communautés existantes. Notre but est de donner une classification des méthodes proposées, d'en illustrer la diversité, et de discuter leurs avantages et leurs inconvénients.

L'étude de découverte de communauté dans les réseaux complexes connaît une continuelle évolution et les auteurs proposent sans cesse des nouvelles approches pour la détection des communautés [FLM04, GSPA04b, LSH08, CNM04, BIL+07a, NL07, EM02a, SPGMA07, ZZZ08, FC08, DDGDA05]. Bien que la liste des méthodes présentées dans ce chapitre est importante elle n'est pas exhaustive. de ce fait, nous avons essayé de généraliser notre classification selon les différentes démarches utilisées. Notre classification permet d'expliciter les différents choix d'une approche possible pour la découverte de communautés et décider la quelle est la plus convenable pour un réseau étudié. Un choix convenable de la méthode de découverte à utilisée apporte un gain considérable en terme de temps d'exécution et de qualité de partition. Notre classification repose sur les trois points de vue suivants :

1- Selon la manière de regroupement des noeuds en groupes, Jain et Dubes [JD88] ont distingué deux approches pour ce faire : agglomérative et séparative. De même, nous avons remarqué que toutes les méthodes de détection de communauté utilisent soit une approche séparative soit une approche agglomérative pour regrouper les noeuds en communautés.

2- Dans le contexte d'évaluation des performances des méthodes de détection de communautés, nous avons constaté que les méthodes déterministes et stochastiques se diffèrent dans leurs apports en terme de complexité en temps et de qualité de partition. A cet effet, il est fort important de distinguer les méthodes déterministes de celles stochastiques.

3- Les approches de découverte de communautés caractérisent les communautés directement ou indirectement, par des propriétés globale du graphe, comme l'intermédiarité, la centralité,.. etc., ou par l'emploie de certains processus comme les promenades aléatoires, la synchronisation,..ect . Les communautés peuvent être également interprété en tant qu'une forme d'organisation topologique du graphe. Ainsi, les différentes démarches pour caractériser les communautés nous ont permis de classifier les méthodes existantes selon la technique utilisée au cours de la découverte de communautés.

3.6 Méthodes agglomératives

Dans les méthodes agglomératives les métriques de similarité entre les pairs de sommets sont calculées au moyen de plusieurs méthodes, et par conséquent les liens , qui relie les pairs de sommets de forte similarité, sont ajoutés progressivement au réseau initial. Ce processus d'ajout de liens peut être arrêté à n'importe quel niveau et les composants connectés obtenus représentent les communautés. La figure 3.4 illustre la classification des méthodes agglomératives. Cependant ces méthodes identifient les noyaux de communautés et n'incluent pas les noeuds périphériques. Les noeuds de noyau dans une communauté ont souvent une forte similarité, et par conséquent, ils sont reliés tôt dans le processus agglomératif, mais les noeuds périphériques sont négligés (ils ne sont pas mis dans la communauté appropriée)[NG04].

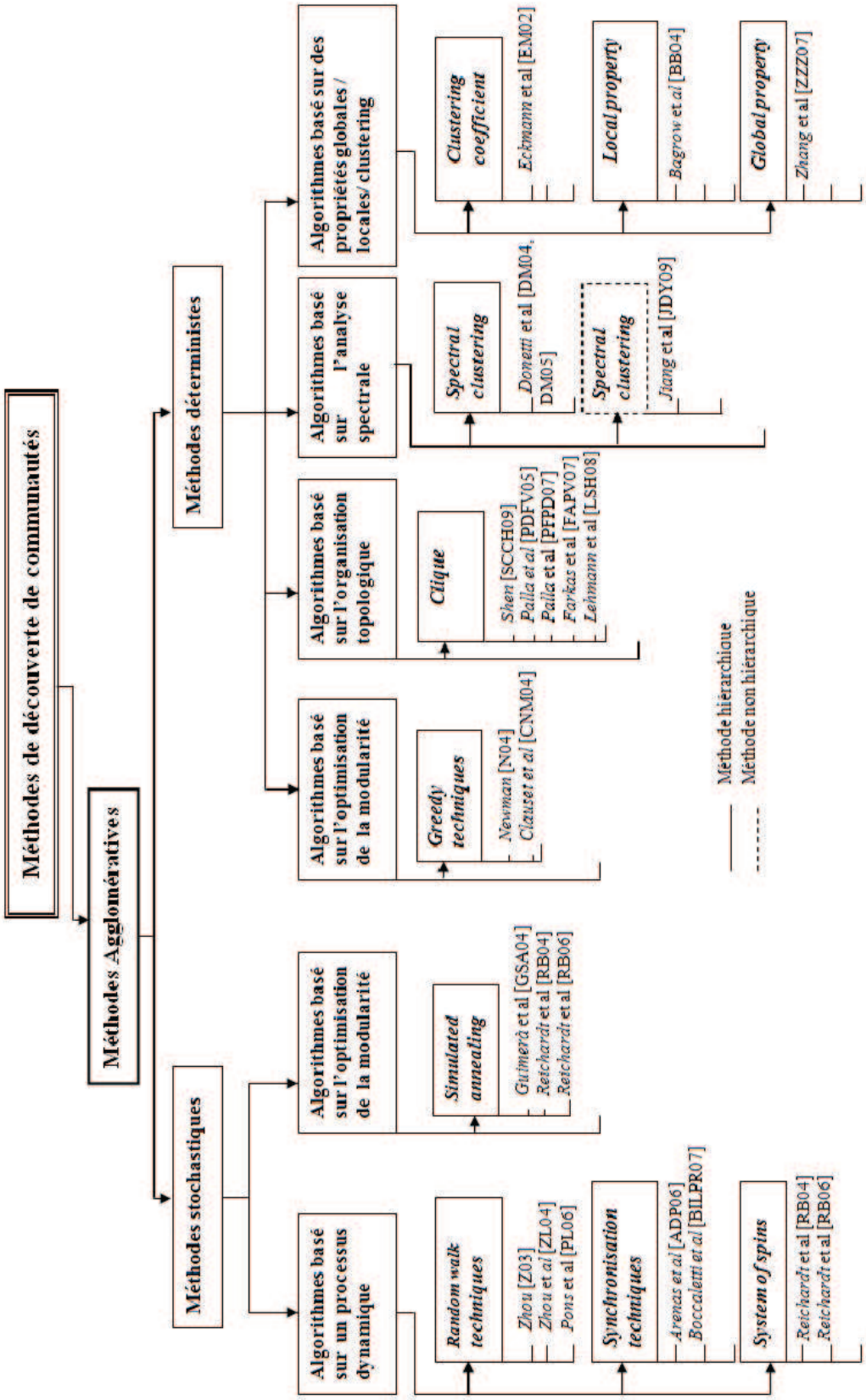


FIGURE 3.4 – Taxonomie des méthodes de découverte de communautés dans les réseaux complexes : 1. Méthodes agglomératives.

3.6.1 Méthodes basées sur l'optimisation de la modularité

Dans le travail [New03b], les auteurs ont montré que des valeurs élevées de modularité indiquent de bonnes partitions en communautés, ce qui a motivé la proposition de nombreuses méthodes pour la maximisation de la modularité.

3.6.1.1 Les méthodes gloutonnes

Newman [New04a] a proposé d'optimiser la valeur de modularité sur toutes les partitions possible afin de trouver la meilleur partition en communautés. Dans le Fast algorithm [New04a], la mesure utilisée pour découvrir les communautés n'est que le changement de Q quand on joint deux communautés. Cette méthode d'optimisation emploie un algorithme d'optimisation glouton. L'algorithme se déroule comme suit :

1. Initialement, on considère chaque communauté se compose d'un seul sommet.
2. Joindre la paire des communautés qui résulte d'une division qui maximise la valeur de modularité, mais ne pas joindre la paire des communautés entre lesquelles il n'existe pas de liens. (Utilisation d'un algorithme glouton)
3. La mis à jours des éléments de la matrice e_{ij} en ajoutant des lignes et des colonnes qui correspondent aux communautés jointes.
4. Répéter l'étape 2 jusqu'à ce que la modularité Q ne puisse pas être améliorée.

Afin d'étudier la performance des algorithmes de détection de communauté, plusieurs réseaux du monde réel ont été utilisés. Le travail [New04a] utilise un graphe de $n = 128$ sommets divisés en quatre communautés de 32 sommets chacune. Les liens ont été établit aléatoirement entre les paires de sommets : la probabilité qu'un lien relie les sommets dans la même communauté est P_{in}, la probabilité qu'un lien relie les sommets dans des communautés distinctes est P_{out}, et les valeurs de probabilité ont été choisit d'une façon que le degré prévu de chaque sommet est égale à 16. Le nombre moyen des liens (inter-communauté) qui relient un sommet aux sommets de n'importe quelle autre communauté est égale à z_{out}. Les résultats des sommets correctement identifiés selon la variation de z_{out} sont illustrés sur Fig. 3.5.
L'algorithme Fast identifie correctement plus que 90% des sommets pour des valeurs $z_{out} \leq 6$. Cependant, quand les liens intra-communauté et les liens inter-communauté par sommet deviennent égaux (z_{out} est proche de la valeur 8), nous constatons une dégradation des la performances de l'algorithme. L'algorithme GN [New04a] identifie correctement les sommets mieux que fast algorithm [New04a] pour des petites valeurs z_{out} (pour $z_{out} = 5$: GN détecte correctement 98.9% de sommets et l'algorithme fast détecte 97.4% de sommets). l' algorithm fast s'opère mieux que l'algorithme GN Pour des valeurs plus élevées de z_{out}.

Les décisions de l'algorithme fast se base sur les informations locales des différentes communautés, tandis que l'algorithme GN emploie les informations du réseau entier.

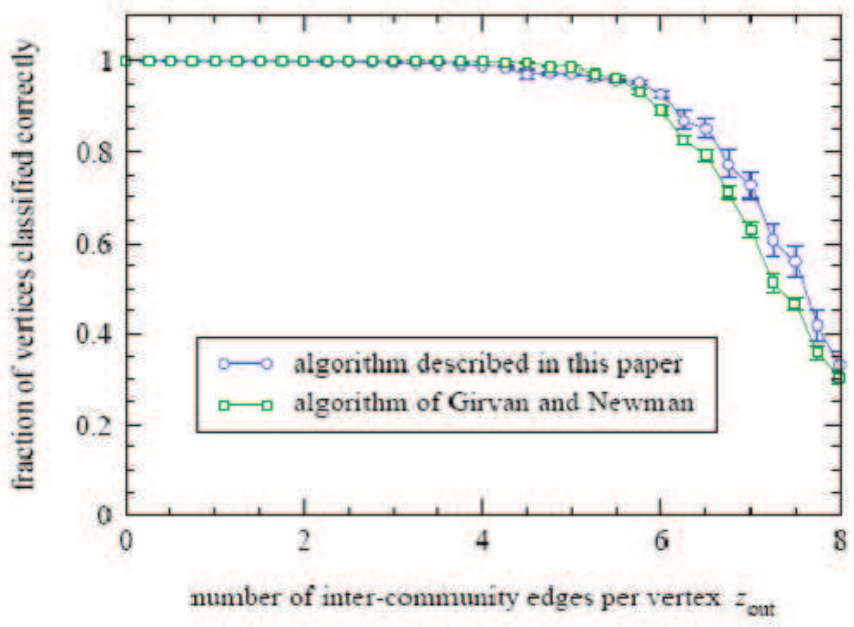


FIGURE 3.5 – Les résultats des noeuds correctement identifiés selon la variation de z_{out}.

3.6.1.2 Méthode de recuit simulé

Guimer et al. [GSPA04a] ont utilisé un algorithme de recuit simulé pour l'optimisation de la modularité. L'idée est d'effectuer un mouvement selon une distribution de probabilité qui dépend de la qualité des différents voisins ; les meilleurs voisins ont une probabilité plus élevée alors que les moins bons ont une probabilité plus faible. L'algorithme de recuit simulé converge rapidement vers la solution optimale mais il est plus efficace pour les petits réseaux.

3.6.2 Méthodes basées sur un processus dynamique

3.6.2.1 Les techniques de marche aléatoire

Les marches aléatoires dans les graphes sont des processus aléatoires dans lesquels un marcheur est positionné sur un sommet du graphe et peut à chaque étape se déplacer vers un des sommets voisins. Le comportement des marches aléatoires est étroitement lié à la structure du graphe. A cet effet, plusieurs approches de détection de communautés se basent sur ces comportements.

Pons et al [PL06] ont proposé l'algorithme "Walktrap" en utilisant le constat intuitif que les marches aléatoires vont se faire piéger dans des zones denses, en d'autres termes lorsqu'un marcheur sera dans une communauté il possédera une forte probabilité de rester dans la même communauté à l'étape suivante (grâce à la forte densité de liens internes et la faible densité de liens externes). Une marche aléatoire dans un graphe G est un processus en temps discret sur l'ensemble des sommets V. Sa matrice de transition P est donnée par :

$P_{ij} = \frac{A_{ij}}{d(i)}$.

P_{ij}^t : est la probabilité d'aller d'un sommet i à un sommet j par une marche aléatoire de longueur t. Le temps est discrétisé $(t = 0, 1, 2, ...)$ et un marcheur est localisé à chaque instant t sur un sommet du graphe G. Le marcheur se déplace à chaque instant aléatoirement et uniformément vers l'un de ses sommets voisins. La suite des sommets visités constitue une marche aléatoire. Ainsi un marcheur possède de grandes chances de rester lors d'une marche de courte distance dans sa communauté d'origine.
L'idée pour comparer la proximité de deux sommets est alors de comparer les distributions de probabilité des marches aléatoires partant de ces deux sommets. La propriété de réversibilité des marches aléatoires indique que les probabilités P_{ij}^t et P_{ji}^t sont directement reliées ; elles sont donc porteuses de la même information. Toute l'information des marches aléatoires concernant un sommet donné $i \in V$ est contenue dans les probabilités $\left(P_{kj}^t\right)_{k \in V}$. Ces probabilités correspondent à la i^{eme} ligne de la matrice P^t, et sont notées par un vecteur colonne $P_{i\bullet}^t$. Pons et al [PL06] ont définit une distance r_{ij} pour comparer les sommets du graphe comme suit :

$$r_{ij} = \sqrt{\sum_{k=1}^{n} \frac{(P_{ik}^t - P_{jk}^t)^2}{w(k)}} = \|D^{-\frac{1}{2}} P_{i\bullet}^t - D^{-\frac{1}{2}} P_{j\bullet}^t\| \tag{3.6}$$

Tel que D est la matrice diagonale des degrés, r est une distance Euclidienne $\in R^n$.

La distance r est directement liée aux propriétés spectrales de la matrice de transition P (deux sommets appartiennent à la même communauté ont des composantes similaires sur les vecteurs propres principaux).
Le problème de détection de communautés peut être réduit à détecter séparément des communautés dans chaque composante connexe. Cette approche permet de tirer profit de ces propriétés spectrales tout en gardant une complexité raisonnable $O(nmlog(n))$ grâce aux calculs de marches aléatoires. Lorsque la longueur des marches devient importante, la qualité des résultats diminue. Ceci est expliqué par le fait que les marches aléatoires atteignent rapidement leur état stationnaire limite.

Les méthodes proposées par Zhou [Zho03b, Zho03a] et Zhou et Lipowsky [ZL04] sont basées sur le nombre moyen d'étapes pour qu'une particule brownienne (mouvement aléatoire d'une particule) atteigne un sommet donné en partant d'un autre sommet.
La distance entre les sommets mesurée par une particule brownienne est utilisée pour identifier la structure de communauté et identifier le noeud central de chaque communauté. Soit un réseau connecté de N noeuds et M liens, A est sa matrice d'adjacence tel que :

$$A_{ij} = \begin{cases} 0 & S'il\ n'existe\ pas\ de\ lien\ entre\ i\ et\ j \\ A_{ij} = A_{ji} > 0 & Sinion,\ cette\ valeur\ designe\ la\ force\ d'interaction \end{cases} \tag{3.7}$$

L'ensemble des plus proches voisins du noeud i est dénotés par E_i, une particule brow-

nienne continue à se déplacer sur le réseau, et à chaque étape elle fait un pas à partir de sa position actuelle (i) vers la position du plus proche voisins (j). La matrice de transfert est donné par :

$$P_{ij} = \frac{A_{ij}}{\sum_{l=1}^{N} A_{il}} \tag{3.8}$$

La distance d_{ij} est le nombre moyen d'étapes pour qu'une particule brownienne se déplace du noeud i au noeud j est calculé comme suit :

$$d_{ij} = \sum_{l=1}^{N} \left(\frac{1}{I - B(j)} \right)_{il} \tag{3.9}$$

Tel que :

I **:** La matrice d'identité ;

$B(j)$ **:** est la matrice formée en remplaçant la j^eme colonne de la matrice P par une colonne de zéros.

Considérant un sommet i comme le sommet origine du réseau, l'ensemble
$\{d_{i1}, ..., d_{i,i-1}, d_{i,i+1}, ..., d_{iN}\}$
mesure à quelle distance tous les autres sommets sont situés de l'origine. Par conséquent, la perspective du réseau entier est identifiée à partir du sommet i. Supposons que les sommets i et j sont des voisins, la différence dans leurs perspectives du réseau peut être quantitativement mesuré. Zhou [Zho03a] a définit l'indexe de dissimilitude suivant :

$$\Lambda(i,j) = \frac{\sqrt{\sum_{k \neq i,j}^{N} [d_{ik} - d_{jk}]^2}}{(N-2)} \tag{3.10}$$

Quand les noeuds i et j appartiennent à la même communauté, la distance moyenne d_{ik} de i à n'importe quel autre sommet $k(k \neq i, j)$ soit presque similaire à la distance moyenne d_{jk} (de j à k). La valeur de dissimilitude $\Lambda(i,j)$ est petite si i et j appartiennent à la même communauté et grande s'ils appartiennent aux communautés différentes.

Dans [Zho03b], l'attracteur global d'un sommet i est le sommet le plus proche à i, tandis que l'attracteur local de i est son voisin le plus proche. Deux types de communautés ont été définit, selon les attracteurs locaux ou globaux. Une communauté L basée sur un attracteur local est identifiée selon les considérations suivantes :

1. Si le noeud $i \in L$ et j est un attracteur local du noeud i, alors $j \in L$.
2. Si $i \in L$ et i est un attracteur local d'un noeud k, alors mettre k dans L.
3. Un sous ensemble de L ne produit pas une communauté.

Zho et al ont définit les communautés basées sur un attracteur global, tel que chaque noeud a une forte probabilité d'être dans la même communauté C de son attracteur global.

Dans [Zho03a], les communautés sont identifiées en utilisant une procédure séparative dont les étapes sont les suivantes :

1. Initialement, le graphe entier est considéré comme une seule communauté. Un seuil maximal de dissimilitude θ_{upp} est attribué à cette communauté ;
2. Pour chaque communauté, un paramètre θ de seuil de résolution est introduit avec la valeur initiale θ_{upp} de cette communauté. Si $\Lambda(i,j) \leq \theta$, les sommets i et j sont marqués comme "amis".
3. Décrémenter la valeur de θ : tous les liens dans la communauté sont examinés pour voir si deux plus proches voisins sont des amis. Différent ensemble d'amis sont alors formées, chacun contient tous les amis des sommets dans l'ensemble. Un sommet qui n'a aucun ami rejoint l'ensemble des amis avec qui il a une forte interaction. Après cette opération, les sommets des communautés sont distribués en un certain nombre de communautés disjointes.
4. Un processus d'ajustement local est exécuté pour déplacer les noeuds qui n'ont pas été correctement classifié.
5. Si les sommets de la communauté n'ont pas été divisés, alors retourner à l'étape (3). Si les sommets sont divisé en deux ou plusieurs communautés, on assigne à la communauté père un seuil inférieur de dissimilitude θ_{low} équivalente à θ. A chaque nouvelle communauté est assignée une valeur θ_{upp} équivalente à la valeur courante de θ. Répéter l'algorithme à partir de l'étape (2) pour traiter les communautés identifiées.
6. Après que toutes les communautés soient traitées, le dendrogramme est dessiné pour démontrer le rapport entre les différentes communautés aussi bien que les seuils de dissimilitude supérieure et inférieure de chaque communauté.

Zhou et Lipowsky [ZL04] ont aussi utilisé le mouvement brownien pour définir l'algorithme Netwalk algorithm (NW) qui emploi la mesure de proximité structurelle (indexe de proximité) de deux sommets en appliquant une méthode de détection de communauté de clustering hiérarchique. Les algorithmes proposés par Zhou [Zho03b, Zho03a] et Zhou et Lipowsky [ZL04] peuvent identifier une structure de communauté significative. Cependant, ces algorithmes sont lents car le calcul des distances entre tous les pairs de sommets se fait en $O(n^3)$. Ce qui rend l'application de ces approches sur des grands graphes inadmissible.

3.6.2.2 Les systèmes à état de spins

Reichardt et al [RB04] ont combiné l'idée de Fu et Anderson [FA86] avec le modèle de clustering de Potts qui a été définit par Blatt et al [BWD96], ceci a permet de convertir les communautés du réseau vers le domaine magnétique. Dans [RB06], Reichardt et al ont proposé un framework pour détecter les communautés en déterminant l'état fondamental de "q-Potts model spin glass" [PMV87].
Reichardt et al [RB04] ont proposé un algorithme de découverte de communauté qui se base

sur le modèle de Potts à Q états. Le modèle de Potts est l'un des modèles les plus utilisés en physique statistique afin de décrire le comportement des corps magnétiques [KF69]. Il correspond à modéliser ces corps comme des spins à Q états situés aux noeuds d'un réseau et qui sont en interaction entre voisins de façon à s'aligner pour un corps ferromagnétique, ou bien à être en opposition pour un corps antiferromagnétique, selon le signe de la constante de couplage.

Fu et Anderson [FA86] ont démontré par analogie qu'il existe une relation entre l'énergie des systèmes physiques (représenté par l'Hamiltonien) et la fonction de coût dans un problème d'optimisation combinatoire. Soit le problème de partitionnement de graphes en deux sous graphes, le nombre de liens qui existent entre les deux sous graphes égale à :

$$\sum_{i>j} \frac{a_{ij}}{4}(\mu_i - \mu_j)^2 \tag{3.11}$$

tel que a_{ij} est le nombre de liens entre les deux sommets i et j, $\mu_i = \pm 1$ est une variable qui indique la partition à laquelle le sommet i appartient. La différence entre le nombre de sommets des deux sous graphes est égale à : $\sum_i \mu_i$.

Ainsi, la fonction de coût s'écrit comme suit :

$$C = \sum_{i>j}(\lambda - \frac{a_{ij}}{2})\mu_i\mu_j \tag{3.12}$$

La fonction de coût a la même forme que l'Hamiltonien d'un spin qui est donnée par :

$$H = \sum_{i>j}(J_0 - J_{ij})s_i s_j \tag{3.13}$$

Tel que un spin s_i à deux orientations "haut, bas" qui correspondent aux $\mu_i = 1$ et $\mu_i = -1$ respectivement. Pour les système de magnétisme aléatoire l'Hamiltonien est composé de deux termes : un composant ferromagnétique avec la constante de couplage J_{ij} et un composant antiferromagnétique avec la constante de couplage J_0. Les auteurs [RB04] ont modifié l'Hamiltonien de Potts à q états en ajoutant une contrainte globale :

$$H = -J \sum_{(i,j)\in E} \delta_{\sigma_i,\sigma_j} + \gamma \sum_{s=1}^{q} \frac{n_s(n_s-1)}{2} \tag{3.14}$$

Tel que :

E : est l'ensemble de liens,

σ_i : dénote les spins individuels $(i = 1, ..., N)$ qui peuvent prendre les valeurs $\sigma_{1,...,q}$.

n_s : dénote le nombre de spins correspondant au spin s, avec $\sum_{s=1}^{q} n_s = N$.

J : est la force d'interaction ferromagnétique,

γ : est un paramètre positif

δ : est le symbole de Kronecker.

Chaque sommet est caractérisé par un spin prenant q valeurs possibles. La première somme est le terme ferromagnétique de Potts qui représente une distribution homogène des spins dans le réseau, et est minimisé par : $H_{ferr} = -JM$. Le deuxième terme additionne tous les pairs de spins qui sont égaux, ce qui représente la diversité de la configuration de spins ou bien les classes de spins existantes.
Pour définir la structure de communauté ça revient à trouver l'état fondamental du système. Les communautés correspondent aux classes de sommets ayant des valeurs de spin égales. Le nombre q de spins possibles correspond au nombre maximal de communautés que l'on peut trouver et doit être choisi de manière à ce qu'il soit supérieur au nombre effectif de communautés. Ce travail emploie l'algorithme de "Monte Carlo single spin flip heat-bath algorithm" pour déterminer l'état fondamental du système (structure de communauté). L'optimisation de l'énergie du système (le deuxième terme de l'Hamiltonien) correspond à favoriser les liens intra-communauté et optimiser les liens inter-communauté.

Une comparaison a été effectué avec l'algorithme de Girvan et Newman (le graphe de $n = 128$ sommets, divisés en quatre communautés de 32 sommets chacune). Deux mesures ont été définit : Sensibilité et spécificité. Sensibilité : une paire de noeuds est positif (négatif) quand il est dans la même communauté (différente communauté), la spécificité désigne la fraction de tout les paires de noeuds positif (négatif) qui sont classifiés correctement par l'algorithme. Selon Fig. 3.6, les performances de l'algorithme sont aussi bien que la méthode de GN.

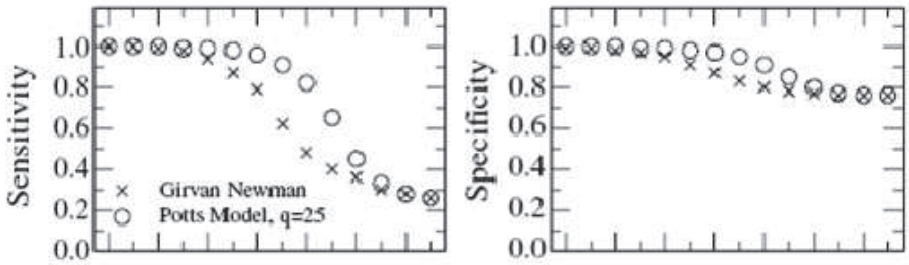


FIGURE 3.6 – Comparaison des résultats des sommets correctement identifiés par les algorithmes GN et l'algorithme de Reichardt et al [RB04] .

L'algorithme proposé par Reichardt et al [RB04] est capable de détecter l'affiliation des noeuds qui appartiennent aux plusieurs communautés, il s'adapte bien aux structures de communautés qui se chevauchent et permet la quantification de la stabilité des communautés. Cependant, le recuit simulé n'est pas une méthode d'optimisation globale efficace et l'algorithme ne pas être appliqué sur des grands réseaux.

3.6.2.3 Les techniques de synchronisation

Plusieurs études ont proposé un processus de synchronisation pour découvrir les communautés en prenant en considération la relation entre la structure topologique et les changements temporels [ADGPV06], [BIL+07b]. Il a été montré que les ensembles d'oscillateurs qui, sont fortement interconnectés et sont placés sur des nœuds de graphe, se synchronisent plus facilement pour former des clusters locaux [ADGPV06]. La complexité en temps de l'algorithme est $o(mn)$, ou n^2 sur des grands graphes [BIL+07b].

3.6.3 Méthodes basées sur l'analyse spectrale

Newman [New06] a réécrit la mesure de modularité Q sous forme matricielle. La méthode proposée vise l'optimisation de la modularité tout en choisissant une division appropriée du réseau.
Soit un réseau de n noeuds. Prenons $s_i = 1$ si le noeud i appartient au groupe 1, $s_i = -1$ si le noeud appartient au groupe 2.

La modularité s'écrit comme suit :

$$Q = \frac{1}{4m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) s_i s_j = \frac{1}{4m} s^T B s \tag{3.15}$$

Tel que :

A_{ij} **:** Matrice d'adjacence ;

k_i **:** Degré du noeud i ;

m **:** le nombre total des liens dans le réseau ($m = 1/2 \sum_i k_i$) ;

$\frac{k_i k_j}{2m}$ **:** le nombre prévu des liens entre les noeuds i et j s'il sont placés aléatoirement ;

$\frac{1}{4m}$: un facteur conventionnel : il est inclut pour la compatibilité avec la définition antérieure de modularité [NG04].

Une nouvelle matrice symétrique de modularité B a été définit comme suit :

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m} \tag{3.16}$$

Cette matrice est une matrice Laplacienne qui est la base de toutes les méthodes les plus connues de partitionnement de graphes.

La modularité peut être écrite en fonction des vecteurs propres u_i de B :

$$Q = \sum_i a_i u_i^T B \sum_j a_j u_j = \sum_{i=1}^{n} (u_i^T s)^2 \beta_i \tag{3.17}$$

Le choix des éléments du vecteur s qui optimisent la modularité revient à résoudre un problème NP-hard similaire aux problèmes de partitionnement spectrale.

1. Calculer le vecteur propre principal de la matrice de modularité (u_1).
2. Diviser les nœuds en deux groupes selon le signe des éléments correspondants dans ce vecteur. Les nœuds qui ont une valeur positive sont mis dans un groupe et les autres nœuds dans le deuxième groupe.

L'auteur [New06] a proposé une extension de sa première méthode pour diviser le réseau en plusieurs communautés. Ainsi, la matrice de modularité est décrite par l'équation suivante :

$$B_{ij}^{(g)} = A_{ij} - \frac{k_i k_j}{2m} - \delta_{ij}[k_i^{(g)} - k_i \frac{d_g}{2m}] \tag{3.18}$$

Tel que :

$B^{(g)}$ **:** Matrice de modularité d'un sous graphe g ;

d_g **:** La somme de degrés du sous graphe g.

L'algorithme se déroule comme suit :

1. Construire la matrice de modularité du graphe et trouver sa principale valeur propre et son principal vecteur propre.
2. Diviser le réseau en deux groupes selon les signes des éléments de vecteur propre.
3. Pour chaque groupe obtenu lors de l'étape 2 répéter le même algorithme de partitionnement.
4. Arrêter le processus de division si la modularité est nulle ou négative (le sous graphe est indivisible).
5. Quand tous les sous graphes sont indivisibles, le critère d' arrêter de l'algorithme est atteint.

Cet algorithme est testé sur le réseau des 105 livres de la politique américaine qui sont vendus sur le site Amazon.com. Sur la figure 3.7, les liens connectent des paires de livres qui sont fréquemment achetés par le même acheteur, les formes représentent l'alignement politique des livres : les cercles sont "liberal books", les carrés sont "conservative books" et les triangles sont "centrist books". Le résultat de l'algorithme est la détection de quatre communautés (marqué par des lignes pointillées).

On observe qu'une de ces communautés est composé entièrement des livres libérales et une autre se compose entièrement des livres conservateurs. Quand à la majorité des "centrist books" ont été identifiés dans les deux communautés restantes. En conséquence, ces livres forment des communautés d'achat qui sont alignées étroitement avec les points de vues politiques, ce qui démontre que l'algorithme proposé par Newman est capable d'extraire des résultats très significatifs. Les propriétés spectrales ont apporté plusieurs avantages à l'algorithme proposé. Cette méthode n'a pas seulement la capacité de diviser le réseau efficacement, mais également de refuser de le diviser quand aucune bonne division n'existe. Cependant, le coût de calcul des vecteurs propres est élevé.

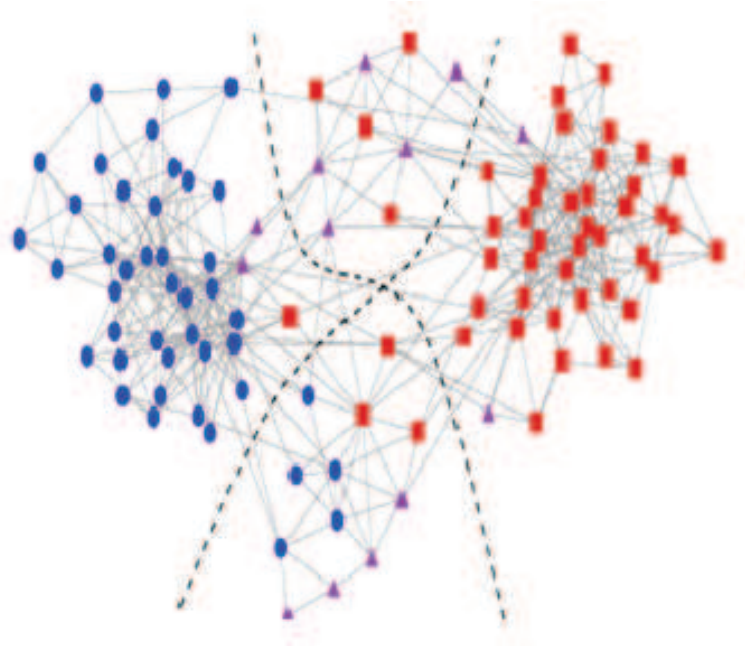


FIGURE 3.7 – "Krebs' network" réseau des livres d ela politique américaine.

3.6.4 Méthodes basées sur la structure topologique

Plusieurs approches de détection de communautés sont basées sur l'observation que la communauté peut être interprétée comme un sous-graphe complet (entièrement connecté). Dans cette section, nous décrivons les méthodes les plus référencées qui font partie des méthodes d'organisation topologique.

3.6.4.1 Cliques

Palla et al [PDFV05, DPV05] ont définit une nouvelle méthode de percolation de clique (CPM) pour détecter les communautés jointes des réseaux. CPM utilise l'information locale qui est la densité de liens. Les auteurs se sont basés sur l'observation qu'une communauté peut être interprétée comme union de plus petits sous graphes complets qui partagent des noeuds entre eux. De tels sous graphes complets dans un réseau s'appellent les $k-cliques$, où k est le nombre de noeuds dans le sous graphe. Deux $k-cliques$ seraient adjacentes si elles partagent $(k-1)$ noeuds, et une communauté est définie en tant que l'union de toutes les $k-cliques$ qui peuvent être atteintes par une série de $k-cliques$ adjacentes. Ces communautés peuvent être mieux visualisées à l'aide d'un modèle "template" $k-clique$ (un objet isomorphe pour un graphe complet de $k-sommets$). Cet objet peut être placé sur un $k-clique$ dans le graphe et roulé vers un $k-clique$ adjacent en changeant un de ses sommets et en gardant ses autres $(k-1)$ sommets. Ainsi, les communautés ($k-clique$ percolation cluster) sont tous ces sous-graphes qui peuvent être entièrement exploré en roulant l'objet $k-clique$ sur eux, comme il est illustré sur Fig. 3.8. Initialement le template est placé sur $A-B-C-D$, puis il est roulé sur le sous graphe $A-C-D-E$. A chaque étape, seulement

un des noeuds est déplacée et les deux $4-$*cliques* (avant et après le roulement) partage $k-1=3$ noeuds . À l'étape finale le template atteint le sous-graphe $C-D-E-F$, et l'ensemble de noeuds visités pendant le processus $A-B-C-D-E-F$ sont considérés comme la communauté identifiée par le CPM.

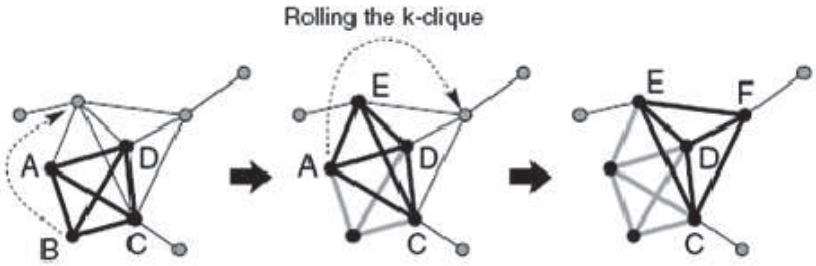


FIGURE 3.8 – Illustration de CPM [PDFV05, DPV05] un $k-$*clique* template roulé sur un petit graphe non orienté ($k=4$).

Une extension de l'algorithme CPM a été proposée pour les réseaux pondérés [FbPV07] et les réseaux orientés [PFP+07]. L'algorithme CPM est efficace pour la détection des communautés qui se chevauchent (un noeud peut être un membre de plusieurs différentes communautés en même temps). Pour détecter les communautés à partir de $k-$*clique*, il faut tout d'abord calculer les cliques maximales. La complexité de temps pour trouver les cliques est exponentielle et directement proportionnelle à la taille du graphe, néanmoins Palla et al ont prouvé que l'algorithme peut s'exécuter en un temps admissible et cela sur des réseaux du monde réels qui peuvent aller jusqu'au 10^5 sommets. Cependant, la méthode CPM suppose que le graphe a un grand nombre de cliques, ainsi elle peut échouer à détecter des partitions significatives pour des graphes contenant juste quelques cliques (faible densité), comme dans les réseaux technologiques.

Dans [SCCH09], Shen et al ont présenté un algorithme pour détecter à la fois la hiérarchie des communautés ainsi que leurs chevauchements en se basant sur l'ensemble de cliques maximales tout en employant un processus agglomératif. La similarité entre une paire de communauté C_1 et C_2 est calculé comme suit :

$$M=\frac{1}{2m}\sum_{v\in C_1,w\in C_2,v\neq w}\left[A_{vw}-\frac{k_v k_w}{2m}\right] \tag{3.19}$$

Tel que :

A_{vw} **:** La matrice d'adjacence ;

k_v **:** Le degré du sommet v ;

$m=$ Le nombre total de liens du réseau ($m=\frac{1}{2}\sum_{vw}A_{vw}$).

Une clique maximale est une clique qui n'est pas un sous-ensemble d'aucune autre clique. Les cliques maximales, dont les sommets font partie à d'autres plus larges cliques maximales,

s'appellent cliques maximales subordonnés. Les cliques maximales subordonnés peuvent dégradé le fonctionnement de l'algorithme et devraient être éliminées. La plupart des cliques maximales subordonnés ont de petites tailles. Ainsi, l'élimination de ces cliques se fait en respectant un seuil k et en négligeant toutes les cliques maximales dont la taille est inférieure à k. L'algorithme se déroule en deux étapes :

1. Trouver toutes les cliques maximales dans le réseau en utilisant l'algorithme de Bron-Kerbosch [BK73]. Ignorer les cliques maximales subordonnés et marquer les autres cliques en tant que communautés initiales. Chaque sommet subordonné est également considéré communauté initiale comportant un seul sommet. Calculer la similarité entre chaque paire de communautés.
2. Choisir la paire de communautés qui a une similarité maximale, les fusionner en une nouvelle et calculer la similarité entre la nouvelle communauté et les autres communautés.
3. Répéter l'étape 2 jusqu'à ce qu'il 'y a qu'une seule communauté qui correspond au graphe entier.

Shen et al ont définit une extension de modularité pour déterminer la qualité de partitionnement de l'algoritjme CPM :

$$EQ = \frac{1}{2m} \sum_{i} \sum_{v \in C_i, w \in Ci} \frac{1}{O_v O_w} \left[A_{vw} - \frac{k_v k_w}{2m} \right] \tag{3.20}$$

Tel que :

O_v **:** Le nombre de communautés auxquelles le sommet v appartient.

Dans la deuxième étape, l'algorithme détermine où couper le dendrogramme selon la valeur maximale de EQ. Les figures 3.9 3.10 comparent les résultats obtenues en appliquant fast algorithm de Nweman [New04a], $k-clique$ de Palla et al [PDFV05] et EAGLE algorithm de Shen [SCCH09] sur un réseau de collaboration scientifique.

L'algorithme AIGLE et l'algorithme de Fast résultent d'un nombre de communautés presque identique à chaque niveau hiérarchique sauf que la taille de ces communautés est un peu différente. L'algorithme AIGLE a détecté une communauté supplémentaire qui représente les liens et les noeuds appartiennent aux communautés jointes. De même, l'algorithme CPM détecte le chevauchement des communautés mais il ne découvert pas la hiérarchie complète des communautés. Cependant, le coût de calcul généré par l'algorithme AIGLE lors de la recherche de cliques maximales est très élevé.

3.6.4.2 Motifs

La forte densité des arêtes au sein d'une communauté détermine la forte corrélation entre les nœuds ce qui est désigné par la présence de motifs. Arenas et al [AFFG08] ont montré comment les motifs peuvent être utilisés pour définir des classes générales de nœuds, y compris les communautés, en réécrivant l'expression mathématique de la modularité de

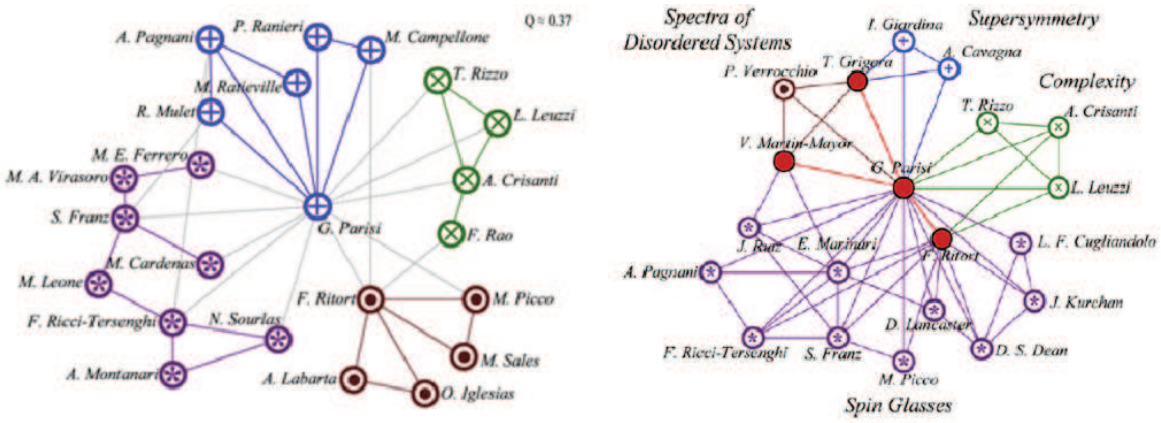


FIGURE 3.9 – Structure de communautés découvertes par fast algorithm (à droite) et l'algorithme k-clique. (Les noeuds et les liens qui appartiennent aux deux ou plusieurs communautés sont colorés en rouge)

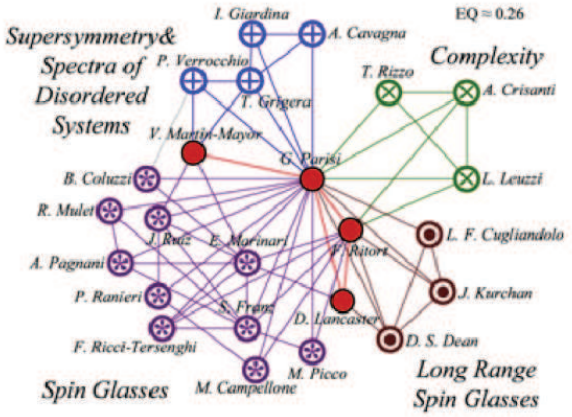


FIGURE 3.10 – Structure de communautés découvertes par EAGLE algorithm.

Newman-Girvan. Ils ont défini la modularité du motif comme la fraction de motifs à l'intérieur des communautés moins la fraction dans un réseau aléatoire.

3.6.5 Méthode basée sur des propriétés locales

Bagrow et al [BB05] ont proposé un algorithme de détection de communautés qui utilise l'information locale sans avoir une connaissance globale sur le réseau entier. Cette information locale représente le nombre de lien interne et externe d'un groupe de sommets se trouvant sur une distance géodésique depuis un sommet de départ. Afin de détecter les communautés localement et en se basant sur un simple critère qui emploie le nombre de liens interne et externe d'un groupe de sommets, Bagrow et al [BB05] ont utilisé la notion de *shell* et ont définit deux mesures de liens : Shell qui est définit en tant qu'un ensemble de sommets sur une distance géodésique depuis un sommet de départ. Le premier shell inclut les plus proches voisins du sommet de départ et le deuxième inclut les prochains voisins des plus

proches voisins de ce sommet et ainsi de suite (jusqu'au l shell). $K_i^e(j)$ est le degré émergent d'un sommet i qui représente le nombre de liens qui relient ce sommet i et les sommets d'un shell partant d'un sommet de départ j. K_j^l le degré émergent total d'un shell de profondeur l partant d'un sommet j. Il est clair que le nombre total des liens reliant les sommets d'un shell l est égale à la somme des degré émergent de tous les sommets ayant un lien vers le l shell :

$$K_j^l = \sum_{i \in S_j^l} k_i^e(j) \tag{3.21}$$

Avec S_j^l : est l'ensemble de tous les sommets à l pas du sommet j.

En outre, le changement de degré émergent totale d'un shell de profondeur l partant d'un sommet j, s'écrit comme suit :

$$\Delta K_j^l = \frac{K_j^l}{K_j^{l-1}} \tag{3.22}$$

L'algorithme étend le nombre de shell, à chaque itération, en ajoutant les sommets qui se trouvent à l pas du sommet j, tout en respectant un seuil de changement α tel que : $\Delta K_j^l < \alpha$. Pour un sommet de départ j faire :

1. Initialiser l shell, $l = 0$, depuis le sommet j, calculer K_j^0 (degré de sommet j) et ajouter j à la liste des membres de communauté.
2. Incrémenter le nombre de shell, $l = 1$, ajouter les sommets se trouvant sur le 1 shell à la liste des membres de communauté et calculer K_j^1.
3. Calculer ΔK_j^1 : Si $\Delta K_j^1 < \alpha$ une communauté a été détectée et le processus s'arrête, sinon, répéter l'algorithme à partir de l'étape 2 pour le prochain shell jusqu'à ce que la contrainte de seuil soit satisfaite ou bien le composant global connecté est ajouté à la liste des communautés.

Bagrow et al [BB05] ont définit une matrice d'adhésion M qui regroupe les vecteurs v_i représentant les communautés de chaque sommet de départ, puis selon la distance entre les vecteurs, un processus est exécuté afin de permuter les lignes qui ont une plus courte distance entre eux, ce qui permet de regrouper les sous communautés appartenant au même communauté. Ce regroupement permet de produire le dendrogramme correspondant à la structure de communautés.
Une illustration sur le réseau de club de karaté (voir Fig. 3.19) montre que l'algorithme atteint le résultat souhaité quand $\alpha = 1.2$, nous constatons que trois noeuds ne sont pas correctement détectés (3,14,20) comme le montre la matrice d'adhésion (Fig. 3.11.a). Ces noeuds se situent à la frontière des deux groupes existants et sont identiquement relié aux deux communautés, comme le montre le dendrogramme (Fig. 3.11.b).

En raison de sa nature locale, l'algorithme est très rapide et efficace dans certaines situations où il s'agit de l'identification d'une seule communauté mais sa version globale génère

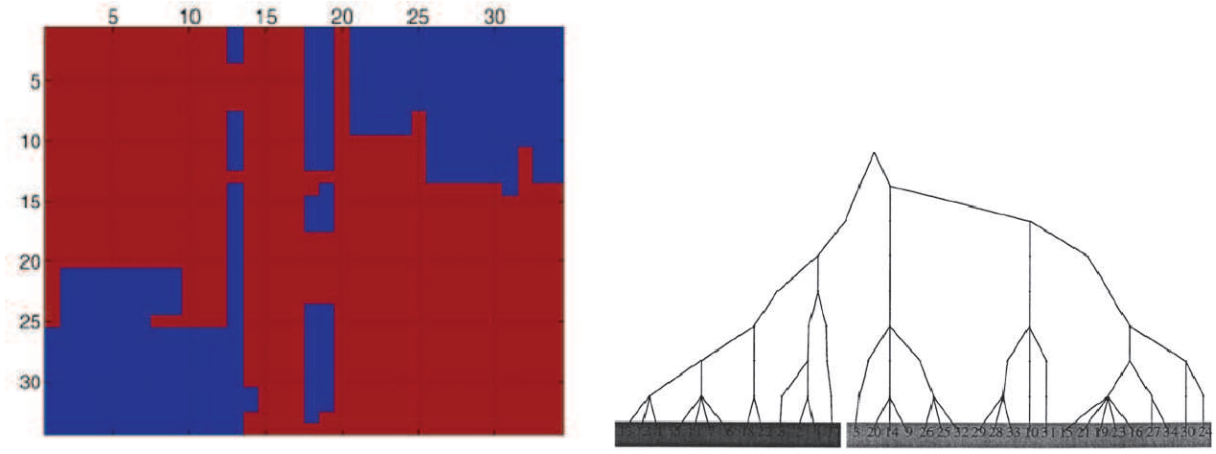


FIGURE 3.11 – (a) : Matrice d'adhésion du réseau de Zachary ($\alpha = 1.2$) ; (b) : Dendrogramme

un coût de calcul élevé $O(N^3)$ ($N \leq$nombre de sommets du réseau). Cependant, le shell peut être s'étaler sur un ou plusieurs autres communautés et cela dépend étroitement de la localisation de sommet de départ surtout si ce dernier est proche à un ou des sommets qui n'appartiennent pas à la communauté du sommet de départ.

3.6.6 Méthodes basées sur la propriété de clustering

Eckmann et al [EM02b] ont proposé une méthode basée sur les propriétés de clustering. L'idée est d'utiliser le coefficient de regroupement des nœuds comme une quantité pour distinguer les sous graphe de neouds connectés. Plus il y a de triangles dans le sous-graphe, plus la distance moyenne est courte.

3.7 Méthodes séparatives

Les méthodes séparatives [Zho03b, VD00, DA05, NL07, New06, LN08, ZWL$^+$08, RCC$^+$04, SPW$^+$08, GN02, NG04, FLM04, SPGMA07] scindent le graphe en plusieurs communautés en retirant progressivement les arêtes reliant deux communautés distinctes. La figure 3.12 illustre la classification des méthodes séparatives. Ces méthodes trouvent les pairs de nœuds qui sont reliés par des liens de faible similarité et les enlèvent au fur et à mesure. Ainsi, ce processus de suppression de liens peut être arrêté à n'importe quelle étape et le réseau est divisé en plusieurs composants représentants les communautés.

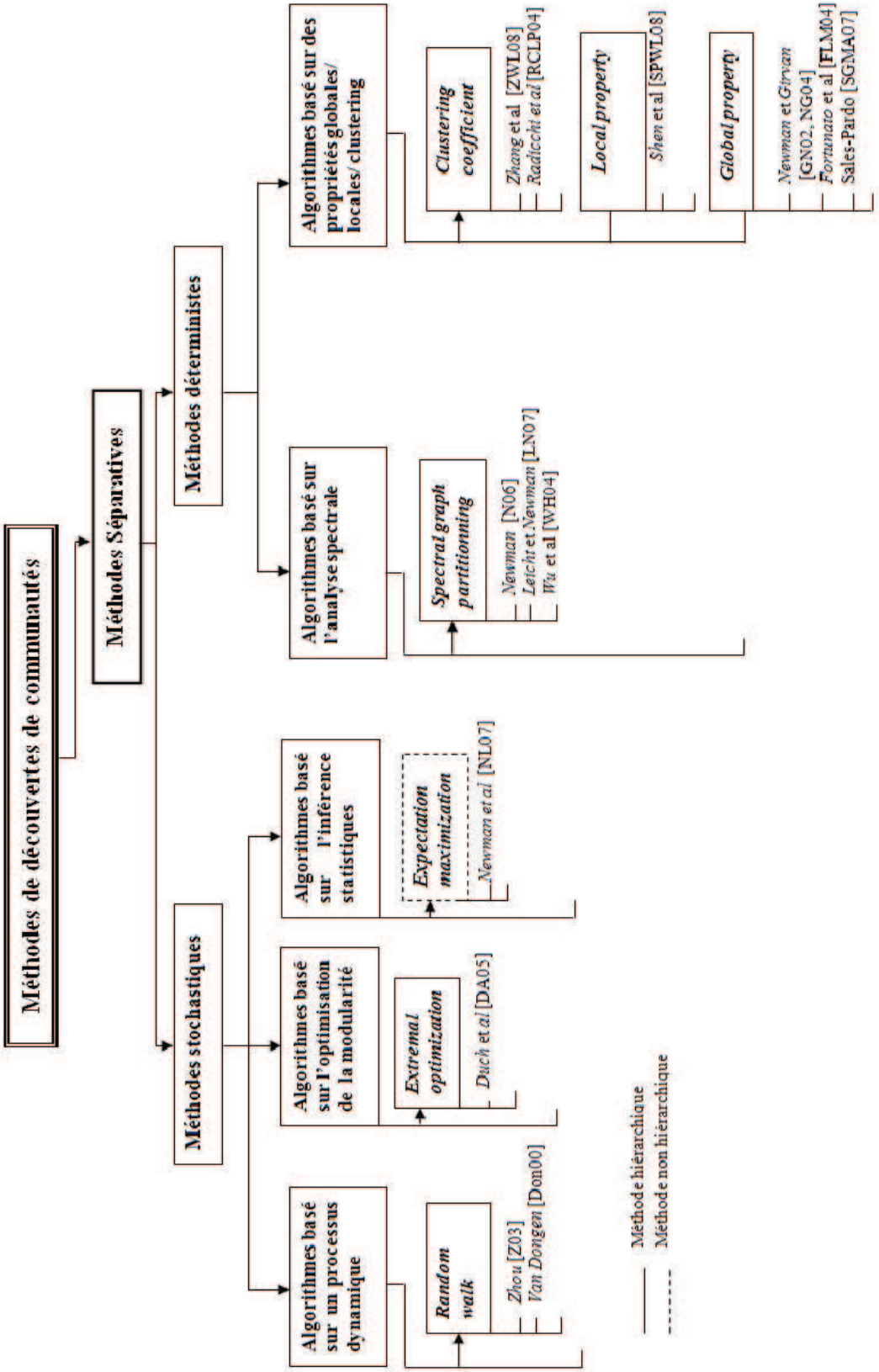


FIGURE 3.12 – Taxonomie des méthodes de découverte de communautés dans les réseaux complexes : 2. Méthode séparatives.

3.7.1 Méthodes de coefficient de clustering

Radicchi et al [RCC+04] ont proposé un algorithme séparatif de détection de communauté en introduisant un nouveau concept appelé coefficient de clustering d'arête. Ils ont définit le coefficient de clustering d'arête par analogie avec le coefficient de clustering de noeud.

Les approches de Radicchi et al [RCC+04] et d'Auber et al [ACJ+03] basées sur le clustering d'arêtes. Radicchi et al [RCC+04] proposent un coefficient de clustering (d'ordre g) d'arêtes. Il est défini comme étant le nombre de cycles de longueur g passant par l'arête divisé par le nombre total de tels cycles possibles. Cet algorithme retire donc à chaque étape l'arête de plus faible clustering.
Dans la méthode de Radicchi et al [RCC+04], chaque suppression d'arête ne demande qu'une mise à jour locale des coefficients de clustering, ce qui améliore la performance de l' algorithme. Cependant, cet algorithme se base sur la présence des triangles dans le réseau ; quand un réseau a peu de triangles, le coefficient de clustering d'arête est de petite valeur pour toutes les arêtes et l'algorithme est incapable de détecter les communautés.

Les réseaux bipartis sont un type important de réseau complexe. Un réseau biparti est composé de deux ensemble de sommets distincts. En fait, beaucoup de réseaux du monde réels sont naturellement bipartis, on cite souvent l'exemple du graphe reliant les acteurs aux films dans lesquels ils jouent, les graphes d'occurrence des mots dans les phrases, ou encore le graphe des auteurs de publications scientifiques. Watts et Strogatz ont introduit en 1998 la notion formelle de coefficient de clustering [WS98]. Il s'agit de la moyenne du ratio du nombre de voisins de u qui sont reliés entre eux sur le nombre total de liens qui pourraient potentiellement exister entre ces voisins. Zhang et al [ZWL+08] ont modifie la définition de coefficient de clustering. Ils l'ont adapté aux réseaux bipartis puis ont proposé un algorithme de détection de communauté pour les réseaux bipartis dont le principe est de retirer les liens qui ont la plus petite valeur de coefficient de lien.
Lind et al [LGH05] ont étudié le coefficient de clustering dans des réseaux bipartis où il n'y a pas de cycles de dimension trois , et par conséquent, la définition standard de coefficient de clustering donné dans [LP49] ne peut pas être utilisé. De ce fait, ils ont définit un coefficient donné par la fraction de cycles à quatre dimensions.

Le calcul des triangles possibles dans un réseau binaire prend en considération tous les liens éventuels entre les voisins les plus proches ; Donc $C_3(i)$ décrit la probabilité que les amis du noeud i soient des amis [WS98]. Le coefficient de clustering $C_4(i)$, qui a été définit par Lind et al [LGH05], est la fraction entre le nombre de carrés existants et le nombre total de tous les carrés possibles (C_4 représente la probabilité que vos amis ont des amis communs hormis vous).

Pour le noeud i (Fig. 3.13), le nombre de carrés (quadruplet de noeuds) possibles est donné par le nombre de voisins communs entre ses voisins m et n. Tandis que les carrés

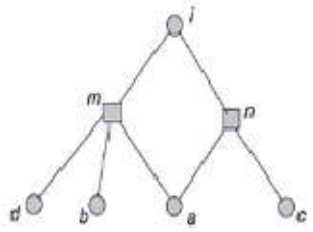


FIGURE 3.13 – Exemple pour illustrer le calcul des carrés.

sous-jacents peuvent être calculées en ajoutant des liens éventuels, par exemple b est l'ami de m mais pas de n ; on l'appelle : un ami commun sous jacent du noeud m et n. Si nous considérons le lien potentiel entre b et n, nous pouvons obtenir un carré sous-jacent $imbn$.

Selon cette considération, l'équation de C_4 est définit comme suit :

$$C_{4,mn} = \frac{q_{imn}}{(k_m - \eta_{imn}) + (k_n - \eta_{imn}) + q_{imn}} \tag{3.23}$$

Tel que :

m,n : Des voisins du noeud i ;

q_m, n : Le nombre de carrés qui incluent les trois noeuds (les carrés existants et sous-jacents).

$\eta_{imn} = 1 + q_{imn}$.

La définition de Lind considère d'éventuel coïncidence des noeuds, quand à la définition de Zhang et al [ZWL$^+$08] considère d'éventuels coïncidence des liens en se basant sur les normes du coefficient de clustering des réseaux binaires. Radicchi et al [RCC$^+$04] ont proposé un algorithme séparatif de détection de communauté en introduisant un nouveau concept appelé coefficient de clustering d'arête. Zhang et al [ZWL$^+$08] ont aussi définit le coefficient de clustering de lien LC_4 et LC_3 pour les réseaux bipartis.

Ainsi, LC_4 s'écrit comme suit :

$$LC_{4,iX} = \frac{q_{iX}}{(k_i - 1)(k_X - 1) + k_i^{(2)} + k_X^{(2)}} \tag{3.24}$$

Tel que :

q_{iX} : Le nombre de carré auxquels le lien l_{iX} appartient ;

k_i : Le degré du noeud i ;

$k_i^{(2)}$: Le nombre des voisins des voisins du noeud i sans les noeuds qui sont des premiers voisins du noeud X.

Dans les réseaux bipartis, les triples sont l'unité de base qui exprime la relation entre deux noeuds du même ensemble. Ainsi, les auteurs ont définit le coefficient de clustering de

lien LC_3 basé sur les triples. LC_3 d'un lien l_{iX} représente la moyenne de la similitude de liens obtenues de tout les triples à lesquels ce lien appartient. Il s'écrit comme suit :

$$LC_{3,iX} = \frac{1}{k_i + k_X - 2}\left(\sum_{m=2}^{k_X} \frac{t_{mi}}{k_m + k_i - t_{mi}} + \sum_{N=2}^{k_i} \frac{t_{NX}}{k_N + k_X - t_{NX}}\right) \tag{3.25}$$

Tel que :

m **et** i **:** sont du même ensemble ;

i **et** X **:** n'appartiennent pas au même ensemble ;

t_{mi} **:** le nombre de triples qui contient les noeuds i et m (de même pour les noeuds N et X).

Lors de la détection des communautés dans les réseaux bipartis, le lien avec la petite valeur LC_4 (ou LC_3) est retiré et cela à chaque étape. Les figures ci-dessous (Fig. 3.14) montrent un réseau biparti qui contient 6 noeuds en haut et 6 noeuds en bas.

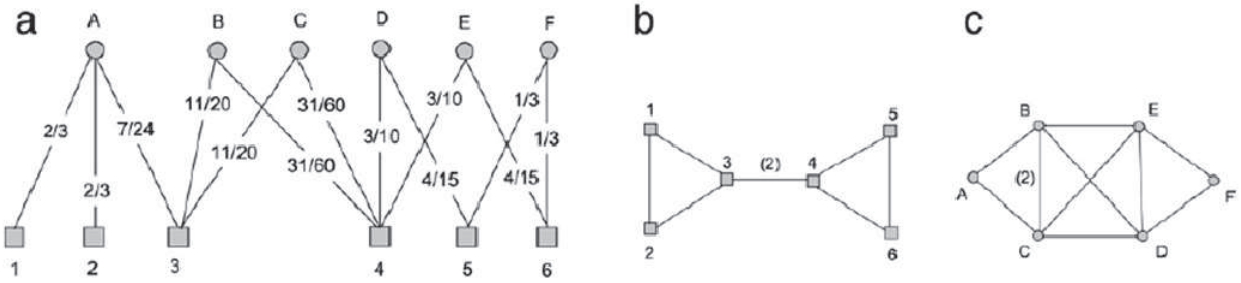


FIGURE 3.14 – Exemple d'un réseau biparti et sa projection.

L'exécution de l'algorithme basé sur le coefficient $LC3$ a résulté de :

a) Dans une première étape, les liens $D5$ et $E6$ sont supprimés et les communautés obtenues sont : $\{A, B, C, D, E, 1, 2, 3, 4\}$ et $\{F, 5, 6\}$,

b) Puis le lien $A3$ est retiré et les communautés obtenues sont : $\{A, 1, 2\}$, $\{B, C, D, E, 3, 4\}$, et $\{F, 5, 6\}$;

c) Dans la troisième étape, les liens $D4$ et $E4$ sont retirés. Il en résulte 4 communautés différentes : $\{A, 1, 2\}$, $\{B, C, 3, 4\}$, $\{D, E\}$, et $\{F, 5, 6\}$.

L'exécution de l'algorithme sur un réseau biparti d'Econophysists [LWFD07] qui se compose de 818 auteurs et de 777 papiers a permis de détecter 20 communautés. La modularité de cette partition est : $M_B(p)_1 = 0.351$ (en utilisant la fonction de modularité des réseaux bipartis qui a été définit dans [GSPA07]). L'algorithme résulte de communautés pertinentes. Cependant, le nombre de communauté à détecter doit être déterminé d'avance et il n'y a pas un critère d'arrêt bien précis pour arrêter la division du réseau en communautés.

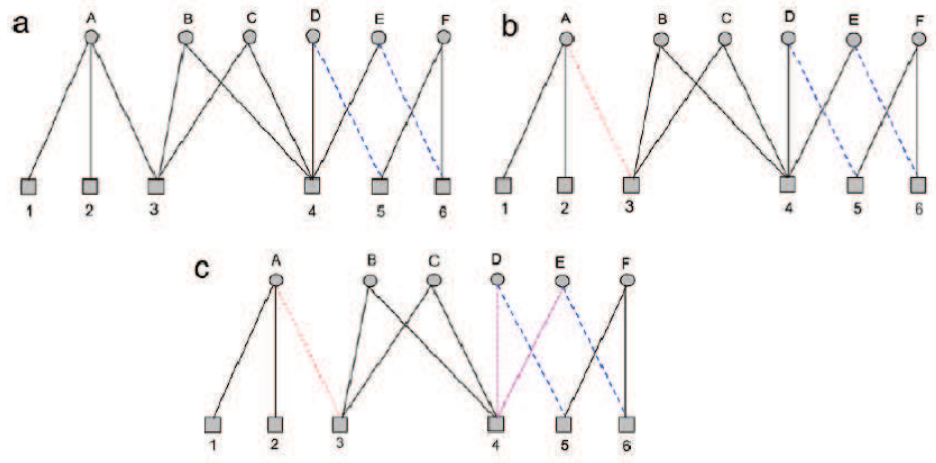


FIGURE 3.15 – L'identification des communautés par l'algorithme basé sur le coefficient de lien.

3.7.2 Méthodes basées sur des propriétés locales

Shen et al [SPW+08] ont proposé une méthode de détection de communautés en supprimant plusieurs liens simultanément dans chaque opération de filtrage, et ont définit un coefficient récursif de communauté pour quantifier la qualité de division au lieu d'utiliser la modularité.
Soit un réseau complexe de m liens et de n noeuds, P_{ij} représente la probabilité pour qu'il y ait des liens entre chaque paire de noeuds i et j.

$$P_{ij} = \frac{k_i k_j}{2m} \tag{3.26}$$

Tel que k_i et k_j sont respectivement les degrés des noeuds i et j.

Selon l'ordre décroissant de la valeur P_{ij}, les éléments de la matrice d'adjacence B_{ij} sont attribués (1 pour des valeurs supérieur, sinon 0).

Si $B_{ij} \neq B_{ji}$ alors : $B_{ij} = B_{ji} = 0$.

Certains liens entre les communautés peuvent être supprimées par l'opération de filtrage décrite par l'équation suivante :

$$C_{ij} = A_{ij} - B_{ij} \tag{3.27}$$

Si $C_{ij} = -1$ Alors $C_{ij} = 0$

Shen et al [SPW+08] ont proposé un coefficient récursif de communauté (CRC), dénoté M, afin de quantifier l'effet de division de réseau.

Soit un réseau, avec n noeuds et m liens, filtré par l'équation de C_{ij} et divisé en c sous réseaux. Soit n_k $(k = 1, ..., c)$ le nombre de noeuds dans le k^{ieme} sous réseau.

M s'écrit comme suit :

$$M = \frac{\frac{1}{2}\sum_{k=1}^{c}\sum_{ij}^{n_k} C_{ij}\delta(n_{w_i}, n_{w_j})}{\frac{1}{2}\sum_{ij}^{n} A_{ij}} \tag{3.28}$$

Tel que : $\delta(n_{w_i}, n_{w_j}) = 1$ si i et j sont dans le même sous graphe, sinon 0.
Répéter

1. Construire le modèle aléatoire du réseau.
2. Diviser le réseau par l'opération de filtrage donnée par l'équation de C_{ij}.
   Si le réseau est divisible aller à l'étape 3,
   sinon appliquer une nouvelle distribution du réseau avant le filtrage et aller à 1.
3. Calculer le coefficient CRC de chaque sous réseaux obtenu à l'étape 2,
   si le CRC est plus petit que celui de son réseau père alors : considérer le sous-réseau en tant que communauté locale et arrêter sa division.
   Sinon considérer chaque sous-réseau en tant qu'une nouvelle communauté et aller à 1.

Jusqu'à ce que toutes les communautés locales soient construites.

La méthode récursive de filtrage proposée par Shen et al [SPW+08] offre un gain en complexité de calcul $O(m^2 + (c+1)m)$, pour un réseau de m liens et c communautés. Cependant, cette méthode devient imprécise quand la densité des liens itra-communautés se rapproche à la densité des liens inter-communautés.

3.7.3 Méthodes basées sur des propriétés globales

Les algorithmes proposés par Newman et Girvan [GN02, NG04] se différencient des algorithmes séparatifs existants dans le fait qu'ils ne se basent pas sur l'enlèvement des liens de faible similarité entre les pairs de sommets, mais sur l'enlèvement des liens de forte similarité en introduisant la mesure appelé centralité "betweenness" qui se focalise sur les lien enter-communautés.
La mesure de betweenness, qui a été définit dans [NG04], consiste à trouver les plus courts chemins entre toutes paires de sommets et calculer l'implication de chaque lien le long de ces chemins. L'approche de Newman et al est inspirée du travail de Freeman [Fre77]. La conception intuitive d'un point central dans la communication, qui se base sur la propriété structurelle "betweenness", a été proposé par Freeman qui a définit ce point comme étant le point qui relie entre d'autres points tout au long de leurs plus courts chemins de communication [Fre77]. Le calcul de plus court chemin entre n'importe quelle pair de sommets peut être effectué en utilisant l'algorithme recherche en largeur d'abord "breadth-first search" (BFS) en temps $O(mn^2)$ [AMO94, CLRS01]. Newman a proposé [New01] un algorithme performant qui calcule le shortest path betweeness en $O(mn)$.

Une autre mesure qui se base sur le signal qui circule à travers le réseau a été définit. Si le signal passe de la source à la destination tout au long des géodésiques chemins et tous les sommets envoient des signaux à tous les autres noeuds par le même taux constant, alors betweenness est la mesure du taux des signaux qui traversent chaque lien. Supposons que le signale ne circule pas le long des plus courts chemins, mais il fait une promenade aléatoire dans le réseau jusqu'à ce qu'ils atteint sa destination. Cela permet de définir une mesure que Newman et al [NG04] l'ont appelé "random walk betweeness" (le nombre de périodes durant lesquelles le signal traverse un lien dans une seule direction). Un random walk betweeness d'un lien (v, w) est définit comme suit : $|V_v - V_w|$

Tel que V est donné par la formule suivante :

$$V = D_t^{-1}(I - M_t)^{-1}s = (D_t - A_t)^{-1}s \tag{3.29}$$

Tel que :

t : Sommet destination ;

s : Le vecteur de la source s ;

A_t : La matrice d'adjacence sans la t^{ieme} ligne et colonne
Enlever le sommet t du graphe car on s'intéresse aux nombre de pas nécessaire pour atteindre t ;

D_t : La matrice diagonale sans la t^{ieme} ligne et colonne ;

k_i : : Le degré du noeud i,
La probabilité de transition de j à i est : A_{ij}/k_j ,$M = A.D^{-1}$

$k_v^{-1}[(I - M_t)^{-1}]$: est le nombre moyen des périodes d'un pas de n'importe quelle longueur qui traverse le lien de v à w.

L'algorithme de détection de communautés proposé par Newman et al [NG04] se déroule comme suit :

1. Calculer les scores de centralité d'intermédiarité "betweeness" pour tous les liens du réseau.
2. Trouver le lien de plus fort score et le retirer du réseau.
3. Recalculer le score betweeness entre tous les liens restants.
4. Répéter l'algorithme à partir de l'étape (2) jusqu'à ce qu'il n'y ait plus de liens à retirer.

Nous citons quelques exemples de l'exécution de l'algorithme de Newman et Girvan sur des réseaux du monde réel :

a) Prenons le réseau de club de karaté de Zachary [Zac77] (Fig. 3.16). L'algorithme résulte de deux communautés qui correspondent parfaitement aux deux groupes du réseau réel. Le dendrogramme correspondant à cette division est donné sur la figure 3.17. La modularité résultante de la division des deux version de l'algorithme (shortest path betweeness et random walk betweeness) est élevés et le réseau est divisé en deux communautés (environ 0.4).

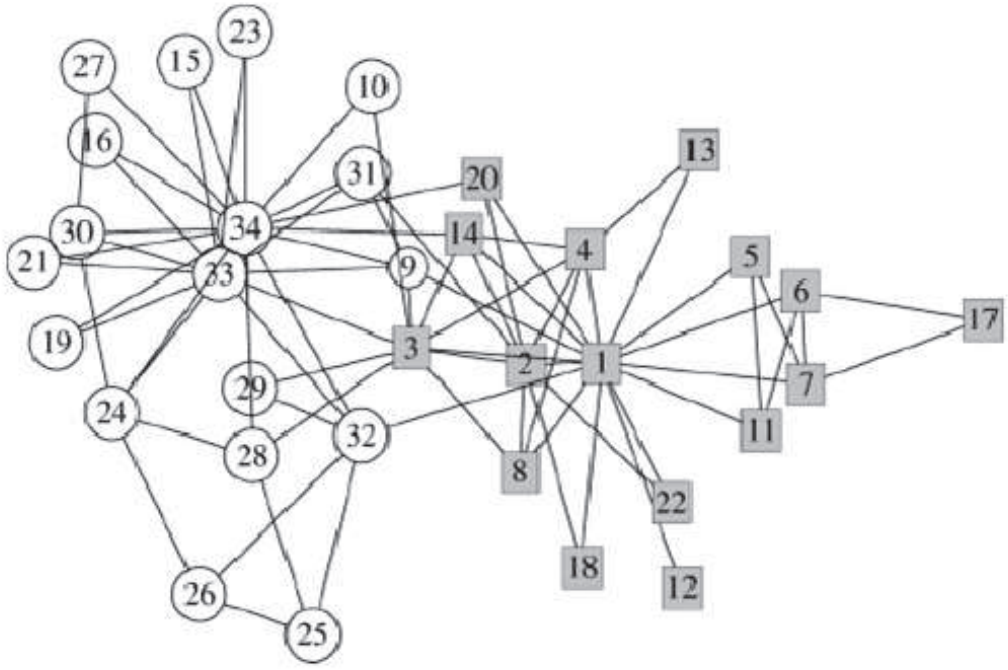


FIGURE 3.16 – Le réseau d'amitiés entre les individus dans l'étude de club de karaté de Zachary. Les carrés représentent les individus qui appartiennent au club administrateur et les cercles représentent les individus appartenant au groupe instructeur.

Cependant, le sommet 3 a été mal classé par la version shortest path betweeness. Quand à l'algorithme random walk betweeness produit une classification correcte de tous les noeuds.

b) Prenons le réseau des personnages du roman Les Misérables de Victor Hugo. L'apparence des personnages ensemble dans une ou plusieurs scènes a été étudiée. La modularité la plus élevée généré par la version de shortest path betweeness algorithm est $Q = 0.54$ et correspond aux 11 communautés. Les communautés reflètent clairement la structure du livre (Jean Valjean et Javert forment le centre des communautés avec leurs adhérents respectifs).

L'algorithme présente de meilleure qualité de partition, notamment dans les réseaux de taille moyenne. Cependant, les deux version de l'algorithme shortest path betweeness et random walk betweeness sont très coûteux en calculs et s'exécute en $O(n^3)$ à cause de nombre de calculs répétés à chaque suppression d'un lien ce qui est inadmissible dans des applications critiques.

3.7.4 Méthode basées sur l'optimisation de la modularité

Duch et al [DA05] ont proposé une procédure de recherche heuristique pour optimiser la valeur optimale de la modularité. Ils considèrent que la modularité globale Q est la somme de la modularité local sur chaque sommet. La variable global à optimiser est la modularité $Q = \sum_r (e_{rr} - a_r^2)$. La définition de la variable locale dans le problème d'optimisation extrémal

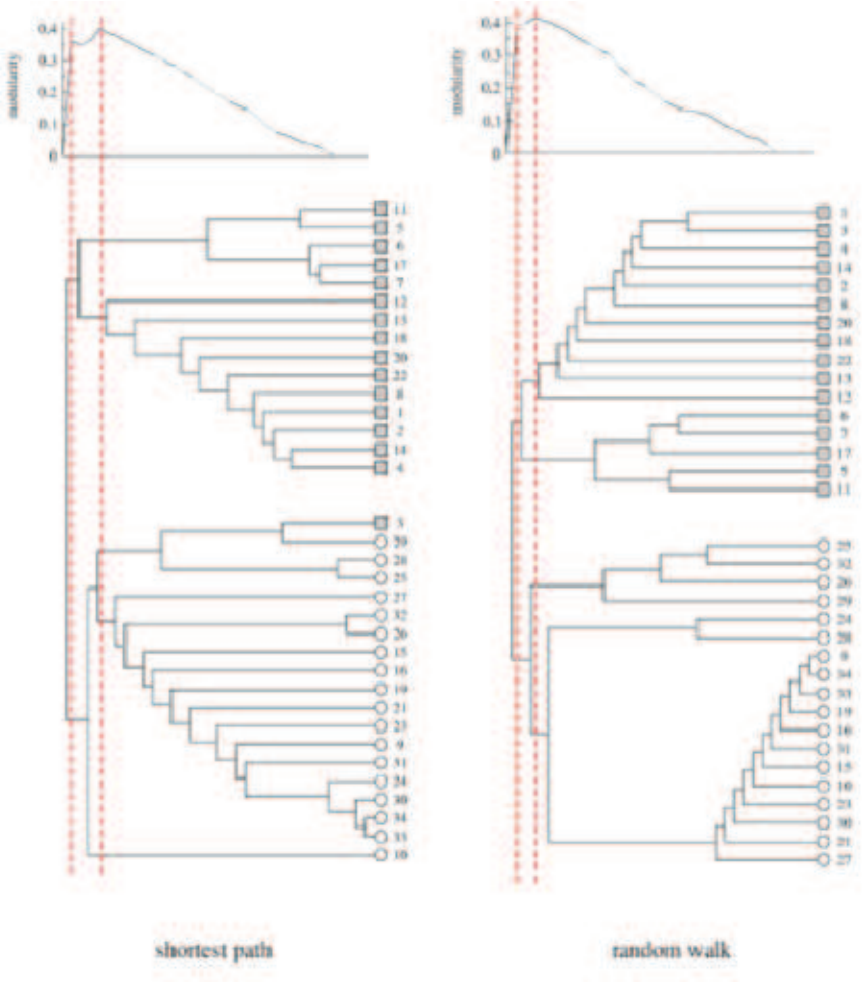


FIGURE 3.17 – Les dendrogrammes qui représentent les communautés généré par l'exécution des algorithme shortest path betweeness et random walk betweeness (réseau de club de karaté de Zachary).

doit être liée à la contribution de différents noeuds i dans la modularité, elle est normalisée dans l'intervalle $[-1, 1]$ et donnée par :

$$\lambda_i = \frac{q_i}{k_i} = \frac{k_r(i)}{k_i} - a_r(i) \tag{3.30}$$

Tel que :

λ_i : La division de la modularité locale de chaque noeud sur son degré ;

k_i : Le degré du noeud i.

$k_r(i)$: Le nombre de liens entre le sommet i et des sommets qui appartiennent à la même communauté r.

La procédure de recherche heuristique proposée pour trouver la valeur optimale de la modularité se déroule comme suit :

FIGURE 3.18 – Les communautés des personnages du roman Les Misérables de Victor Hugo.

1. Initialement, les noeuds des graphes sont divisés en deux partitions aléatoires contenant le même nombre de noeuds ;
2. À chaque itération, le système s'auto-organise en déplaçant le noeud de plus faible valeur de fitness à une autre partition. Ces déplacement modifient les partitions, donc la valeur de fitness doit être recalculée ;
3. Les liens entre les deux partitions sont supprimés et on répète l'étape 2 et 3 pour chaque nouveau composant ;
4. Le processus est répété jusqu'à ce que la modularité Q ne puisse pas être amélioré (l'état optimal est atteint).

L'exécution de l'algorithme d'optimisation extrémale sur le réseau de Zachary produit une valeur de modularité optimale, après trois itérations, et divise le réseau en 4 communautés (Fig. 3.19 et Fig. 3.20). La valeur de la modularité est 0.419 supérieur à la valeur 0.318 produite par l'algorithme de Newman [New04a]. Aussi, cette valeur est supérieur à 0.406 donné par l'algorithme de Reichardt et al [RB04] et supérieur aux résultats de l'algorithme de Donetti et al [DM04] (on a 0.412 sur le réseau de Zachary).

Bien que l'algorithme de Duch et al [DA05] produit une modularité optimale qui dépasse plusieurs algorithmes existants tout en optimisant la complexité de calcul en $O(n^2 log(n))$, il dépend étroitement de l'étape d'initialisation du réseau en partition aléatoire. De ce fait, l'optimisation de la modularité ne conduit pas forcément à la partition souhaitée.

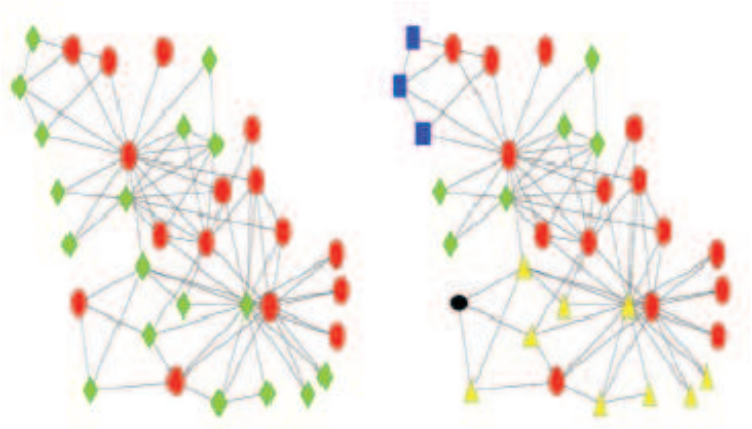


FIGURE 3.19 – Division initiale aléatoire du réseau de Zachary (le nombre de composants initial connecté dans les deux partitions est 5).

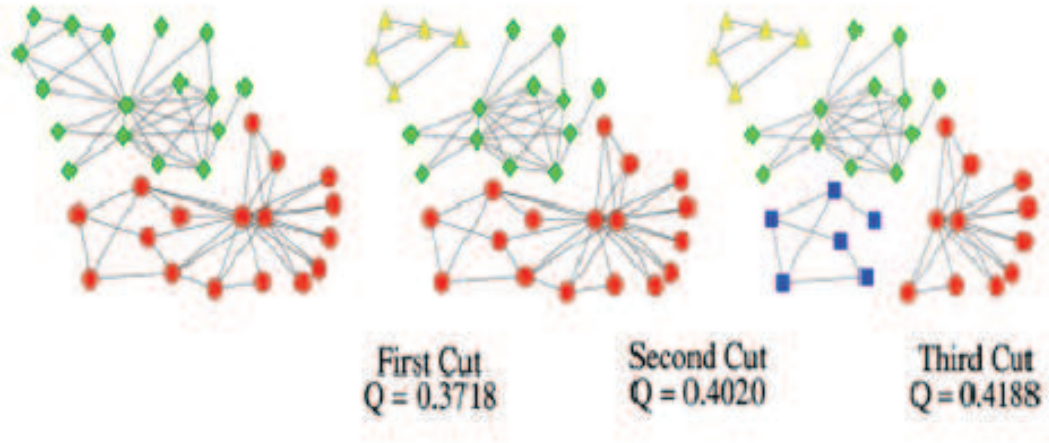


FIGURE 3.20 – Communautés obtenues après l'exécution de l'algorithme d'optimisation extrémale.

3.7.5 Méthodes basées sur l'analyse spectrale

Plusieurs auteurs ont étudié les réseaux orientés [NL07, GSPA07, ADFG07, RB08]. Dans le travail proposé par Leicht et Newman [LN08] la fonction de modularité a été généralisé afin d'incorporer l'information utile contenue dans l'orientation de lien. Vue que plusieurs réseaux complexes sont orientés, tels que le World Wide Web, food webs, beaucoup de réseaux biologiques et les réseaux sociaux, Leicht et Newman [LN08] ont proposé une extension de la méthode d'optimisation spectrale de la modularité pour les réseaux complexes orientés.

Dans [New06], Newman a écrit la fonction de modularité sous sa forme matricielle comme suit :

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta_{c_i, c_j} \tag{3.31}$$

Leicht et Newman [LN08] ont réécrit la fonction de modularité des réseaux orientés. Considérons deux sommets i et j. Supposons que le sommet i a un degré sortant élevé et un faible degré entrant, et inversement pour le noeud j. Donc, la probabilité qu'un lien partant d'un sommet i est orienté vers j est : $\frac{k_i^{in} k_j^{out}}{m}$. Ainsi, la modularité est définit comme suit :

$$Q = \frac{1}{m} \sum_{ij} \left[A_{ij} - \frac{k_i^{in} k_j^{out}}{m} \right] \delta_{c_i, c_j} \tag{3.32}$$

La matrice de modularité est donnée par :

$$B_{ij} = A_{ij} - \frac{k_i^{in} k_j^{out}}{m} \tag{3.33}$$

Pour obtenir la matrice symétrique, Q est donné par :

$$Q = \frac{1}{4m} s^T (B + B^T) s = \beta_i (v_i^T s)^2 \tag{3.34}$$

Tel que β_i est la valeur propre de $(B + B^T)$ qui correspond au vecteur propre v_i.

Pour diviser le réseau en deux communautés, on calcule le vecteur propre correspondant à la plus grande valeur propre positive de la matrice symétrique $(B + B^T)$ et on assigne alors les communautés en se basant sur les signes des éléments du vecteur propre.

Pour diviser le réseau en plusieurs communautés, une généralisation de la matrice de modularité a été donnée par la formulation suivante :

$$B_{ij}^{(g)} = B_{ij} - \delta_{ij} \sum_{k \in g} B_{ik} \tag{3.35}$$

Tel que $B^{(g)}$ est la matrice de modularité du sous graphe g. L'algorithme se déroule comme suit :

1. Construire la matrice de modularité du graphe $(B + B^T)$ et trouver la plus grande valeur propre positive et son principal vecteur propre.
2. Diviser le réseau en deux groupes selon les signes des éléments de ce vecteur.
3. Un processus d'ajustement local est exécuté pour déplacer les noeuds qui n'ont pas été correctement classifié.
4. Pour chaque communauté obtenue lors de l'étape 2 répéter le même algorithme de division en utilisant la matrice de modularité généralisée.
5. Si l'algorithme ne trouve aucune division qui peut maximiser la modularité d'une communauté donnée, donc la communauté ne peut pas être divisée en des sous communautés. Quand toutes les communautés atteindraient cet état l'algorithme s'arrête.

Leicht et Newman [LN08] ont utilisé plusieurs exemples des réseaux pour démontrer l'efficacité de leur algorithme. Un exemple du réseau qui représente la relation entre un ensemble de termes techniques, telles que "vertex", et "edge" et "community", contenu dans

un glossaire dérivé des papiers publiés récemment par Newman [New03b] et Boccaletti et al [BLM+06].

Les sommets dans ce réseau représentent les termes techniques et il y a un lien orienté d'un sommet vers l'autre si le premier terme a été employé dans la définition du deuxième terme. Fig. 3.21 illustre le résultat de l'exécution de l'algorithme de modularité orienté, il en résulte 6 communautés. Chaque communauté regroupe les termes commun à un concept donné (on trouve par exemple une communauté qui représente les termes de réseau orienté). L'algorithme a pu trouver une structure de communauté significative permettre de bien comprendre le contexte étudié dans les papiers [New03b] et [BLM+06].

L'algorithme de modularité dans sa version non orientée a été également appliqué sur ce même réseau, ce qui résulte de quatre groupes. Deux de ces derniers sont étroitement semblables à ceux trouvés par l'algorithme orienté. Cependant, les autres groupes contiennent un mélange de termes qui ne correspondent pas strictement aux mêmes concepts de réseau, avec des mots tel que "vertex," "diameter," "cycle," et "motif" ont été regroupé ensemble. Les méthodes de découverte de communautés des réseaux non orientés sont le plus souvent incapables de détecter une partie très significative de la structure de communautés puisqu'elles ignorent l'information contenue dans l'orientation de lien. La méthode d'optimisation spectrale de la modularité dans sa version destinée aux réseaux complexes orientés extrait l'information d'orientation de liens pour identifier la structure de communautés, ce qui donne une structure de communautés significative. Aussi, son coût de calcul qui est $O(n^2 log(n))$ rend son utilisation très bénéfique.

Les méthodes spectrales consistent à plonger le graphe dans un espace euclidien de sorte que les sommets fortement reliés soient représentés dans une même partie de l'espace et les sommets sans ou avec peu de connexions soient représentés à distance. Donetti et al [DM04] ont proposé une approche basée sur les propriétés spectrales de la matrice Laplacienne du graphe. Les coordonnées i et j des vecteurs propres correspondant aux plus petites valeurs propres non nulles sont corrélées lorsque les sommets i et j sont dans la même communauté. Une distance (distance euclidienne ou distance angulaire) entre sommets est alors calculée à partir de ces vecteurs propres, cette distance étant ensuite utilisée dans un algorithme de clustering hiérarchique. Le nombre de vecteurs propres à considérer est à priori inconnu. Plusieurs calculs sont successivement effectués en prenant en compte différents nombres de vecteurs propres, et le meilleur résultat est retenu. Les performances de l'algorithme sont limitées par les calculs des vecteurs propres qui se fait en $O(n^3)$ pour une matrice creuse. Une amélioration de cette approche a été proposée en utilisant une version normalisée de la matrice Laplacienne [DM05].

Dans [JDY09], les auteurs ont reformulé la mesure de modularité "Q" en utilisant le clustering spectral afin de maximiser la modularité et en conséquence détecter correctement la structure de communauté du réseau. Les méthodes spectrales ont montré expérimentalement de meilleurs résultats mais elles sont coûteuses en terme de complexité car la détermination des valeurs et vecteurs propres d'une matrice creuse nécessite un temps de calcul en $O(n^3)$.

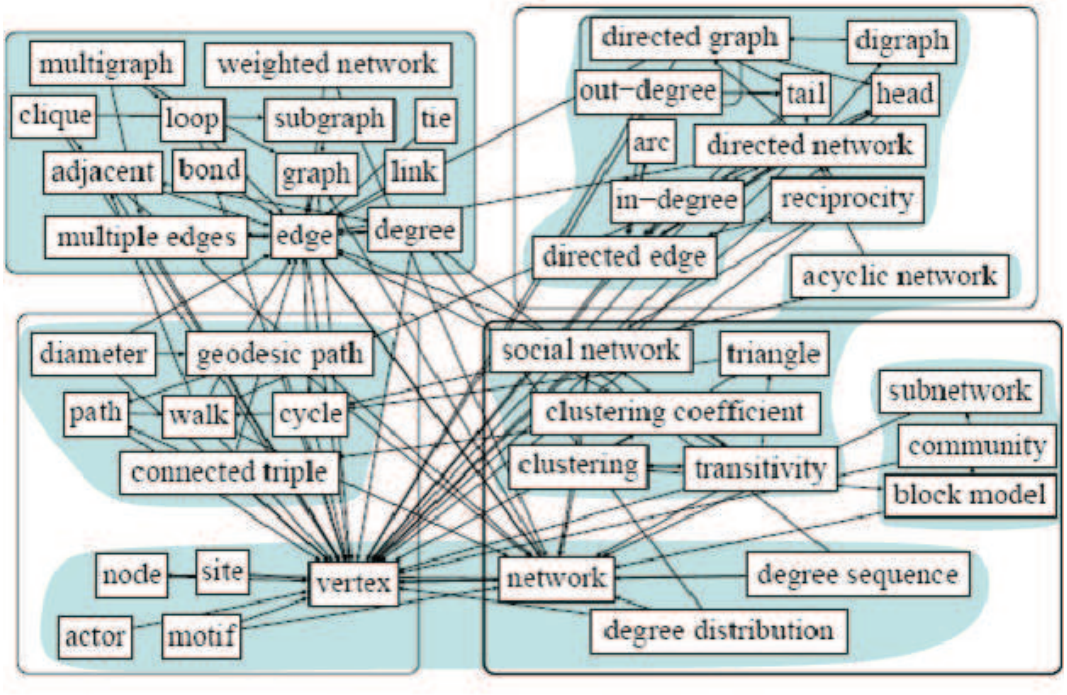


FIGURE 3.21 – Réseau des termes techniques illustre la découverte de communautés en appliquant l'algorithme de modularité orienté (groupes en bleu) et l'algorithme d'optimisation de modularité (groupes encadrés).

3.8 Étude comparative des algorithmes de découverte de communautés

La disponibilité d'une partition de référence a permis de proposer différentes mesures pour évaluer la qualité d'une partition identifiée par un algorithme de découverte de communautés. Le travail [GHL06] présente une revue des différents critères pour comparer les partitions détectées.
Nous citons un exemple des mesures qui sont basées sur l'information mutuelle. Une version normalisée de l'information mutuelle (NMI) est introduite dans [DDGDA05] est donnée par :

$$N\ M\ I(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} N_{ij} log(\frac{N_{ij}N}{N_{i\bullet}N_{\bullet j}})}{\sum_{i=1}^{C_A} N_{i\bullet} log(\frac{N_{i\bullet}}{N}) + \sum_{j=1}^{C_B} N_{\bullet j} log(\frac{N_{\bullet j}}{N})} \tag{3.36}$$

Tel que :

N : La matrice de confusion. Les éléments N_{ij} représentent le nombre de sommets dans la communauté de référence i qui sont également dans la communauté détectée j.

C_A : Le nombre de communautés dans la partition de référence A

C_B : Le nombre de communautés dans la partition de référence B

Si les communautés identifiées sont identiques aux communautés de référence, alors $NMI(A, B)$ prend sa valeur maximale 1.

Une autre mesure utile de similarité entre les partitions est l'indice de Jaccard qui s'écrit comme suit :

$$I_j(A, B) = \frac{n_{11}}{n_{11} + n_{01} + n_{10}} \tag{3.37}$$

Tel que :

n_{11} : Le nombre de paires de sommets qui sont mis dans la même communauté dans les deux partitions A et B ;

n_{10} : Le nombre de paires de sommets qui sont mis dans la même communauté dans la partition A et dans différentes communautés dans B.

n_{01} : Le nombre de paires de sommets qui sont mis dans la même communauté dans la partition B et dans différentes communautés dans A.

Bien que la qualité de partition est une considération essentielle pour choisir une méthode de découverte de communautés, la rapidité d'exécution est aussi un facteur important particulièrement pour les grands réseaux. Le tableau 3.1 récapitule la complexité en temps des déférentes méthodes. Par exemple, la complexité de l'algorithme de Newman [NG04] est $O(n^3)$ ce qui génère un coût de calcul très élevé sur de grands réseaux (des graphes de plus de 1000 sommets). Le temps d'exécution de l'algorithme "eigenvector-based algorithm" est relative à la taille du système, et il est de $O(n^2 log(n))$ pour un graphe dense, cela est considérablement meilleur que le temps d'exécution de "betweenness-based algorithm". La complexité de l'algorithme d'optimisation extrémale est de $O(n^2 log^2(n))$ ce qui offre un gain de performance. L'algorithme "fast" apporte plus de performance ($O(nlog^2(n))$) et convient aux grands réseaux complexes.

La comparaison des différentes techniques est difficile car il est possible que les méthodes les plus performant en termes de temps d'exécution ne peuvent pas identifient des partitions en communautés les plus pertinentes. La recherche d'un compromis entre la qualité des résultats et le coût de calcul reste toujours un défi à relever.

3.9 Conclusion

L'étude des réseaux complexes est une activité en plein essor. Un problème central dans l'étude des réseaux complexes est celui de la détection des communautés : des sous-graphes denses faiblement connectés entre eux. La découverte de la structure communautaire d'un graphe permet d'enrichir nos connaissance sur la structure interne des schéma des interactions mais aussi nous renseigner sur les possibilités d'évolution du graphe. Dans ce chapitre, nous avons passer en revue les principales méthodes de découverte de communautés dans les réseaux complexes. Nous avons discuté leurs apports et leurs inconvénients et proposé une classification des différentes approches.

TABLE 3.1 – Récapitulatif de complexité en temps des différentes méthodes.

Algorithme	**Référence**	**Complexité en temps**
Random-walk algorithm	Newman et al [NG04]	$O(n^3)$
Betweenness-based algorithm	Girvan et al [GN02] Newman et al [NG04]	$O(m^2n)$
Extremal optimization algorithm	Duch et al [DA05]	$O(n^2log^2(n))$
Fast algorithm	Newman [New04a]	$O(nlog^2(n))$
Simulated annealing based algorithm	Guimerà et al [GSPA04b]	Inconnue
Q-state Potts model based algorithm	Reichardt et al [RB04, RB06]	Dépend des paramètres
Local algorithm	Bagrow et al [BB05]	$O(n^3)$
RCLP algorithm	Radicchi et al [RCC+04]	$O(n^2)$
Eigenvector-based algorithm	Newman [New06]	$O(n^2log(n))$
Greedy algorithm	Clauset et al [CNM04]	$O(d.mlog(n))$
Directed modularity maximization algorithm	Leicht et Newman [LN08]	$O(n^2log())$
Divisive algorithm of bipartite networks	Zhang et al [ZWL+08]	Inconnue
Recursive filtration algorithm	Shen et al [SPW+08]	$O(m^2 + (c+1)m)$
Biclique algorithm	Lehmann et al [LSH08]	$O(n^2)$
k-clique "CPM"	Palla et al [PDFV05]	Inconnue
Markov cluster algorithm	Van Dongen [VD00]	$O(n.k^2)$
EAGLE algorithm	Shen et al [SCCH09]	$O(n^2.s)$

n : Le nombre des noeuds du réseau ;

m : Le nombre des liens du réseau ;

c : Le nombre de communauté dans le réseau ;

$k \leq n$: Le nombre maximal des éléments non nuls par colonne [VD00] ;

d : La profondeur du dendrogramme ;

s : Le nombre de cliques maximale.

Deuxième partie

Contribution

Chapitre 4

Extraction de connaissances dans le processus de fouille de l'usage du web

Sommaire

4.1 Introduction

Dans les dernières années il y a eu une croissance exponentielle du nombre de sites web et de leurs usagers. On recense jusqu'à la fin du mois de Juin 2018 environ de 4 Milliards d'internautes (dont 18,5 Millions en Algérie) [1] pour 1,4 Milliards de sites web au monde [2]. Cette croissance phénoménale a produit une quantité énorme de données liées aux interactions d'utilisateurs avec les sites web, stockés par les serveurs web dans des fichiers Logs. Ces fichiers logs peuvent être utilisés par les administrateurs de sites web pour découvrir les intérêts de leurs visiteurs afin d'améliorer le service par l'adaptation du contenu et de la structure des sites à leurs préférences. L'analyse des fichiers logs permet à identifier des modèles du comportement des usagers, ce qui peut être exploité à la personnalisation du web [PPPS03]. Dans un processus général de la fouille des usagers du web (W.U.M) on distingue trois phases principales : prétraitement de données, découverte du modèle et analyse du modèle [TT04].

4.2 La fouille de l'usage du web

La structure fonctionnelle du processus de la fouille des usagers Web est structuré en six modules principaux comme représenter dans le figure suivante (Fig. 4.1).

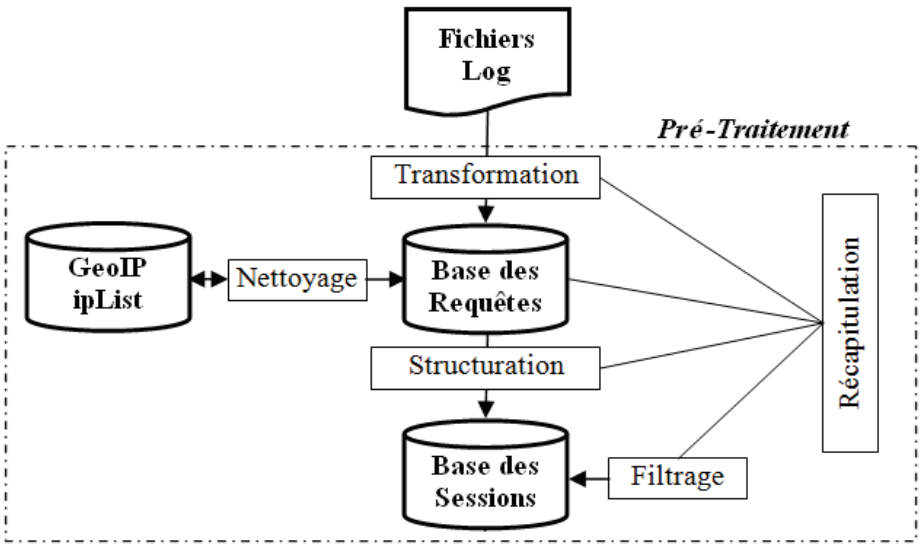


FIGURE 4.1 – Architecture du système de Pré-Traitement des Fichiers Logs.

.

1. http ://www.internetworldstats.com
2. http ://www.netcraft.com

4.2.1 Présentation des fichiers d'accès (Fichiers Logs)

Un fichier Log est un fichier texte qui contient l'historique de tous les requêtes faites par les usagers au serveur web enregistrées en ordre chronologique. Les formats les plus utilisés pour les fichiers logs sont CLF (Common Log Format), ECLF (Extended CLF), NECLF (New ECLF) et IIS. Chaque ligne dans le standard ECLF donne des informations sur la requête de l'utilisateur : ip, nom, login de l'usager, date et l'heure, méthode, URL, prototype, statut, taille et L'URL qui a référencé la requête, et l'agent de navigation.

4.2.2 Prétraitement des données

La première étape du prétraitement, consiste à transformer le fichier log de sa forme textuelle brute, en une forme structurée exploitable. Tout d'abord, nous définissons les terminologie utilisés dans ce travail [Tan05] :

- Un utilisateur : personne qui utilise un navigateur Web.
- Une ressource Web : est une ressource accessible par une version du protocole HTTP ou un protocole similaire.
- Une page Web : est un ensemble des informations, consistant en une (ou plusieurs) ressource(s) Web, identifiée(s) par un seul URI (Uniform Resource Identifier). Exemple : un fichier HTML, un fichier image et un applet Java accessibles par un seul URI constituent une page Web.
- Une session utilisateur : consiste en un nombre délimité de demandes Web explicites d'un utilisateur sur un ou plusieurs serveurs Web.
- Une vue de page (page view) se produit à un moment précis en temps, lorsqu'un navigateur Web affiche une page Web. Une vue de page (ou page) peut être composée de plusieurs pages Web et ressources Web comme.
- Une visite ou une activité de navigation : (comportement de navigation) représente un sous-ensemble de vues de pages consécutives pendant une session utilisateur Les clics de l'utilisateur peuvent être décomposés dans plusieurs visites en calculant la distance temporelle entre deux requêtes HTTP consécutives et si cette distance excède un certain seuil, une nouvelle visite commence(mesurée au moyen d'un seuil de temps ou d'une distance sémantique entre les pages).

Une analyse lexicale permet d'extraire les différents champs de chaque ligne et les enregistrer dans une table composée de plusieurs colonnes. Chaque colonne correspond à un champ spécifique du fichier logs .

Pour faciliter les traitements et l'analyse du fichier Log une structuration de la table Log depuis sa forme plate à une forme relationnelle constituée des tables suivantes (Fig. 4.2) :

1. Les tables IP, Temps, Methode, Ressource, Statut et Agent contient tous les identifiants distincts respectifs ainsi que leurs désignations.

2. Les tables Country et City sont chargées depuis la base de données GEOip de maxmind.com. Elles permettent la localisation géographique des utilisateurs à partir de leurs IP [3].
3. La table Rebot est créée à partir de la base de données iplist.com. Elle permet l'identification des Robots d'indexation du web et des aspirateurs dont de leurs IP sont connus [4].
4. La table User définit l'utilisateur en fonction de l'IP et l'user agent.
5. La tables Requête contient le même nombre d'enregistrements que la table Log, c'est la représentation structurée de cette table. Elle a un identifiant Num, et contient les identifiants de User, Temps, Methode, Ressource et Statut.
6. La tables Session est vide à ce stade, car les sessions ne sont pas encore identifiées.

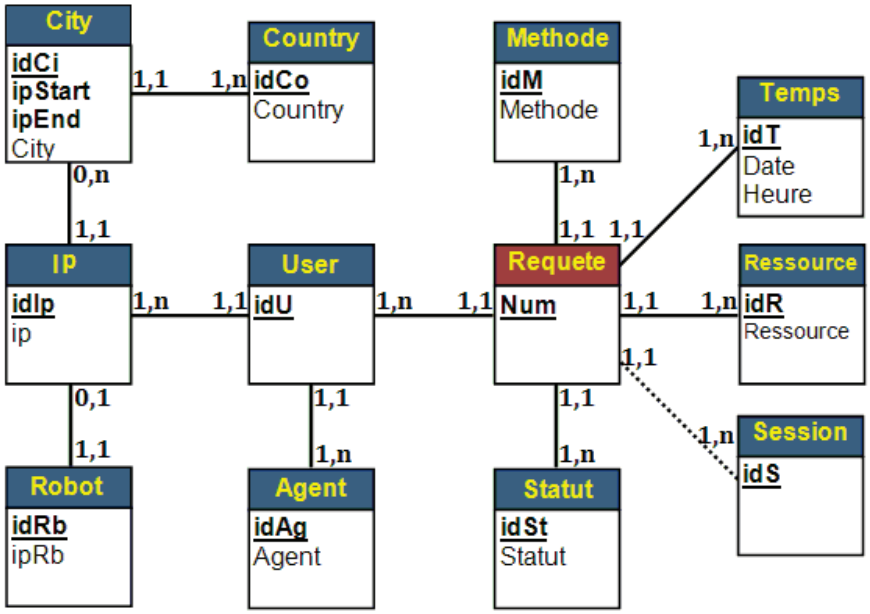


FIGURE 4.2 – Représentation relationnelle de la table Log.

Ensuite un nettoyage de données est prévu pour supprimer les enregistrements inutiles afin de maintenir seulement les actions liées aux comportements de navigation des utilisateurs. Le choix des données à nettoyer dépend de l'objectif du système de personnalisation du site. L'objectif est de développer un système WUM qui offre une personnalisation des liens dynamiques pour les visiteurs du site, ainsi nous ne retenons que des enregistrements associés à des requêtes explicites des usagers et qui représentent effectivement des actions des usagers.

En conséquence, le nettoyage des données consiste à éliminer les requêtes suivantes :

3. http ://geolite.maxmind.com/download/geoip/database/
4. http ://iplists.com/

1. Les requêtes dont la méthode est différente de "Get" ; car ce sont des requêtes non explicites et représentent souvent des accès avec CGI, des visites de robots, etc.
2. Les requêtes dont le statut est différente de "200" ; car ce sont des requêtes ayant un échec.
3. Les requêtes vers des fichiers de style, fichier Multimédia ou scripte js ; car ce sont des requêtes sur des fichiers lancés automatiquement coté serveur avec le chargement de la page demandée.
4. Les requêtes des robots des moteurs de recherches et les aspirateurs. L'identification des ces robots est réalisée par plusieurs méthodes [TK02] :
   (a) IP des robots et des aspirateurs connus sur la base iplist.com,
   (b) Usagers qui accèdent à l'URL "robots.txt",
   (c) Usagers qui ont les mots "crawler", "spider" ou "bot" dans leurs noms d'agent,
   (d) Usagers qui ont une vitesse de navigation rapide et arbitraire inférieur à ΔMin.

Résultats

Notre analyse a été testée sur un fichier accès au serveur web du site de l'Université Ferhat Abas Sétif (www.univ-setif.dz). Le fichier couvre la période du 17/01/2010 04 :03 :30 au 14/02/2010 09 :09 :00 ; La taille du fichier est 100 448 034 Octets. Après sa structuration et chargement dans la table Log, on obtient 365 863 Enregistrements (Requêtes). Après nettoyage on a trouvé 72 089 (19,70%) de requêtes valides (Fig. 4.3), couvrant 9242 ressources utilisées par 10238 usagers.

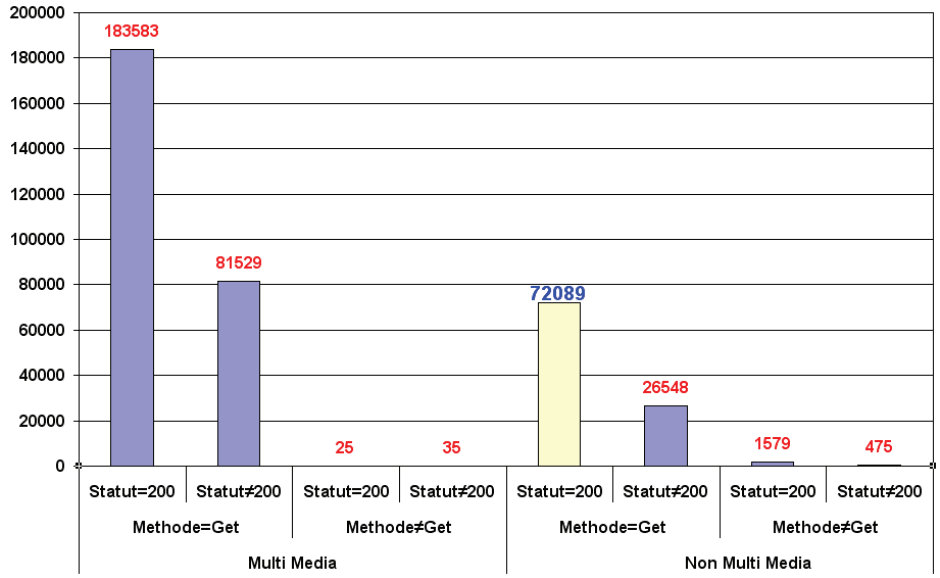


FIGURE 4.3 – Présentation des Catégories des requêtes nettoyées.

4.2.3 Structuration des Sessions

Une session est composée de l'ensemble de pages visitées par le même utilisateur durant la période d'analyse. Chaque session est caractérisée par le nombre de requêtes effectuées par l'utilisateur durant cette session, le nombre de pages consultées et la durée de la session. On considère la combinaison adresse IP plus l'agent de navigation comme étant un critère acceptable pour l'identification d'un utilisateur dans le cadre d'une activité ponctuelle. Le couple (idU,idT) sera utilisé pour identifier les requêtes effectuées pendant une Session idS, sachant que la période entre deux accès ne doit dépasser ΔMax sinon elle sera considérée comme une nouvelle session (Voir [Coo00], [PPK+00] et [SSM05]).

Algorithme 5 : Identification des sessions.

Données : ΔMax ; Requete(Num, idU, idT, idR)
Résultat : Session(idS,idU) ; Navigation(idS,Num,idT,idR)

```
Classer les Requêtes par idU & idT
idS ← 0 ; n ← 0 ;
pour chaque chaque idU faire
    n ← n + 1 ;
    idS ← idS + 1 ; (*Nouvelle Session*)
    Ajouter Session(idS,idU)
    Ajouter Navigation(n,idS,Num,idT,idR) ;
    pour chaque chaque Num faire
        si (idT -idT_Prec)> ΔMax alors
            idS ← idS +1 ; (*Nouvelle Session*)
            Ajouter Session(idS,idU) ;
        n ← n + 1 ;
        Ajouter Navigation(n,idS,Num,idT,idR) ;
```

Résultats

Après la phase du prétraitement, on a identifié les sessions pour un $\Delta Max = 30min$ par l'Algorithme 1, et on a identifié 20 445 sessions. Une classification de ses sessions a été élaboré selon le nombre de ressources accédées et la durée moyenne de la visite (voir Tab. 4.1).

Après l'identification des sessions, on est capable d'identifier les robots des moteurs de recherche et les aspirateurs de sites web en utilisant les méthodes décrite dans la section 2.4. Le tableau (Tab. 4.2) présente le nombre des sessions et navigations identifiées comme étant une navigation automatique par un robot ou un aspirateur. On remarque un chevauchement entre les catégories, expliqué par l'identification de l'usager par les trois méthodes à la fois. La quatrième catégorie contient les sessions ayant une seule navigation.

En fixant la vitesse moyenne minimum de navigation entre les pages à 10 Secondes, et ignorant toutes les ressource qui sont utilisés moins de 70 fois, on garde 3 372 Sessions pour

Catégorie	Nombre	Durée Moyenne	
		Session	Ressource
nb <= 1	14523	0.00	0.00
1 < nb <= 5	3 744	374,56	233,45
5 < nb <= 10	1 109	761,42	115,10
10 < nb <= 50	984	1 912,85	103,70
50 < nb <= 100	48	8 891,02	127,10
> 100	37	30 365,51	145,27
Total	20 445	109,97	188,28

TABLE 4.1 – Classification des sessions par nombre de ressources.

Catégorie	ip	Agent	Usager	Session	Navigation
Agent (crawler, spider ou bot)	593	54	616	8 356	20 564
Resouce = robots.txt	379	64	385	5 451	21073
IP (Robot, Aspirateur)	546	221	800	6 516	13 180
Nombre Global	886	277	1 146	9 557	30 841
Session ayant une seul Navigation				14 523	14 523
Après nettoyage				4 339	34 715

TABLE 4.2 – Identification des robots et des aspirateurs.

16 431 Navigations qui accèdent à 63 ressources.

4.3 Les règles d'association pour l'identification de l'usage du Web

Après structuration du fichier Log sous forme de sessions, on va adapter les données obtenues pour l'application de l'algorithme des règles d'association.

4.3.1 Notation de motif web fréquent

L'ensemble des items : On définit l'ensemble des items comme étant l'ensemble des ressources, et on obtient $I = R = \{r_1, r_2, ..., r_{n_r}\}$ définit dans la section 3.2

La base de données : On définit l'ensemble des transactions comme étant l'ensemble des sessions extraites dans le paragraphe 3.3, et on obtient $\beta = S = (s^{(1)}, s^{(2)}, ..., s^{(n_s)})$; Où chaque transaction est un ensemble de ressources (Requêtes) demandées pendant une session donnée.

k-Itemset : est un Itemset contenant k items.

Une règle d'association : est de la forme $X \Rightarrow Y$, où X et Y sont des itemsets ; $X \subseteq I$, $Y \subseteq I$, et $X \cap Y = \emptyset$.

Le support de l'itemset X : est le nombre de transactions de la base β contenant X divisé par le nombre total de transactions.

$$sup(X) = \frac{|\{t \in \beta / X \subseteq \beta\}|}{|\beta|} \tag{4.1}$$

Le support d'une règle d'association $X \Rightarrow Y$: est le rapport entre le nombre de transactions de Bita contenant $X \Rightarrow Y$, et le nombre total de transactions.

$$sup(X \Rightarrow Y) = \frac{|\{t \in \beta / X \cup Y \subseteq \beta\}|}{|\beta|} \tag{4.2}$$

La confiance d'une règle : est le rapport entre le nombre de transactions de B contenant $X \Rightarrow Y$, et le nombre de transactions de B contenant X.

$$conf(X \Rightarrow Y) = \frac{|\{t \in \beta / X \cup Y \subseteq \beta\}|}{|\{t \in \beta / X \subseteq \beta\}|} = \frac{sup(X \cup Y)}{sup(X)} \tag{4.3}$$

Itemset fréquent : On dit que l'itemset X est un itemset fréquent si :

$$sup(X) \geq minsup \tag{4.4}$$

4.3.2 L'algorithme Apriori appliqué au WUM

La recherche de règles d'associations dans un ensemble de transactions s'opère en deux temps :

1. On cherche les ensembles d'items fréquents, c'est-à-dire ceux qui apparaissent un nombre minimum de fois dans l'ensemble des transactions.
2. On génère les règles d'associations pertinentes, c'est-à-dire celles qui vérifient simultanément la contrainte minimale sur le support et la confiance.

La recherche de tous les sous-ensembles fréquents consiste à déterminer parmi l'ensemble de toutes les parties de $X = (X_1; X_2; ...; X_p)$ les sous-ensembles fréquents, c'est-à-dire présents dans un nombre assez conséquent de transactions.

L'algorithme Apriori consiste à chercher des ensembles fréquents de cardinal $k+1$ à partir des ensembles fréquents de cardinal k. Ainsi pour trouver les ensembles fréquents ayant deux items, on utilisera exclusivement les ensembles fréquents ayant un item.

Le nombre d'ensembles fréquents diminue avec le nombre d'items : il y a moins d'ensemble fréquents à 2 items qu'à un item. Cette propriété permet ainsi de restreindre la taille de l'espace à explorer pour trouver tous les ensembles fréquents nécessaires à la deuxième étape de l'algorithme qui comporte deux points :

1. Pour chaque ensemble fréquent X_a on génère tous les sous-ensembles non vides.
2. Pour chaque sous-ensemble non vide $X_b \subset X_a$
   si $\frac{supp(X_a)}{supp(X_b)} > c_0$ Alors génère la règle $(X_b \Rightarrow (X_a - X_b))$

Algorithme 6 : L'algorithme Apriori

Données : $\beta, minsup$
$L_1 \leftarrow$ { Ensemble de 1-item qui apparaissent dans au moins $minsup$ transitions }
$k \leftarrow 2$
tant que $L_{k-1} \neq \emptyset$ **faire**
  /* Génère l'ensemble de candidats */
  $C_k \leftarrow Generate(L_{k-1})$
  pour chaque $t \in \beta$ **faire**
    /* Sélection des candidats de C_k présent dans t*/
    $C_t \leftarrow SousEnsemble(C_k, t)$ **pour chaque** $c \in C_t$ **faire**
      $count[c] = count[c] + 1$
  $L_k \leftarrow \{c \in C_k \mid count[c] \geq minsup\}$
  $k \leftarrow k + 1$
return $U_k L_k$

4.3.3 Résultats des règles d'associations

La dernière phase de notre travail, consiste à appliquer l'algorithme des règles d'associations sur la base des sessions trouvées dans la section 4.4.

L'algorithme Apriori, a comme données en entrée un ensemble d'items (l'ensemble des ressources) et une base de données des transitions. Avec les paramètres :

- seuil de filtrage des requêtes $\epsilon = 100$;
- Support min des items $minsup = 5$;
- Confiance min des règles $confmin = 60$.

L'Algorithme Apriori opère en deux phases :

- La cherche des ensembles d'items fréquents ; On a trouver :
  - 105 1-items,
  - 60 2-itmes,
  - 10 3-items.
- La recherche des règles d'associations, a permet d'extraire deux catégories de règles :
  - 1ère Catégorie de règles :
    - $R_3, R_{26} \rightarrow R_1$: support = 2 , Confiance = 89.
    - $R_3, R_{34} \rightarrow R_1$: support = 2 , Confiance = 92.
    - $R_2, R_6 \rightarrow R_1$: support = 4 , Confiance = 73.
    - $R_3, R_{26} \rightarrow R_1$: support = 2 , Confiance = 89.
    - $R_3, R_{26} \rightarrow R_1$: support = 2 , Confiance = 89.
  - 2ème Catégorie de règles :

* $R_5, R_{17} \rightarrow R_1$: support = 3 , Confiance = 90.
* $R_5, R_{31} \rightarrow R_1$: support = 2 , Confiance = 90

Sachant que ces ressources ont la sémantique suivante :

- R_1 : Page d'accueil
- R_2 : Page de Formation Poste Graduation
- R_3 : Page du Concours d'accès à la Post Graduation (MAGISTER) 2008-2009
- R_{26} : Document Word sur le Concours de la faculté des sciences économiques et sciences de gestion.
- R_{34} : Document Word sur le Concours de la faculté des sciences.
- R_5 : Galerie d'images.
- R_{17} : Page 1 de la Galerie d'images.
- R_{31} : Page 2 de la Galerie d'images.

Donc on distingue clairement qu'il y a deux types de navigations :

- Les usagers qui ont été intéressés par les concours de poste graduation et Magistère. Et on doit bien souligner que la période d'étude (du 12/10/2008 au 12/11/2008) est la période des concours dans les déférentes facultés.
- La deuxième catégorie de navigation qui a été décelé, et la navigation entre les déférentes galeries d'images.

4.4 Découverte de communautés de l'usage du web

Dans les réseaux complexes, la présence de groupes de sommets (communautés) fortement liés entre eux et faiblement liés avec l'extérieur fonde la propriété de structure de communauté. Les communautés sont des groupes de sommets qui partagent beaucoup de propriétés communes des rôles semblables dans le réseau en question. Ainsi, les communautés peuvent correspondre aux groupes de pages Web traitant le même sujet [FLGC02], aux modules fonctionnels tels que les cycles et les voies dans les réseaux métaboliques [GA05], [PDFV05], aux groupes d'individus relatifs dans les réseaux sociaux [GN02], [LN04], et aux subdivisions dans les chaînes alimentaires [Pim79], [KFM$^+$03] etc. Dans ce chapitre, nous analysons le comportement des utilisateurs dans le processus de fouille de données d'usage du Web en capturant les caractéristiques sociales et sociétales de la structure du réseau à travers le Web [SMLD12]. Il s'agit d'introduire la notion de communauté Web pour modéliser un ensemble d'objets Web. Notre objectif est de tirer profit des pouvoir classificatoire des algorithmes de de détection de communautés dans l'extraction des nouveaux patterns afin d'améliorer l'efficacité des tâches d'extraction du processus WUM [SMLD11].

4.4.1 Création d'un graphe à partir des sessions

Notre idée consiste à modéliser les sessions des usagers web avec un graphe $G(V, E)$. Sachant qu'une session idS est une suite de pages visitées idR_i. Nous procédons à la transformation de l'ensemble des sessions en un graphe $G(V, E)$ (Fig. 4.4), tel que :

- Chaque ressource idR correspond à un nœud du graphe,

- Chaque couple (idR_v,idR_w) de pages visitées dans l'ordre chronologique du temps durant une session idS correspond à une arrête dans le graphe entre ces deux nœuds.

On utilise l'algorithme 2 pour la construction de la matricer d'adjacence A (une matrice binaire symétrique).

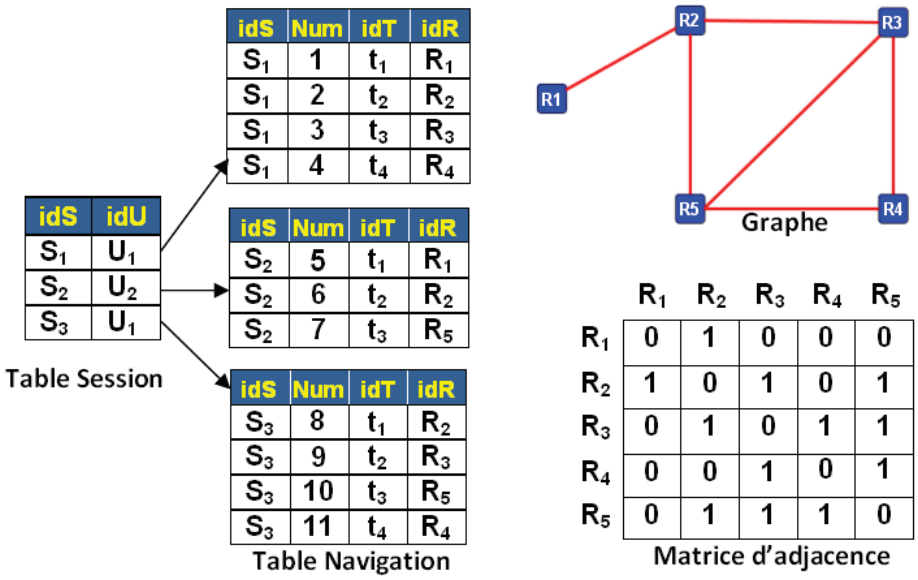


FIGURE 4.4 – Création du Graphe a partir des sessions.

Algorithme 7 : Création du Graphe

Données : Session(idS,idU) ; Navigation(idS,Num,idT,idR)
Résultat : A[Card$_{Ressource}$,Card$_{Ressource}$]

```
m ← Card_Ressource ;
pour i ← 1 à m faire
    pour j ← 1 à m faire
        A[i, j] ← 0 ;(* initialisation de A par des 0*)
pour chaque idS faire
    i← idR ;(* Première ressource de la session *)
    pour chaque idR de la session idS faire
        j← idR ;
        A[i, j] ← 1 ;
        A[j, i] ← 1 ;
        i← j ;
```

Si on prend en considération le nombre de fois que les couples (idR_v,idR_w) apparaissent dans les sessions on obtient un graphe pondéré en affectant le poids adéquat aux éléments de la matrice d'adjacence .

4.4.2 Notion de Communautés Web

Une communauté de page Web, notée C_i, $\forall i \in 1..K$, est un sous-graphe de pages web auxquelles les utilisateurs ont accédé plusieurs fois. Autrement dit, nous cherchons des sous graphes fortement connectés dont les liens internes représentent les utilisateurs qui parcourent des pages Web de proximité similaires et les liens externes expriment la tendance des utilisateurs à naviguer sur le site Web pour y accéder à d'autres types de pages Web [DBS13] [SM12b].

4.4.3 Mesures de la qualité de l'identification des communautés

Comment savoir si les communautés détectées sont bonnes ou non et comment évaluer une telle partition ? Quelle est la meilleure partition pour le graphe en question ? Pour répondre à ces questions, [NG04] ont introduit une mesure de la qualité d'une partition particulière du graphe appelée "modularité".

Supposons une partition particulière de notre graphe G en k communautés. Soit $P(k*k)$ une matrice symétrique, ses éléments p_{ij} représente le nombre de tous les arrêtes dans le graphe qui relient les sommets de la communauté C_i aux sommets de la communauté C_j.

La trace de la matrice P : $Tr(P) = \sum_i p_{ii}$ représente le nombre de tous les arrêtes entre les sommets dans les mêmes communautés. Une valeur élevée de la trace indique une bonne partition en communautés. Cependant, la trace n'est pas toujours un bon indicateur de la qualité de la partition, puisque en regroupant tous les sommets dans une seule communauté

cela produit une valeur maximale égale à $2m$ [NG04].

On définit :

- $s_i = \sum_j p_{ij}$: la somme de n'importe quelle ligne (ou colonne) de P qui correspond au nombre de tous les arrêtes reliées aux sommets de la communauté i.

- $\frac{s_i s_j}{4m^2}$: estimation du nombre de tous les arrêtes entre la communauté C_i et C_j générées par un modèle aléatoire.

Ainsi, la mesure de modularité est définit comme suit :

$$Q = \sum_i (p_{ii} - \frac{s_i^2}{4m^2}) = Tr(P) - \frac{\|P^2\|}{4m^2} \tag{4.5}$$

4.4.4 Identification des communautés depuis un graphe non pondéré

Dans la deuxième phase du processus WUM qui consiste à la découverte du comportement de l'utilisateur lors de la navigation, nous avons utilisé l'algorithme Fast [CNM04]. Au départ, on a n communautés composées d'un sommet chacune. Les sommets sont regroupés itérativement en nouvelles communautés plus grandes et le regroupement de communautés est poursuivi jusqu'à l'obtention d'une seule communauté regroupant tous les sommets. Une structure hiérarchique de communautés est ainsi construite.

Supposons que G est le graphe résultant de la phase de prétraitement de données, V est l'ensemble des nœuds (ressources) et E représente l'ensemble des arêtes. Pour appliquer l'algorithme hiérarchique ascendant (Fast) sur le graphe G, nous utilisons la formule de modularité qui a été définie par [New04a] comme suit :

$$Q = \frac{1}{2m} \sum_{r_v r_w} \left[A_{r_v r_w} - \frac{k_{r_v} k_{r_w}}{2m} \right] \delta\left(c_{r_v}, c_{r_w}\right) \tag{4.6}$$

Tel que :

m : le nombre total des arrêtes du graphe,

k_{r_v} : le degré du nœud,

c_{r_v} : la communauté qui contient le nœud,

$$A_{r_v r_w}(\textit{matrice d'adjacence}) = \begin{cases} 1 & \text{si } r_v \text{ et } r_w \text{ sont connectées} \\ 0 & \text{sinon} \end{cases}$$

$$\delta(c_{r_v}, c_{r_w}) = \begin{cases} 1 & \text{si } r_v \text{ et } r_w \text{ sont dans la même communauté} \\ 0 & \text{sinon} \end{cases}$$

La variation de la modularité est utilisée pour fusionner deux communautés en une seule. Ainsi, elle est définie comme suit :

$$\Delta Q_{r_v r_w} = \begin{cases} \frac{1}{2m} - \frac{k_{r_v} k_{r_w}}{(2m)^2} & \text{si } r_v \text{ et } r_w \text{ sont connectées} \\ 0 & \text{sinon} \end{cases} \tag{4.7}$$

L'algorithme se déroule comme suit :

1. Initialement, on considère que chaque communauté se compose d'un seul sommet.
2. Rechercher la paire des communautés où la variation de modularité est maximale.
3. Construire une matrice ayant une ligne et une colonne en moins, en regroupant les éléments dans une nouvelle communauté. La mise à jours des éléments de la matrice et de la matrice en prenant en compte les communautés jointes et la modification des structures des données utilisées par l'algorithme.
4. Répéter l'étape 2 jusqu'à ce qu'a l'obtention d'une seule communauté.

4.4.5 Identification des communautés depuis un graphe pondéré

Dans cette section nous présentons la deuxième approche que nous appliquons pour la découverte des communautés à partir d'un graphe pondéré [SM12a]. Nous attribuons au graphe des poids aux arrêtes selon le nombre de fois qu'on a visité une ressource en tenant compte des successions de navigation entre ces ressources (voir Alg. 2). On pourrait obtenir des informations sur le comportement des graphes pondérés très simplement en les convertissant aux multigraphes non pondérés tel que chaque lien de poids n peut être remplacé par n liens parallèles de poids et ainsi toutes les techniques qui peuvent être appliquées sur des graphes non pondérés peuvent être appliquées aux multigraphes [New04b]. D'après ce constat, nous appliquons le même algorithme Fast tout en insérant la fonction de poids $W_{r_v r_w}$ (entier positif), la matrice d'adjacence s'écrit :

$$A_{r_v r_w}(\textit{matrice d'adjacence}) = \begin{cases} W_{r_v r_w} & \text{poids de connection de } r_v \text{ et } r_w \\ 0 & \text{sinon} \end{cases}$$

L'algorithme Fast procède de même en partant d'un seul nœud qui constitue une communauté de taille 1 et à chaque étape, les deux communautés (et / ou nœuds) produisant un changement maximal de la valeur ΔQ sont fusionnées, et ce jusqu'à ce que tous les nœuds appartiennent à une seule communauté. Cette approche qui se base sur la matrice d'adjacence du graphe pondéré extrait l'information potentiellement utile contenue dans les poids des liens ce qui améliore la qualité de partition du réseau en communautés.

4.4.6 Résultats des approches de découverte de communauté

Après la phase de prétraitement des fichiers Logs, nous avons visualisé les activités du site de l'université de Farhat Abbas en construisant le graphe qui montre la structure du réseau originale (comportant 63 nœuds et 1 214 arrêtes). Dans le but d'identifier la structure des communautés et de détecter les comportements de navigation des utilisateurs, nous avons implémenté deux approches d'identification de communautés .

Une communauté peut être décrite comme collection de sommets dans un graphe qui sont fortement reliés entre eux-mêmes mais faiblement relié du reste du graphe [New04a]. La progression entière des deux algorithmes de découverte de communautés est illustrée par les

deux dendrogrammes qui représentent le regroupement des sommets depuis le graphe initial au graphe résultant partitionné en communauté comme le montre la figure (Fig. 4.5). Les coupures horizontales de l'arbre hiérarchique représentent clairement toutes les communautés à chaque niveau.

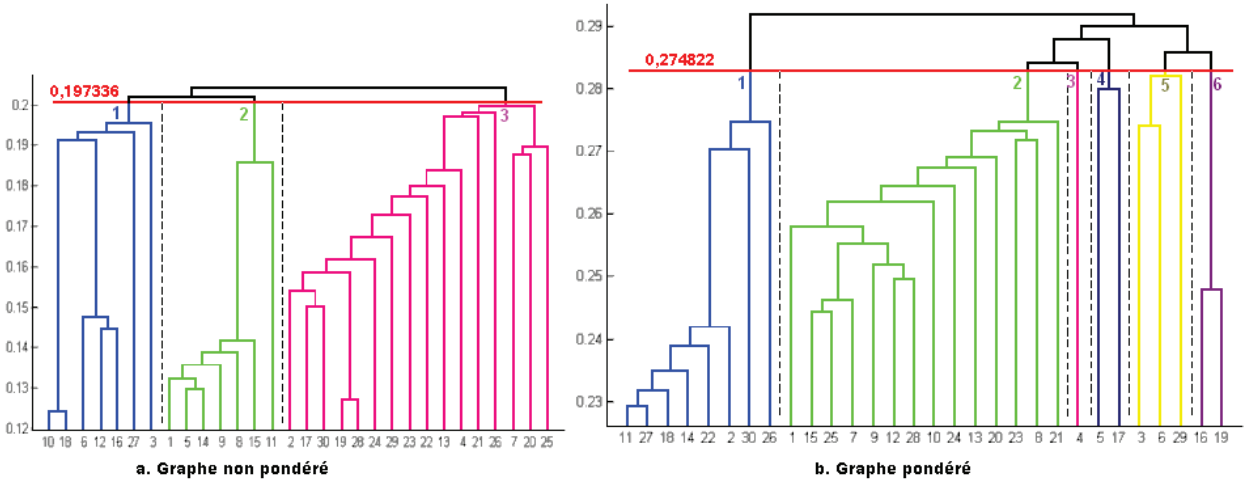


FIGURE 4.5 – Dendrogrammes de découvert de communautés sur le Graphe d'accès

4.4.7 Discussion

Les algorithmes calculent $\Delta Q_{r_v r_w}$ pour fusionner la paire de communautés C_{r_v} et C_{r_w} qui résulte d'une valeur maximale de $\Delta Q_{r_v r_w}$. Afin de déterminer la section optimale de deux dendrogrammes, nous précisons la valeur optimale de changement de modularité comme le montre la figure (Fig. 4.6) . Ainsi, nous constatons que la valeur optimale obtenue dans le cas du graphe non pondéré est de 0.197 et dans le cas du graphe pondéré la valeur est de 0.274.

L'algorithme dans sa version non pondérée a identifié trois (3) communautés. Les communautés découvertes permettent de donner une vue globale sur l'utilisation du site :

- Le premier groupe montre les visites effectuées aux pages Web de formation et de pédagogie, donc les utilisateurs de cette communauté sont intéressés aux modalités d'inscription, au contact des départements, au règlement des études et à la formation assurée par l'université.
- Le deuxième groupe regroupe les accès aux pages des manifestations scientifiques (par exemple appel à communication, soumission des papiers), ainsi que les visites effectuées aux pages de la formation en post-graduation (concours d'accès aux PG, modalités d'inscription).

- Le troisième groupe identifie les différentes visites sur des pages de valorisation et recherche, les facultés, les liens utiles, la recherche bibliographique.

Ces analyses ont permis d'identifier des classes homogènes de visiteurs. Cependant, nous avons observé que l'algorithme dans sa version non pondérée n'a pas abouti à un regroupement pertinent des nœuds qui ont un comportement similaire dans la troisième communauté.

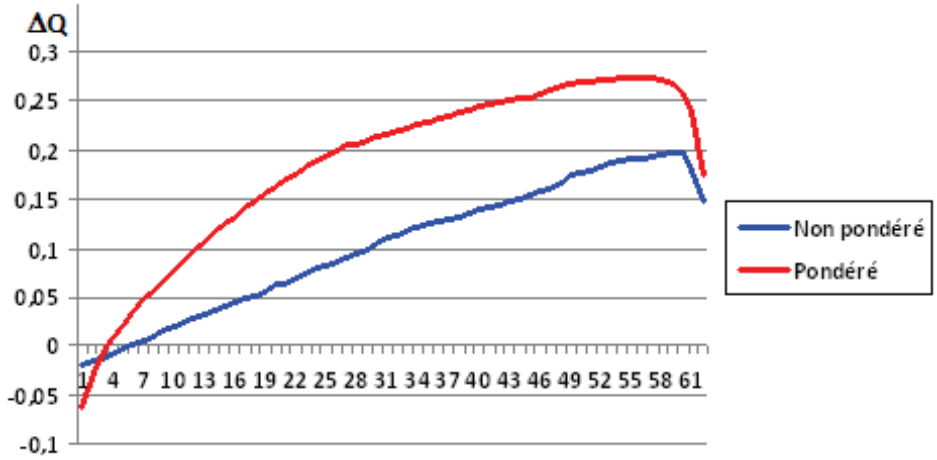


FIGURE 4.6 – Changement de la Modularité par le Fast Algorihme

L'algorithme dans sa version pondérée a identifié six(6) communautés représentées dans le dendrogramme de droite (Fig. 5) :

- La première communauté illustre les visites aux pages de coopération et celles du Rectorat de l'université.
- La deuxième communauté identifie les accès aux pages des différentes facultés et départements. Les usagers sont intéressés aux formations assurées par les départements, les modalités d'inscription, etc.
- Les visiteurs intéressés par les pages Web de recherche et de valorisation ont été regroupé dans la troisième communauté.
- La quatrième communauté illustre les visites effectuées aux pages de la post-graduation.
- Nous avons aussi observé la cinquième communauté qui représente les accès aux rubriques de contact et des liens utiles.
- La sixième communauté a détecté les accès aux revues et ressources électroniques en particulier l'accès à la revue Science Direct.

Nous avons remarqué que l'utilisation de l'information du poids sur le graphe a été très utile pour mieux extraire les intérêts des internautes, par exemple la découverte de la communauté d'accès aux revues montre que l'objectif de ces utilisateurs est bien précis (la recherche). L'algorithme nous a permis d'identifier des usagers d'intérêt commun tel que les groupes 1 et 2.

La détection de communautés réalisée a permet de bien comprendre la structure et le comportement des utilisateurs du site web de l'université de farhat Abbas de sétif. Les résultats obtenus seront exploitées dans le processus de personnalisation du site Web.

4.5 Conclusion

Notre travail a pour objectif l'utilisation du processus du W.U.M et son expérimentation afin d'extraire des modèles d'accès Web et d'améliorer le design du site web tout en introduisant des techniques performant dans les phases de ce processus à savoir la phase de prétraitement et la phase de découverte et d'analyse du modèle. D'après les résultats expérimentaux obtenus dans la première phase de prétraitement, nous avons extraie les requêtes les plus pertinents et regroupé dans un certain nombre de sessions utilisateur que nous avons employé pour construire le graphe modélisant les activités de notre site.

Dans la deuxième phase, l'implémentation de l'algorithme de règle d'association nous a permis d'analyser le comportement des utilisateurs.

Dans une dernière phase ; nous avons centré notre étude sur l'intégration des méthodes de découvertes de communautés dans le processus WUM. En effet, nous avons montré qu'un ensemble de sessions des utilisateurs peut être ramené à un graphe. Cette nouvelle modélisation permet de capter plus d'informations sur les similarités de comportement de navigation des utilisateurs ainsi que les relations entre les ressources Web. Ceci est fait en appliquant l'algorithme Fast sur un graphe non pondéré et un graphe pondéré. Les résultats obtenus sont significatifs et satisfaisant.

Chapitre 5

Méthodes d'identifications de communautés dans le processus de fouille d'usage du Web

Sommaire

5.1 Introduction

Les méthodes de partitionnement de données cherche à grouper des objets génériques. La stratégie dans ce domaine est typiquement d'établir une fonction de similarité entre les objets, puis d'exécuter un algorithme chargé de trouver des clusters où les similarités internes sont fortes. On y trouve des techniques de partitionnement hiérarchique (les communautés fusionnent ou se divisent au fur et à mesure de l'exécution) et de coupure de graphe (en minimisant le nombre d'arêtes entre les parties). Dans ce chapitre, nous proposons une nouvelle méthode hiérarchique de découverte de communautés basée sur les probabilités d'accès aux ressources lors des sessions des utilisateurs [SMD14]. En conséquence, nous avons définit un coefficient de pertinence qui quantifie l' information utile contenant dans les séquences

d'accès de l'utilisateur et reflète la probabilité d'appartenance d'un utilisateur à une communauté [SMD15], [DBSG17]. Nous présentons les résultats obtenus en faisant une comparaison avec la méthode de marche aléatoire.

5.2 Résultats de la phase de prétraitement

Nous avons présenté une méthodologie complète pour le pré-traitement des fichiers logs dans [SM10] [SMG09a]. Dans nos recherches, notre méthodologie de prétraitement a été testée sur les fichiers logs stockés sur le serveur du site Web de l'université de Ferhat Abbas de Sétif. Ce site web est disponible sur l'URL http ://www.univ-setif.dz. Les logs traités couvrent les activités du site web pendant la période allant de 18-12-2011 à 04h :04 :57 au 19-01-2012 à 08h :47 :04. Après la structuration, nous avons obtenu 1 708 385 requêtes. Ensuite, nous avons appliqué une étape de nettoyage de données afin de maintenir seulement les actions liées aux comportements de navigation des utilisateurs. Après le nettoyage, nous avons obtenu 314 115 (i.e. 18,38%) requêtes valides pour 27 520 utilisateurs qui ont accèdé aux 23 872 pages web.

Afin d'identifier les sessions des utilisateurs, nous avons appliqué un algorithme de structuration [Nas05] pour déterminer les sessions en tenant compte du temps d'accèes maximum $\Delta Max_t = 30min$ entre deux accèes consécutifs [Coo00, PPK+00]. Notre phase de prétraitement résulte de 50 131 sessions pour 314 115 navigations. Ensuite, nous avons identifié les requêtes à l'origine des robots Web [TK02]. Le temps minimum écoulé entre les accès consécutifs est également fixé à $\Delta Min_t = 10seconds$. En conséquence, nous avons obtenu 31 017 sessions pour 175 681 navigations.

Après l'identification des sessions des utilisateur, nous avons effectué une étape de filtrage de données pour supprimer les ressources les moins demandées. Pour chaque ressource r_i, nous considérons le nombre de sessions NS_i qui exploitent une ressource r_i. A cet effet, nous avons supprimé toutes les requêtes dont $NS_i < \epsilon$ tel que ϵ est un seuil bien déterminé (100). Nous avons donc obtenu un nombre de pages significatifs (136) que nous allons utilisé lors de la prochaine phase de découverte de pattern.

5.3 Méthode de découverte de communautés

Plusieurs études sur les propriétés mathématiques des réseaux complexes ont révélé que ces réseaux partagent des propriétés macroscopiques. Parmi ces propriétés, les travaux analysent des propriétés prototypes telles que l'effet petit-monde [WS98], l'échelle-libre [BA99], les propriétés dynamiques telles que la diffusion [BP01], [ESMS03] et les propriétés structurelles telles que la structure communautaire [Moo01],[FLGC02]. La structure de communauté permet de comprendre la relation entre un simple noeud dans la microscopie et des groupes dans la macroscopie. De plus, les communautés (ou les clusters ou les modules) sont des

groupes de sommets qui partagent probablement des propriétés communes et/ou jouent des rà´les similaires dans un réseau complexe [BLM+06]. Dans ce qui suit, nous proposons un algorithme de découverte de communautés et nous décrivons les concepts et les définitions que nous avons utilisés. Ensuite, nous allons employer la méthode proposée afin de trouver le centre d'intérêt des utilisateurs du site Web étudié et ainsi le restructurer selon leurs besoins. Enfin, nous illustrons les résultats expérimentaux obtenus.

5.3.1 Algorithme de découverte de communautés basé sur le coefficient de pertinence

Les méthodes de découverte de communautés basées sur des graphes pondérés sont souvent incapables de détecter une partie très significative de la structure des communautés car ils ignorent les informations contenues dans le poids du lien. A cette fin, nous définissons un coefficient de pertinence qui conserve l'information potentiellement utile dans les différentes sessions des utilisateur. L'algorithme proposé est défini en termes de sessions qui ont été identifiées dans la phase de prétraitement. Une session est un ensemble contigu de ressources qui ont été consultées par le même utilisateur lors de sa visite du site Web.

5.3.1.1 La création des sessions

Nous définissons :

Soit : $R = \{r_1, r_2, ..., r_{n_r}\}$ l'ensemble de toutes les ressources distinctes du site Web.

Soit : $U = \{u_1, u_2, ..., u_{n_u}\}$ un ensemble de tous les utilisateurs qui ont accédé au site

Soit : $S = \{s^{(1)}, s^{(2)}, ..., s^{(n_s)}\}$ un ensemble de sessions de navigation, de telle sorte que chaque session d'utilisateur soit définie comme suit :

$s^{(i)} = (u^{(i)}, t^{(i)}, r^{(i)})$, tel que :

- $u^{(i)} \in U$: est l'identificateur d'utilisateur ;
- $t^{(i)}$: est le temps d'accès durant toute la session ;
- $r^{(i)}$: est l'ensemble de toutes les ressources demandées au cours de la i^{ime} session (selon le temps d'accès correspondant) ;
- $r^{(i)} = ((t_1^{(i)}, r_1^{(i)}), (t_2^{(i)}, r_2^{(i)}), ..., (t_{(n_i)}^{(i)}, r_{(n_i)}^{(i)}))$ tel que $r_j^{(i)} \in R$.

5.3.1.2 La création du graphe

Une fois que les sessions utilisateur ont été identifiées, nous devons les utiliser pour extraire le graphe Web qui représente le réseau analytique en utilisant l'algorithme de restructuration défini dans [Nas05]. Une base de session peut être considérée comme une séquence

de paires de sessions définissant la transition entre les pages. Chaque ressource r_i correspond à un noeud du graphe. Chaque couple (r_i, r_j) de pages visitées dans l'ordre chronologique du temps pendant une session correspond à un lien dans le graphe entre ces deux noeuds.
Soit : $G = (V, E)$ un graphe décrivant une donnée prétraitée de la base de session avec V l'ensemble des noeuds et E l'ensemble des arêtes. $A_{i,j}$: est la matrice d'adjacence du réseau. $A_{i,j}$est égale à 1 si les ressources r_i et r_j sont connectées, sinon 0.

5.3.1.3 Coefficient de pertinence

Afin d'identifier le nombre de fois que les couples (r_i, r_j) apparaissent dans les sessions, nous calculons les transactions des utilisateurs.
Soit : $\beta = \{T^{(1)}, T^{(2)}, ..., T^{(n_s)}\}$ l'ensemble des transactions. Chaque transaction $T^{(i)}$ se compose d'un ensemble de ressources $r_j^{(i)}$ (item) demandées pour une session donnée i.

$k - Itemset$: est un ensemble d'item qui contient k items. Nous associons un support pour chaque élément. Le support est définit comme le nombre de fois qu'un item apparait dans la base de données. Le support d'une pair (i, j) représente le support de $k - item$ de second ordre $(k = 2)$. C'est le nombre d'occurrences de (i, j) dans les sessions divisées par le nombre total de toutes les sessions.

$$Supp(i, j) = \frac{|\{t \in \beta / (i, j) \subseteq \beta\}|}{|\beta|} \tag{5.1}$$

Nous prenons en compte le nombre de fois que les couples (i, j) apparaissent dans les sessions en attribuant un poids approprié aux liens du graphe comme le montre la figure 5.1.

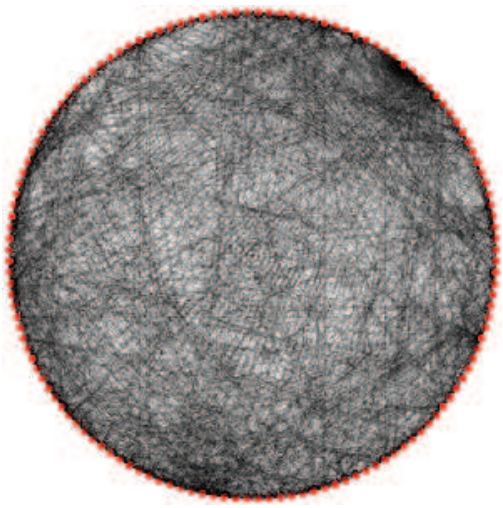


FIGURE 5.1 – Création du réseau analytique depuis les sessions.

Nous définissons une matrice S qui représente la probabilité d'avoir des relations entre

les noeuds dans les différentes communautés. La matrice S_{ij} peut être décrite comme suit :

$$S_{i,j} = A_{i,j} * Supp(i,j) \tag{5.2}$$

Cette matrice exprime le comportement des internautes au cours des différentes transactions ce qui reflète tendance des noeuds à former des communautés. Ensuite, nous définissons un facteur qui fait référence à la fréquence d'apparition du lien (i,j) pour chaque séquence $k-$item. En effet, nous pouvons définir :

$$f_{i,j} = \frac{S_{i,j}}{d_i} \tag{5.3}$$

Tel que d_i est le degrée du noeud i.

Notre idée est basé sur le fait qu' un noeud a une forte probabilité d'être membre à une communauté s'il présente un nombre d'occurrence significative au sein d'un même groupe. Par conséquent, nous définissons un coefficient que nous avons appelé "coefficient de pertinence". Il est définit comme suit :

$$\lambda(i,C) = \frac{\sum_{j \in C} s_{i,j}}{\sum_{j \in V} s_{i,j}} \tag{5.4}$$

$$= \sum_{j \in C} f_{i,j} \tag{5.5}$$

Dans ce travail, la détection des liens intercommunautaires est basé sur le fait que ces liens se situent dans des zones moins denses. C'est la valeur de pertinence du noeud i par rapport à une communauté c, divisé par le nombre total de ces valeurs possibles pour toutes les communautés.

5.3.1.4 Algorithme de détection de communautés

La modularité mesure la qualité d'une division particulière du réseau. La fraction des liens reliant des sommets de la même communauté est plus grande que la fraction des liens intra-communaitaires [NG04]. De ce fait, la modularité est définie comme suit : Supposons une partition particulière d'un réseau en k communautés. Soit e une matrice symétrique , ses éléments représente la fraction de tous les liens dans le réseau qui relient les sommets de la communauté i aux sommets de la communauté j. La trace de cette matrice donne la fraction de liens dans le réseau qui relie les sommets dans la même communauté :

$$Tr_e = \sum_i e_{i,i} \tag{5.6}$$

Une valeur élevée de la trace indique une bonne division en communautés. Cependant, la trace elle-même n'est pas un bon indicateur de la qualité de la division puisque, par exemple,

le faite de mettre tous les sommets dans une seule communauté donnerait la valeur maximale de la trace $Tr = 1$ sans donner d'informations sur la structure de la communauté. La somme sur les lignes (colonnes) représente la fraction des liens qui se connectent aux sommets au sein de la communauté i, est alors définie comme suit :

$$a_i = \sum_j e_{i,j} \tag{5.7}$$

Si le réseau ne présente pas de structure de communauté, ou si les partitions sont attribuées sans tenir compte de la structure sous-jacente, alors, la valeur attendue de la fraction de liens en des partitions peut être estimée. Si le réseau ne possède pas la propriété de structure de communauté, alors, la valeur prévue des fractions des liens dans une partition peut être estimée. C'est la probabilité qu'un sommet d'extrémité d'un lien soit dans la communauté i, donc a_i, multiplier par la fraction des liens qui se termine par un sommet dans la communauté i, donc a_i. En conséquence, le nombre de liens intra-communauté qui sont prévus est égale à $a_i a_i$. De plus, nous avons la fraction réelle des liens dans une partition qui est égale à e_{ii}. Nous pouvons donc comparer les deux fractions et effectuer la somme de toutes les partitions du graphe.

$$Q = \sum_i (e_{ii} - a_i^2) = Tr_e - \|e^2\| \tag{5.8}$$

Cette quantité mesure la fraction des liens reliant des sommets de la même communauté sans la valeur prévue de la même quantité avec la mêmes partition en communautés mais en employant des connexions aléatoires entre les sommets. Cependant, cette fonction de modularité soulève des limites de résolution [FB07] et ne prends pas le chevauchement de la structure de la communauté. Par conséquent, plusieurs travaux [NMCM09] ont proposé des extention de la modularité pour quantifier la structure communautaire en chevauchement dans des réseaux orientés et pondérés.

Dans ce travail, nous avons proposé de quantifier la structure de la communauté en chevauchement dans un réseau pondéré. Dans le cas des communautés qui ne se chevauchent pas, un noeud appartient à une seule communauté. Le coefficient de pertinence reflète la probabilité d'appartenance d'un noeud i à une communauté C. Nous avons ainsi réécrit la modularité dans l'équation (6.8) :

$$Q = \frac{1}{2m} \sum_{c \in C} \sum_{i,j \in V} [s_{i,j} - \frac{d_i d_j}{2m}] \lambda(i,c) \lambda(j,c) \tag{5.9}$$

Tel que :

$$0 \leq \lambda_{(i,C)} \leq 1 \tag{5.10}$$

Les noeuds sont divisés en communautés de sorte que si le noeud i appartient à une seule communauté C, $\lambda_{(i,C)}$ est égal à 1 ; Si le noeud i n'appartient plus à la communauté C, alors $\lambda_{(i,C)} = 0$.

Nous exploitons le coefficient de pertinence pour révéler la structure de la communauté d'un réseau en utilisant un algorithme agglomératif que nous avons défini dans l'algorithme 8.

Algorithme 8 : Méthode de Découverte de Communautés basé sur la coefficient de

Data : V

· initialisation $P^{(0)} = \{C_1^{(0)}, C_2^{(0)}, ..., C_n^{(0)}\}$

répéter

· Calculer la Coefficient de Pertinence entre chaque deux communautés adjacentes

· Sélectionner les deux communautés C_1 et C_2 de $P^{(k)}$ qui maximise la modularité

· Créer la nouvelle partition $P^{(k+1)}$

· Mise à jour de S_{ij} pour tous les communautés adjacentes à la nouvelle communautés.

jusqu'à *Avoir une seul communauté*;

Couper le Dendrogram au niveau du partitionnement qui maximise la fonction de Qualité

5.3.2 Algorithme de marche aléatoire

Pons et al [PL06] ont proposé une méthode hiérarchique agglomérative basée sur les marches aléatoires. L'algorithme Walk-Trap utilise le rapport intuitif selon lequel, si un marcheur est dans une communauté, il est fort probable qu'il reste dans la même communauté à l'étape suivante. Une marche aléatoire dans le graphe G est un processus en temps discret. Sa probabilité de transition P est définie comme suit :

$$P_{i,j} = \frac{A_{i,j}}{w_i} \tag{5.11}$$

tel que : $P^t_{i,j}$ est la probabilité de transition du noeud i vers le noeud j par une marche aléatoire d'une longueur t.
Le temps est discrétisé $(t = 0, 1, 2, ...)$ et un marcheur est localisé à tout moment t sur un noeud du graphe G.
Le marcheur se déplace à chaque instant de manière aléatoire et uniforme vers l'un de ses noeuds voisins. La matrice de transition peut être définie à partir de la matrice d' adjacence A du graphe et de la matrice diagonale D selon la formule suivante :

$$P = D^{-1}A \tag{5.12}$$

Soit $\rho(t)$ est le vecteur de distribution de probabilité de la position du marcheur après t étape tel que : $\forall i \in v, \rho_i(t) >= 0, \sum_{i \in v} \rho_i(t) = 1$ Ainsi, un marcheur a de grandes chances de rester pour une courte marche dans sa communauté d'origine. La distribution $\rho(0)$ donne la position initiale du marcheur, par exemple, si le marcheur commence à partir d'un seul noeud i, $\rho_i(0) = 1, \forall j \neq i, \rho_j(0) = 0$ Le marcheur peut également se déplacer à partir de plusieurs noeuds avec des probabilités différentes pour chaque noeud. La probabilité d'atteindre un noeud j au temps $(t+1)$ est directement liée aux probabilités de position à l'instant t et aux probabilités de transition $P_{i,j}$:

$$\forall j, \rho_j(t+1) = \sum_{i \in V} \rho_i(t) P_{ij} \tag{5.13}$$

L' equation est écrit comme suit :

$$\rho(t+1) = P^T \rho(t) \tag{5.14}$$

P^T est la matrice transposée de P. Nous pouvons donc calculé $\rho(t)$:

$$\forall j, \rho_j(t) = \sum_{i \in V} (P^t)_{ij} \rho_i(0) \tag{5.15}$$

Ce qui permet d'interpréter la valeur P^t_{ij} comme la probabilité de passer du noeud i au noeud j durant t étape. La propriété de réversibilité des étapes de marche aléatoire indique que les probabilités P^t_{ij} et P^t_{ji} sont dépendant. Les transitions aléatoires d'un noeud donné $i \in V$ sont définit par les probabilités $(P^t_{kj})_{k \in V}$. Ces probabilités correspondent à la i^{ime} ligne

de la matrice P^t et sont décrites par une colonne vectorielle $P^t_{.i}$. Pons et al [PL06] ont définit des points importants afin de comparer deux noeuds i et j en fonction de la distance entre eux :

- Les sommets d'une même communauté ont tendance à voir les sommets éloignes de la même façon, ainsi si i et j sont dans la même communauté et k dans une autre communauté il y a de fortes chances que $\forall k, P^t_{ik} \approx P^t_{jk}$.
- Si deux noeuds i et j sont dans la même communauté, la probabilité P^t_{ij} sera sûrement élevée, mais une probabilité importante P^t_{ij} ne signifie pas que i et j sont nécessairement dans la même communauté.
- La probabilité P^t_{ij} est liée au degré du sommet d'arrivée $d(j)$ car il est plus facile d'atteindre les sommets de fort degré par une marche aléatoire.

La distance r_{in} entre les noeuds du graphe est définie comme suit :

$$r_{ij} = \sqrt{\sum_{k=1}^{n} \frac{(P^t_{ik} - P^t_{jk})^2}{w(k)}} \tag{5.16}$$

Où D est la matrice de diagonale, et r est une distance euclidienne sur R^n. La généralisation de cette distance entre les communautés est définit comme suit :

$$P^t_{C.} = \frac{1}{|C|} \sum_{i \in C} P^t_{i.} \tag{5.17}$$

Tel que : $C \subset V$ est une communauté. $P^t_{C.}$ est le vecteur de probabilité qui représente la longueur des marches aléatoires t commençant uniformément à partir d'un noeud de la communauté C. Cela permet de définir la distance entre un noeud i et une communauté C :

$$r_{iC} = ||D^{\frac{1}{2}} P^t_{C.} - D^{\frac{1}{2}} P^t_{i.}|| \tag{5.18}$$

Cette métrique de distance est liée aux approches spectrales qui sont basées sur le fait que deux noeuds appartenant à la même communauté ont des composants similaires sur les principaux vecteurs propres [PL06]. L'algorithme calcule les composants connectés et applique ensuite un processus agglomératif qui découvre des communautés sur des sous-graphes connectés.

5.4 Résultats expérimentaux

Dans la phase de la découverte du modèle, nous avons identifier la structure de la communauté et en conséquence nous serons capable de détecter le comportement de navigation des utilisateurs, ce dernier peut être exploités dans le processus de personnalisation du Web. Les méthodes de découverte de communautés que nous avons utilisées sont des algorithmes agglomératifs qui commencent avec chaque nœud dans une seule communauté fusionnant itérativement les paires de communautés selon leurs distances.

5.4.1 Description du graphe

Le degré de distribution du réseaux nous permet d'avoir une idée sur sa structure comme le montre la figure 5.2. Le degré d'un noeud est sa propriété structurelle la plus fondamentale. Nous remarquons que les degrés de tous les nœuds sont répartis autour de la moyenne, ce qui correspond à 2.5, et un grand nombre de ressources se sont connectées avec ceux qui partagent un sujet similaire. Le tableau 5.1 présente les résultats de l'analyse du réseau étudié et synthétise l'information sur les degrés.

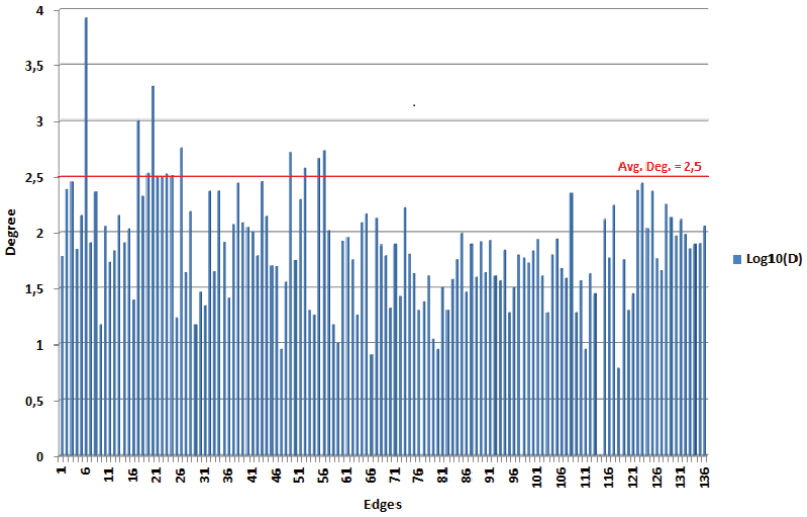


FIGURE 5.2 – Distribution des degrées

TABLE 5.1 – Description du graphe.

Catégorie	Valeur
Nombre des liens (n)	136
Nombr des noeuds (m)	2 456
Degrée moyen	377.588
Densité	0.134

5.4.2 Mesure de la qualité de partitionnement

La qualité de partition résulté par notre algorithme et par l'algorithme de random walk est illustrée par le graphe de la modularité sur la figure (5.3). L'algorithme du coefficient de pertinence identifie 12 communautés avec une valeur maximale de modularité qui est égale à 0.347. L'algorithme de marche aléatoire trouve une structure communautaire de 13 communautés avec une valeur de modularité qui est égale à $Q_{max} = 0.247$. En effet, les deux méthodes découvrent une structure communautaire significative. La méthode proposée 5.3 résulte d'une valuer plus grande de modularité par rapport à l'algorithme Random walk.

Et cela est dû au fait que l'algorithme de coefficient de pertinence permet aux nœuds de détecter leurs communautés localement. La méthode de Pons et al [PL06] tire profit des propriétés spectrales sans recourir à un calcul explicite des valeurs propres et des vecteurs propres. Sa complexité en temps est de $O(nmlog(n))$, cependant, lorsque la longueur des étapes est importantes, la qualité des résultats diminue. Les informations contenues dans les liens pondérés par la valeur de coefficient de pertinence peuvent être efficaces pour détecter les communautés et préserver l'information utile décrivant le comportement des utilisateurs. Lorsque le degré d'un nœud est faible, la capacité de classification de l'algorithme diminue. Les deux méthodes sont capables de découvrir des communautés et ont des résultats plus similaires du nombre de communautés.

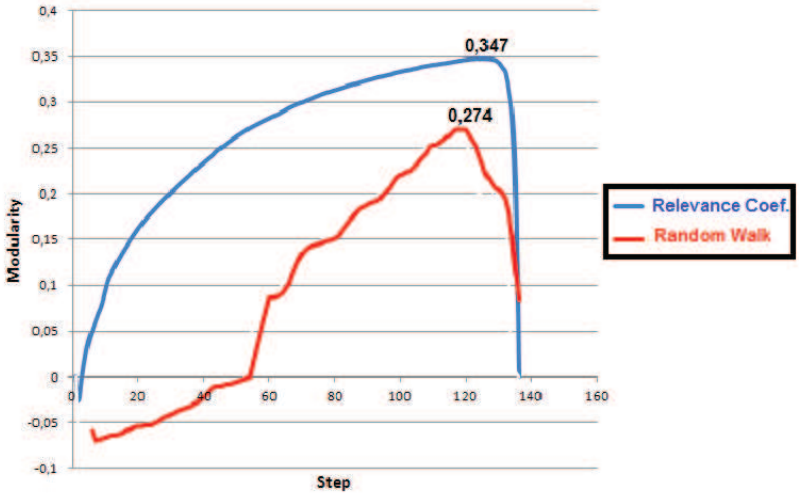


FIGURE 5.3 – Comparaison de la variation de modularités par l'Algorithme des Coefficients de pertinence et l'algorithme Walk-Trap

Nous utilisons un graphe pondéré (figure 5.1) qui représente la fréquence des accès aux ressource par les utilisateurs lors des sessions. La complexité en temps de notre algorithme est de $O(mn)$. Les communautés sont identifiées comme des sous-ensembles de nœuds, lesquels leurs connexions internes sont plus denses tandis que leurs connexions intercommuantaires sont faiblement reliées (voir figure 5.4. Nous avons identifié l'intérêt des visiteurs aux pages Web au sujet de la "Post-graduation" et les accès aux pages des concours, cette communauté présentée par 12 noeuds de couleur mauves intenses. Nous avons aussi obtenu une communauté regroupant les accès aux pages de "la Faculté des sciences sociales" (elle contient 09 noeuds en rouges). Une communauté se composant de noeuds blues claire illustre les visites aux pages Web de "coopération" et celles du "vice-rectorat de l'université". Sur la figure, les 48 noeuds marqués par le marron représentent les accès aux pages de diverses "activités d'enseignement" au niveau des facultés et des départements. La communauté jaune détecte les accès aux "revues et aux ressources électroniques" en particulier les revues du site Web "Science Direct". En conséquence, la structure trouvée identifier clairement les sessions des utilisateurs et leurs comportement de navigation.

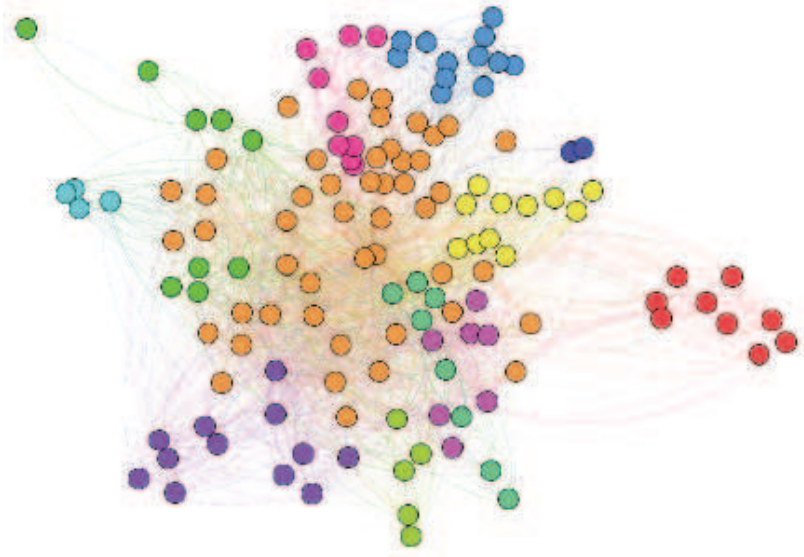


FIGURE 5.4 – La structure des communautés identifiée

5.5 Conclusion

Ce travail est consacré à la découverte des patterns intéressants de fouille d'usage du Web. L'idée de la méthode proposée est d'introduire un coefficient de pertinence qui caractérise l'accès à chaque ressource Web par un utilisateur. Ainsi, elle est révélatrice de la structure du graphe et peut être utilisée pour comparer la probabilité d'appartenance d'un nœud à une communauté. On peut alors agréger de façon itérative, et couper le dendrogramme là où la modularité est maximale. Notre approche révèle une structure des communautés pertinente et donne une description significative des communautés existantes. Ceci nous as permis d'envisager une meilleure personnalisation du site Web de l'université de Sétif.

Chapitre 6

Modélisation de l'usage du web par les chaines de Markov

Sommaire

6.1 Introduction

La fouille d'usage du web est un processus d'extraction et d'analyse de données permettant de détecter le comportement de navigation des utilisateurs sur un site web [Tan05]. Cette tâche est basée sur la fouille de données, dans laquelle de nombreuses étapes sont nécessaires pour la réalisation de l'ensemble du processus [Coo00]. Le prétraitement des données comprend le nettoyage des données, la normalisation, la transformation, le filtrage et la synthèse [PPPS03]. Les phases de découverte et d'analyse utilisent une méthode d'extraction d'information pour identifier un pattern représentant le modèles d'accès des utilisateurs. Ces

modèles d'extraction des patterns semblent insuffisants pour extraire des informations comportementales et structurelles à partir des sites Web. Ainsi nous proposons dans ce chapitre une méthode d'extraction d'information capable de révéler les comportements de navigation des utilisateurs.

Par conséquent, les principales contributions de ce travail [SMDL18] sont les suivantes :

- Modélisation de toutes les séquences fréquentes au cours de différentes sessions d'utilisateur par un processus stochastique basé sur les chaînes de Markov.
- Le processus stochastique est employé pour crée un graphe orienté et pondéré qui révèle l'information utile présentée dans les séquences de navigation.
- Ce processus stochastique dérivé est utilisé pour chercher la probabilité stationnaire et réécrire la modularité pour avoir des partitions optimales.
- Proposer une méthode de détection de communautés en modifiant l'algorithme défini par chavent et Lechevalier [CL02]. L' algorithme crée des prototype de communautés tout en optimisant un critère d'adéquation.
- Ce travail présente une méthode hybride combinant la simplicité des séquences fréquentes, l'efficacité des chaînes de Markov et la robustesse des méthodes de détection de communautés. Nos expérimentations qui ont été effectués sur le site Web msnbc.com et le site Web de l'université de Ferhat Abbas de Sétif montrent des résultats satisfaisants.

Ce chapitre est organisé comme suit : section **??** explique notre motivation et les objectifs visés par l'amélioration de la conception d'un système web. Section 6.3 décrit la phase de prétraitement de données, ce qui augmente la qualité des données obtenues à la fin de cette étape. Notre méthode est présentée à la section 6.4. Les résultats expérimentaux sont décrits dans la section 6.5 et enfin, une conclusion générale est présentée dans la section 6.6.

6.2 Contexte et motivations

Les objectifs de cette contribution consiste en l'amélioration de l'architecture du site Web, l'analyse des performances du système, et la compréhension des interaction et des motivations des utilisateurs [MNLM06, SCDT00, TJK99]. Dans la littératures plusieurs méthodes ont été proposées pour la personnalisation d'un site Web afin de découvrir des modèles d'utilisation intéressants à partir de données Web et mieux répondre aux besoins des applications Web [FL05]. Il existe des méthodes qui servent aux système de recommandation [JT98, MM09, ZGH03]. Le filtrage collaboratif est également connu comme une technique de recommandation basée sur des éléments de prédiction [BLO+06]. En outre, certaines approches adaptent le contenu du site Web aux visiteurs afin de fournir des informations personnalisées. Ces information sont exploité pour développer des systèmes Web intelligents [Bon08]. De nombreuse approches inspirées des méthodes de détection communautaire ont été

appliquées pour améliorer l'éxtraction des données et fournir une structure communautaire significative [New09]. Dans le travail [JCY13], les auteurs ont proposé un algorithme efficace basé sur la détection des communautés depuis un graphe, et il est utilisé pour transformer la recherche de motif (l, d) en une séquences sous la contrainte ZOMOPS (zero, one or multiple occurrences of the motif instances per sequence)(Zéro, une ou plusieurs occurrences des instances des motifs par séquence), où l est la longueur d'un motif et d est le nombre maximal de mutations entre une instance de motif et le motif lui-même, afin de découvrir des sous-graphes denses dans un graphe. Lu et al [LZG14] ont proposé un algorithme de détection de communauté basé sur la séquence de similarité alors que les similitudes de noeuds sont triées par ordre décroissant pour obtenir une séquence. Ensuite, les liens de noeuds sont fusionnés selon la séquence pour construire une structure communautaire pertinente en utilisant une approche agglomérative. Motivée par le pouvoir classificatoire des méthodes de découverte de communautés, nous allons donc proposer une nouvelle méthode adaptée à l'identification du comportement de navigation de l'utilisateur. Dans notre travail, nous recherchons des motifs fréquents en observant toutes les transitions de la page v_i à la page w_i pendant une session pour chaque utilisateur. Le modèle stochastique employé estime les probabilités de transition en utilisant une approche agglomérative pour découvrir des sous-graphes denses à partir du graphe extrait des comportements de navigation.

6.3 Prétraitement des données

Dans la phase de prétraitement, nous transformons le fichier log brut en un ensemble de transactions pour réduire la quantité de données à analysées et, en même temps, améliorer sa qualité [LMN11]. Dans notre travail, nous avons effectué une étape de prétraitement sur les fichiers logs en utilisant la méthodologie que nous avons définie dans [SMLD11]. Notre méthode de prétraitement a été testée sur les fichiers logs du serveur du site web de l'université de Sétif 1 (Algérie). Le fichier traité couvre les activités du site web pendant la période de 18-12-2011 à 04 :04 :57 jusqu'au 19-01-2012 à 08 :47 :04. Après la transformation des données, nous avons obtenu 1 708 385 requêtes. Ensuite, nous avons appliqué une étape de nettoyage des données afin de maintenir uniquement les requêtes liées au comportement de navigation des utilisateurs. Le nettoyage des données a résulté de 314 115 requêtes valides (c'est-à-dire 18,38%) pour les utilisateurs 27 520 qui ont accès aux 23 872 pages. La table 6.1 montre les résultats des sessions et l'identification du robot.

6.4 Détection des communautés pour le WUM

6.4.1 Problématique

La fouille des usagers du Web vise l'analyse et la découverte des modèles intéressants de l'utilisateur. Dans cette section, nous nous intéressons à détecter les communautés d'utilisa-

TABLE 6.1 – Identification des sessions et des robots.

Category		User	Page	Session	Navigation
Total before identification robots		27 520	23 872	50 131	**314 115**
Identification	Ip	303	8 362	5 083	80 484
method	URL	783	19 128	14 543	124 519
of Robots	Agent	1 369	20 223	18 496	137 073
Total robots		1 438	20 266	19 114	138 434
Total after identification robots		26 082	7 259	31 017	175 681
Total after removal of ΔMin_t Req.		10 899	5 753	12 297	**147 711**

teurs en exploitant leurs comportements de navigation Web au lieu d'explorer la structure de liens hypertexte des pages Web. Ce qui est plus utile pour découvrir l'intérêt commun d'un groupe d'utilisateurs. Nous modélisons les séquences de navigation des utilisateurs par les chaînes de Markov. Nous avons introduit un processus stochastique pour calculer davantage la similarité entre les différentes sessions de navigation des utilisateurs et récrire la fonction de qualité pour comparer les partitions. De plus, nous proposons un algorithme de découverte de communautés que nous appelons CDMMC (Community Discovery Method based on Markov Chain, CDMMC)(Méthode de Détection de Communautés basé sur les Chaines de Markov) qui extrait un modèle intéressant. Notre démarche nous permet de comprendre comment les utilisateurs accède aux site Web et, en conséquence, nous permet de proposer le design le plus appropriée au système Web étudier.

Soit S un ensemble de n_s sessions, durant lesquelles les utilisateurs accèdent à des pages web n. Nous définissons :

- $S = s_1, s_2, ..., s_{n_s}$: un ensemble de sessions de navigations.
- $V = v_1, v_2, ..., v_n$: un ensemble de pages visitées ;
- $S_i = (v_i^{(1)}, v_i^{(2)}, ..., v_i^{(n_i)})$: les séquences des pages visitées pendant la session S_i

Le nombre total de pages visitées n_v est :

$$n_v = \sum_{i=1}^{n_s} n_i \tag{6.1}$$

Tel que n_i est le nombre de pages visitées durant la session i.

Supposons que de nombreux utilisateurs aient accédé aux plusieurs pages, par exemple, $V = \{v_1, v_2, v_3, v_4, v_5\}$ durant trois sessions, nous obtenons alors l'ensemble de sessions suivant :

$$S_1 = (v_1, v_2, v_3, v_4, v_2, v_3, v_5, v_1, v_4)$$
$$S_2 = (v_1, v_4, v_2, v_3, v_5, v_1, v_3, v_4)$$
$$S_3 = (v_2, v_3, v_4, v_2, v_3, v_5)$$

La méthode que nous introduisons est basée sur un modèle stochastique qui extrait l'information significative contenue dans les séquences fréquentes de navigations. Dans ce qui suit, nous décrivons les étapes principales de notre méthode.

6.4.2 Étape 1 :recherche des motifs fréquents

Dans cette phase, nous modélisons la navigation de l'utilisateur pendant les sessions en utilisant une méthode formelle pour l'identification des ensembles d'items fréquents. En général, une base de données contenant les sessions de navigations peut être considérée comme une séquence de paires définissant la transition entre les pages Web [AS94]. Ainsi, nous représentons les ensembles distincts de paires afin de trouver les motifs fréquents et créer le graphe de comportement de navigation.

Formellement, nous définissons la paire (v_i, w_i) comme une transition de la page v_i vers la page w_i durant la session $i \in [1..n_s]$.
Soit M_i la sequence des transitions de toutes les paires au cours de la session i :

$$M_i = \Big((v_i^{(t)}, w_i^{(t)}) | w_i^{(t)} = v_i^{(t+1)}; \quad t = [1..n_i - 1]\Big) \tag{6.2}$$

De ce fait, les transitions complètes qui se produisent dans toutes les sessions décrivant l'ensemble de paires de séquences observées seront notées :

$$M = \Big((v, w) | (v, w) \in U_{i=1}^{n_s} M_i\Big) \tag{6.3}$$

En conséquence, la cardinalité de M est égale à $|M| = n_v - n_s$. Supposons que chaque paire (v, w) dans la séquence M ait un numéro d'occurrence $n_v(v, w)$. Cette valeur indique les informations potentielles qui peuvent se produire au cours des sessions de navigation n_s en considérant toute paire de pages qui ont été visitée au moins deux fois non consécutive. Le nombre d'occurrence est donné par :

$$n_v(v, w) = \sum_{(x,y) \in M} \delta_v\Big((v, w), (x, y)\Big) \tag{6.4}$$

Tel que δ_v est la fonction Kronecker delta définit comme suit :

$$\delta_v((v, w), (x, y)) = \begin{cases} 1 & \text{if } v = x \text{ and } w = y \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

Pour caractériser l'accès globale depuis chaque page du site Web, nous définissons le numéro d'occurrence d'une page de départ v comme suit :

$$n_{out}(v) = \sum_{w \in V} n_v(v, w) \tag{6.6}$$

Nous pouvons également définit le numéro d'occurrence de la page w, s'il s'agit d'une page d'arrivée comme suit :

$$n_{in}(w) = \sum_{v \in V} n_v(v, w) \tag{6.7}$$

Notre idée est d'extraire tous les ensembles fréquents en utilisant le nombre d'occurrences de paires des pages. Nous calculons le support de toutes les pages pour toutes les paires (page de départ, page d'arrivée) dans la base de sessions. Par conséquent, nous prenons en compte les pages qui ont un support supérieur à un seuil minimal (*minsup*). Le support de la page v est son numéro d'occurrence durant les sessions divisé par le nombre total de toutes les sessions.

$$Supp(v) = \frac{n_{out}(v) + n_{in}(v)}{2n_s} \tag{6.8}$$

Le support d'une paire (v, w) représente le support de k-item de second ordre ($k = 2$), autrement dit, c'est le nombre d'occurrences de (v, w) durant les sessions divisé par le nombre total de toutes les sessions. Il s'est écrit comme suit :

$$Supp(v, w) = \frac{n_v(v, w)}{n_s} \tag{6.9}$$

6.4.3 Étape 2 : La création du graphe

Soit $G = (V, E)$ un graphe pondéré et orienté qui représente les sessions de navigation des utilisateurs, tel que :

V est l'ensemble des noeuds.

E est l'ensemble des liens.

E représente les paires distinctes de M ($|E| = m \leq |M|$), définies comme suit :

$$E = \Big\{(v, w)|(v, w) \in M\Big\} \tag{6.10}$$

Nous définissons la matrice d'adjacence A à partir de la paire fréquente de pages (v, w) :

$$A_{i,j} = n_v(v_i, v_j) \tag{6.11}$$

Voir la Fig. 6.1 à titre d'exemple : nous avons calculé le degré des nœuds strictement lié à la fréquence d'accès à une paire de pages ce qui nous as permis de creér le graphe de comportement de navigation.

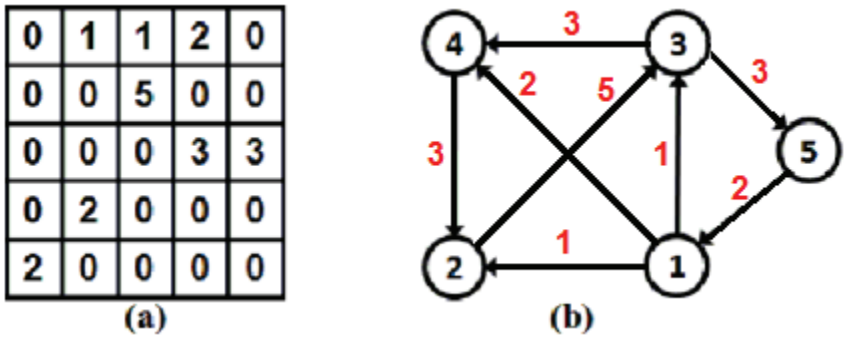


FIGURE 6.1 – (a) Matrice d'adjacence - (b) Graphe de Navigation

6.4.4 Étape 3 : Création d'un modèle stochastique

Considérons une variable aléatoire X observable dans un processus de chaîne de Markov. X représente la navigation des utilisateurs depuis la page de démarrage $v^{(t)}$ jusqu'à une page de fin $w^{(t+1)}$. L'observation des sessions aléatoires peut être représentée par le processus de Markov [Fel08a, Fel08b], tel que :

$$S_i = \left\{ (X^{(i,t)}, X^{(i,t+1)}) | t = 1..n_i - 1 \quad , \quad (X^{(i,t)}, X^{(i,t+1)}) \in V \right\} \tag{6.12}$$

L'échantillon pour toutes les sessions est défini comme suit :

$$\begin{aligned} E_x = \Big(& (X^{(1,1)}, X^{(1,2)}), (X^{(1,2)}, X^{(1,3)}), ..., (X^{(1,n_1-1)}, X^{(1,n_1)}), \\ & (X^{(2,1)}, X^{(2,2)}), (X^{(2,2)}, X^{(2,3)}), ..., (X^{(2,n_2-1)}, X^{(2,n_2)}), ... \\ & (X^{(n_s,1)}, X^{(n_s,2)}), (X^{(n_s,2)}, X^{(n_s,3)}), ..., (X^{(n_s,n_n-1)}, X^{(n_s,n_n)}) \Big) \end{aligned} \tag{6.13}$$

La taille de l'échantillon E_x est :

$$n_E = \sum_{i=1}^{n_s} (n_i - 1) = \sum_{i=1}^{n_s} n_i - n_s = n_v - n_s \tag{6.14}$$

n_E : représente le nombre total de transition pendant toutes les sessions, en tenant compte du fait qu'il n'y a pas de transition entre la session S_i et la prochaine session S_{i+1}. n_E peut être également calculé comme suit :

$$n_E = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} \tag{6.15}$$

L'estimation de la transition de la page v à la page w est la suivante :

$$E_x = \left\{ (X^{(t)}, X^{(t+1)}) | t = 1..n_E \quad , \quad X^{(t)}, X^{(t+1)} \in V \right\} \tag{6.16}$$

La probabilité d'occurrence d'une transition pendant toutes les sessions est définie comme suit :

$$P_E = \frac{1}{n_E} \tag{6.17}$$

Dans le cas d'une chaîne de Markov de premier ordre, les probabilités de transition ne dépendent pas de l'historique de tout le processus comme l'état précédent est pris en considération. Ainsi, les probabilités de transition d'état peuvent être décrites :

$$P[X^{(t)} = v \quad \text{and} \quad X^{(t+1)} = w] = \frac{A_{vw}}{n_E} \tag{6.18}$$

tel que :

$$P[X^{(t)} = v] = \frac{\sum_{w=1}^{n} A_{v,w}}{n_E} \tag{6.19}$$

et

$$P[X^{(t+1)} = w] = \frac{\sum_{v=1}^{n} A_{v,w}}{n_E} \tag{6.20}$$

Nous observons que la propriété de Markov est définie comme suit :

$$P[X^{(t+1)} = w | X^{(t)} = v] = \frac{P[X^{(t)} = v \quad \text{and} \quad X^{(t+1)} = w]}{P[X^{(t)} = v]} = \frac{A_{v,w}}{\sum_{w=1}^{n} A_{v,w}} \tag{6.21}$$

Ensuite, nous définissons la matrice de probabilité de transition à la page w sachant que on est à la page v, $P = [P_{v,w}]$, de sorte que :

$$P_{v,w} = \frac{A_{v,w}}{\sum_{w=1}^{n} A_{v,w}} \quad \text{with} \quad \forall v : \sum_{w=1}^{n} P_{v,w} = 1 \tag{6.22}$$

Nous définissons aussi le vecteur de probabilité sachant que v est une page de départ $P_{out} = [Pout_v]$, de sorte que :

$$Pout_v = \frac{\sum_{w=1}^{n} A_{v,w}}{\sum_{v=1}^{n} \sum_{w=1}^{n} A_{v,w}} \tag{6.23}$$

Le vecteur de probabilité de w une page d'arrivée $P_{in} = [Pin_v]$ est donné par :

$$Pin_w = \frac{\sum_{v=1}^{n} A_{v,w}}{\sum_{v=1}^{n} \sum_{w=1}^{n} A_{v,w}} \tag{6.24}$$

6.4.5 Étape 4 : Rechercher la probabilité stationnaire

Soit $\pi^{(1)}$ un vecteur de probabilité initial de l'état de navigation des utilisateurs dans le temps $t = 1$, nous l'écrivons comme suit :

$$\pi^{(1)} = \{\pi_i^{(1)}\}_{i=1..n} \tag{6.25}$$

Ainsi, nous obtenons :

$$\pi_i^{(1)} = P[X^{(1)} = i] \quad \text{where } \forall i \pi_i \geq 0 \quad , \quad \sum_{i=1}^{n} \pi_i = 1 \tag{6.26}$$

Donc, nous avons le vecteur de probabilité de l'état de navigation dans le temps $t + 1$:

$$\pi_i^{(t+1)} = P * \pi_i^{(t)} \tag{6.27}$$

Il existe l'unique vecteur de probabilités à l'état stable π^* [KHMG03], [ANTT02] qui est le vecteur propre à gauche de P noté par :

$$\lim_{t \to +\infty} \pi^t = \pi^* \tag{6.28}$$

Où π^* est le vecteur propre associé à la valeur la valeur propre est $(\lambda = 1)$.

En conséquence, pour obtenir la distribution stationnaire de la chaine de Markov nous cherchons :

$$\pi_i^* = P * \pi_i^* \tag{6.29}$$

6.4.6 Étape 5 : Découverte de communautés

Dans cette section, nous réécrivons la modularité en fonction de la probabilité de transition et nous proposons une nouvelle approche de découverte de communautés.
Plusieurs algorithmes ont été proposé pour extraire les similarités entre les communautés dans les réseaux complexes. Dans le travail [PL06], les auteurs ont utilisé une mesure de distance entre les nœuds pour identifier les similarités. La distance Euclidienne entre les nœuds a été implémentée dans un algorithme de clustering hiérarchique basé sur les propriétés spectrales de Laplacienne [DM04]. Marta et al [MRAALAN07] ont observé l'organisation du système complexe dans chaque niveau et ont définit un concept de proximité entre tous les pairs de nœuds. Dans ce travail, nous calculons la distance entre deux pages v et w basée sur la probabilité de visite des pages k en tant que la racine carrée de la somme des carrés des différences entre les probabilités des deux pages. Nous avons, donc, obtenu une matrice symétrique D définie comme suit :

$$d_{vw} = \sqrt{(\pi_v^* - \pi_w^*)^2} \tag{6.30}$$

Soit la matrice symétrique $S = [s_{vw}]$, tel que : s_{vw} mesure la similarité normalisée entre les pages v et w de l'ensemble V, et $d = \max_{(v,w)\in V^2} d_{vw}$. Nous définissons S comme suit :

$$s_{vw} = 1 - \frac{d_{vw}}{d} \tag{6.31}$$

Pour illustrer notre idée, nous présentons les résultats obtenus dans l'exemple de la figure (Fig.6.1). Nous avons obtenu la matrice de probabilité P et les deux vecteurs de probabilité des liens entrants P_{in} et des liens sortants P_{out} de chaque nœud. Nous présentons aussi le vecteur de probabilité à l'état stable $\pi^{(*)}$ et la matrice de similarité normalisée S (Fig. 6.2).

P

0	¼	¼	½	0
0	0	1	0	0
0	0	0	½	½
0	1	0	0	0
1	0	0	0	0

P_{in}

0.20
0.25
0.30
0.15
0.10

$\pi^{(*)}$

0.14
0.24
0.27
0.21
0.14

S

1.00	0.25	0.00	0.50	1.00
0.25	1.00	0.75	0.75	0.25
0.00	0.75	1.00	0.50	0.00
0.50	0.75	0.50	1.00	0.50
1.00	0.25	0.00	0.50	1.00

P_{out}

0.10	0.20	0.30	0.25	0.15

FIGURE 6.2 – Exemple de la matrice stochastique et symétrique.

La modularité est utilisée comme le critère pour déterminer la meilleure partition qui pourrait être détectée. Nous réécrivons la modularité comme suit :

$$Q = \frac{1}{2 * m} \sum_{v,w} [P_{vw} - (n.\pi_v^*.\pi_w^*)]\delta(C_v, C_w) \tag{6.32}$$

Tel que :

$$\delta(C_v, C_w) = \begin{cases} 1 & \text{si } \{v, w\} \text{ sont dans la même communauté} \\ 0 & \text{sinon.} \end{cases}$$

Nous employons S pour révéler la structure communautaire du réseau en adaptant l'algorithme défini dans [CL02]. Chavent et Lechevallier ont définit un prototype d'un cluster en optimisant un critère d'adéquation classique basé sur la distance de Hausdorff. Nous adaptons leurs méthode pour créer des prototype de communautés en optimisant un critère d'adéquation J qui mesure l'ajustement entre une partition en communautés $P = (C_1, \ldots, C_K)$ et ses prototypes $G = (G_1, \ldots, G_K)$, nous l'écrivons comme suit :

$$J(P, G) = \sum_{k=1}^{K} \sum_{v_i \in C_k} s_{(v_i, G_k)}, \tag{6.33}$$

Tel que $s_{(v_i,G_k)}$ est la similarité entre v_i et le prototype de son communauté G_k.

Notre algorithme CDMMC détecte la meilleure partition en communautés qui résulte d'une valeur de la modularité optimal (voir l'algorithme 1).

Algorithme 9 : Découverte de la meilleur partition en communautés

Données : S
$QMax \leftarrow 0$
$KMax \leftarrow 1$
pour *k=1 to n* **faire**
  Call $[P_k, G_k]$=CDMMC(S, k)
  Compute Q with formule (32)
  si $QMax < Q$ **alors**
    $QMax \leftarrow Q$
    $KMax \leftarrow k$

Notre approche est structuré en trois principales étapes (voir l'algorithme 2).

Algorithme 10 : Méthode de découverte de communautés (CDMMC)

Données : S, k
$t \leftarrow 0$
Select k prototypes from centers : $G^{(0)} = (G_1^{(0)}, \ldots, G_k^{(0)})$
répéter
  Etape 1 : *Build the best partition*
  début
    $t \leftarrow t+1$
    $test \leftarrow 0$
    $P^{(t)} \leftarrow P^{(t-1)}$
    pour *m=1 to k* **faire**
      pour chaque $v_i \in C_m^{(t)}$ **faire**
        find the new winning cluster $C_l^{(t)}$ such that :
        $l = \arg\max_{1 \leq h \leq k} s(v_i, G_h)$
        si $l \neq m$ **alors**
          $test \leftarrow 1$
          $C_l^{(t)} \leftarrow C_l^{(t)} \cup \{v_i\}$
          $C_m^{(t)} \leftarrow C_m^{(t)} \setminus \{v_i\}$
  Etape 2 : *Find the best prototypes*
  begin
    pour *m=1 to k* **faire**
      Compute the prototype $G_m^{(t)} \in V$ of cluster $C_m^{(t)}$ according to :
      $G_m^{(t)} = \arg\max_{G \in C_m^{(t)}} \sum_{v_i \in C_m^{(t)}} s(v_i, G)$
jusqu'à $test = 0$;
return P_k, G_k

Nous appliquons CDMMC sur le graphe présenté dans la figure 6.1 (voir la figure 6.3). L'algorithme identifie correctement trois communautés cohésives. Le dendrogramme illustre les niveaux hiérarchique de partition en communautés.

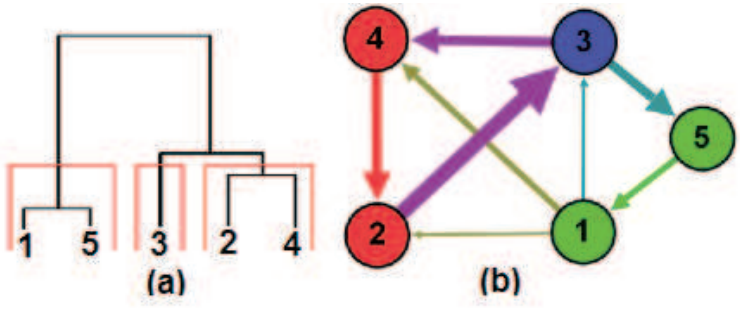


FIGURE 6.3 – (a) Dendrograme - (b) Graphe de partitions

6.5 Les résultats expérimentaux de l'algorithme de découverte de communautés CDMMC

Dans l'étape de la découverte de modèles, nous avons détecter les communautés Web et ainsi le comportement de navigation des utilisateurs qui peut être exploité dans le processus de personnalisation du web. Nous avons, donc, effectué nos expérimentations sur deux systèmes Web :

- La base de données des pages Web que nous avons utilisée décrit les visites de pages des utilisateurs qui ont visité le site Web : msnbc.com, et cela en date du 28 septembre 1999. Les données proviennent des logs IIS (Internet Information Services) pour msnbc.com et des parties relatives aux actualités de msn.com pour toute la journée du 28 septembre 1999. Chaque séquence de l'ensemble de données correspond aux pages consultées par un utilisateur pendant cette période de vingt-quatre heures. Chaque événement de la séquence correspond à une requête de demande d'une page web d'un utilisateur. Les demandes ne sont pas enregistrées dans l'URL mais elles sont plutôt enregistrées dans la catégorie des pages web. Les catégories sont "frontpage", "news", "tech", "local", "opinion", "on-air", "misc", "weather", "health", "living", "business", "sports", "summary", "bbs" (bulletin board service), "travel", "msn-news", and "msn-sports". Comme le nombre d'utilisateurs est 989 818, le nombre moyen de visites par utilisateur est égale à 5,7 [BGLL08]. Après avoir créé le graphe de comportement de navigation, nous avons obtenu un graphe avec 17 noeuds et 289 liens, avec un poids total du graphe égale à 3 708 976.
- Le graphe Web créé est un graphe orienté qui représente les sessions d' utilisateurs du site Web de l'université de Sétif. Il contient 136 nœud et 2 456 liens pour un poids total égale à 25 676 (Voir figure 6.4).

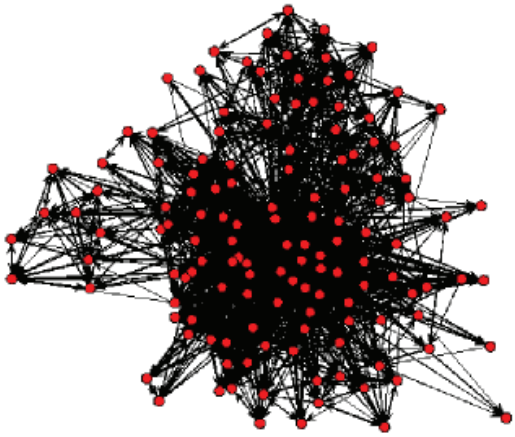


FIGURE 6.4 – Graphe du comportement de navigation

6.5.1 Évaluation de la modularité

Afin de tester la performance de l'algorithme CDMMC, nous analysons la valeur de la modularité et comparons ses résultats avec ceux obtenus avec l'algorithme PAM (Partitioning Around Medoids), l'algorithme de Ward, l'algorithme walktrap et l'algorithme k-means.

6.5.1.1 L'algorithme PAM

Le principe de l'algorithme PAM (Partitioning Around Medoids) de Kaufman et Rousseeuw (1990) est comme suit : Un médoïde (medoid) est le représentant d'une classe, choisie comme son objet le plus central, ce dont on s'assure en permutant systématiquement un représentant et un autre objet de la population choisis au hasard, pour vois si la qualité de la classification croît, c'est à dire si la somme des distances de tous les objets à leurs représentant décroît. L'algorithme s'arrête lorsque plus aucune permutation n'améliore la qualité de la classification. Dans une varaiantes de PAM [RRdlIRS06], les auteurs ont calculé les k objets représentatifs dont la dissimilarité moyenne par rapport à tous les objets du cluster est minimale. Les k objets représentatifs devraient minimiser une fonction objective, qui est la somme des dissimilarités de tous les objets par rapport leurs médoïde la plus proche.

6.5.1.2 L'algorithme de Ward

La méthode de Ward est une méthode de classification hiérarchique agglomérative. Elle consiste à regrouper les classes de façon que l'augmentation de l'inertie interclasse soit maximum. Dans la travail [SR05], Rizzo a proposé une extension de la méthode de Ward basée sur la distance euclidienne. Il a proposé une fonction objective et une distance de cluster en fonction de puissance α de la distance euclidienne dans l'intervalle $(0, 2]$, afin d'obtenir une meilleure capacité à identifier des clusters avec des centres presque égaux.

6.5.1.3 L'algorithme Walktrap

L'algorithme Walktrap est basé sur l'intuition qu'une marche aléatoire partant d'un nœud a plus de probabilité à rester piégée pendant un certain temps dans la communauté du nœud de départ. Pons et al [PL06] ont défini une métrique de distance liée aux approches spectrales qui sont basées sur le fait que deux nœuds appartenant à la même communauté ont des composantes similaires des vecteurs propres principaux. L'algorithme identifie les composantes connectées en calculant pour chaque nœud dans le graphe un vecteur qui donne la probabilité qu'un marcheur aléatoire arrive aux autres nœuds du réseau en k pas de temps. Les vecteurs de probabilité ainsi calculés pour chaque nœud sont utilisés pour calculer des similarités entre les nœuds.

6.5.1.4 L'algorithme k-means

k-means est une méthode de partitionnement de données. Dans la variante des k-means de Mac Queen (1967)un objet peut être affecté à une classe au cours d'une itération puis changer de classe à l'itération suivante, ce qui n'est pas possible avec la classification ascendante hiérarchique pour laquelle une affectation est irréversible. En multipliant les points de départ et les répétitions on peut explorer plusieurs solutions possibles.

Dans cette section, nous comparons la qualité de partitionnement obtenue en appliquons la méthode CDMMC et les autres approches choisies. Nous réalisons nos expérimentations sur le jeu de données du site Web "msnbc.com". La figure 6.5 illustre la variation de la modularité des algorithmes. Nous remarquons que l'algorithme CDMMC produit une modularité maximale est égale à $0,489$. K-means identifie aussi une bonne partition en communauté quand $Q = 0,46$. Par conséquent, ces résultats montrent les capacités de chaque algorithme à identifier la structure des communautés. L'algorithme CDMMC identifie trois communautés dans le graphe du comportement de navigation du site Web "msnbc.com" (voir figure 6.6).

Les résultats de comparaison de la modularité entre les différents algorithmes appliqués sur le graphe du site Web de l'université de Ferhat Abbas Setif sont représentés sur la figure 6.7. En appliquant notre algorithme, la modularité optimale qui résulte d'une partition en communautés pertinente est égale à $Q = 0,247$. L'algorithme de Ward détecte une structure communautaire significative avec une valeur de modularité de $Q = 0,236$. L'algorithme PAM identifie une bonne partition en communautés lorsque $Q = 0,213$. En ce qui concerne l'algorithme Walktrap, la modularité maximale est de $Q = 0,160$. La modularité produit par k-means est $Q = 0,186$. Nous constatons que la modularité produit par l'application de l'algorithme Walktrap est petite car la qualité des résultats est affecté par le choix du longueur de pas. Les résultats de l'algorithme k-means présentent une structure communautaire non pertinente du fait que le calcul du centroïde influe sur le résultat final de la partition. Nous remarquons que l'algorithme PAM maximise la modularité et réduit l'exploration de l'espace

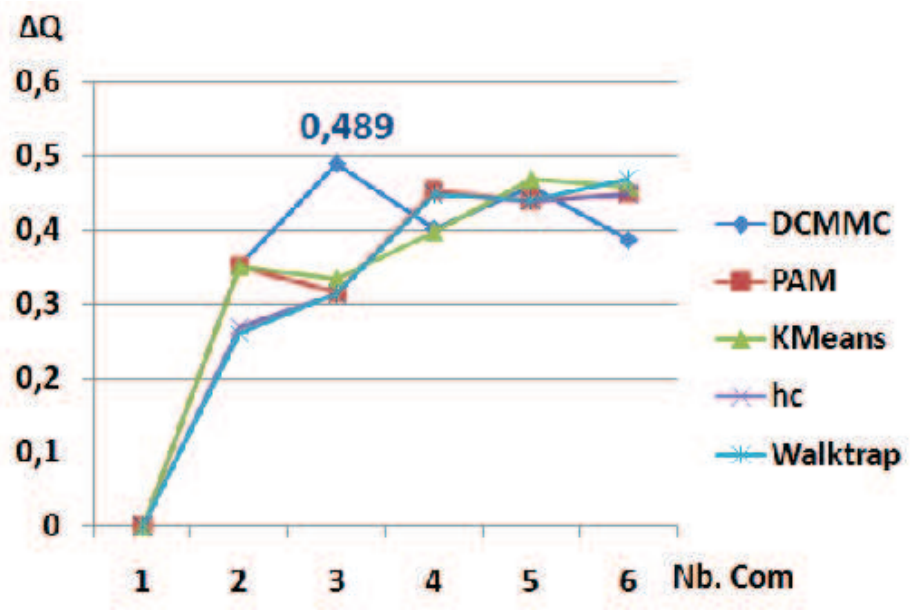


FIGURE 6.5 – Comparaison de la variation de modularité pour les partitions en communautés réalisées sur le graphe du comportement de navigation extrait depuis "msnbc.com"

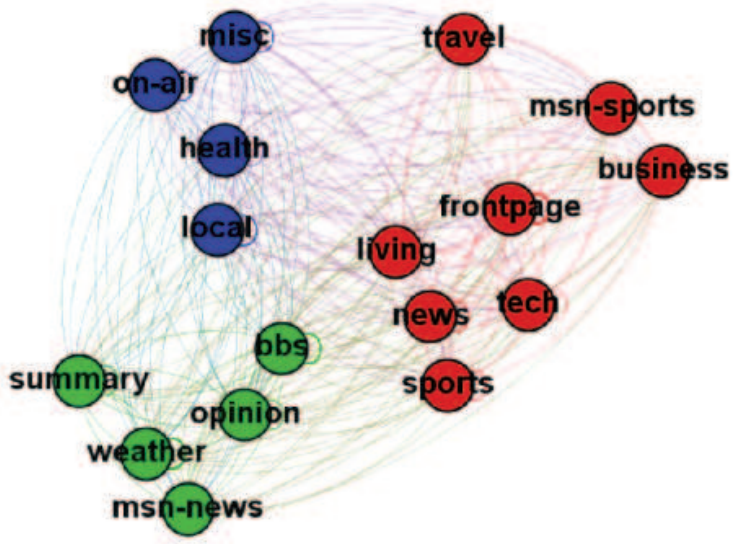


FIGURE 6.6 – Découverte de communautés dans le graphe du comportement de navigation du site Web msnbc.com

de recherche. L'algorithme de Ward révèle aussi une structure communautaire significative. Notre approche CDMMC montre une structure de communautés pertinentes car elle conserve l'information utile contenue dans les séquences de sessions, ce qui permet une meilleure identification de la structure communautaire.

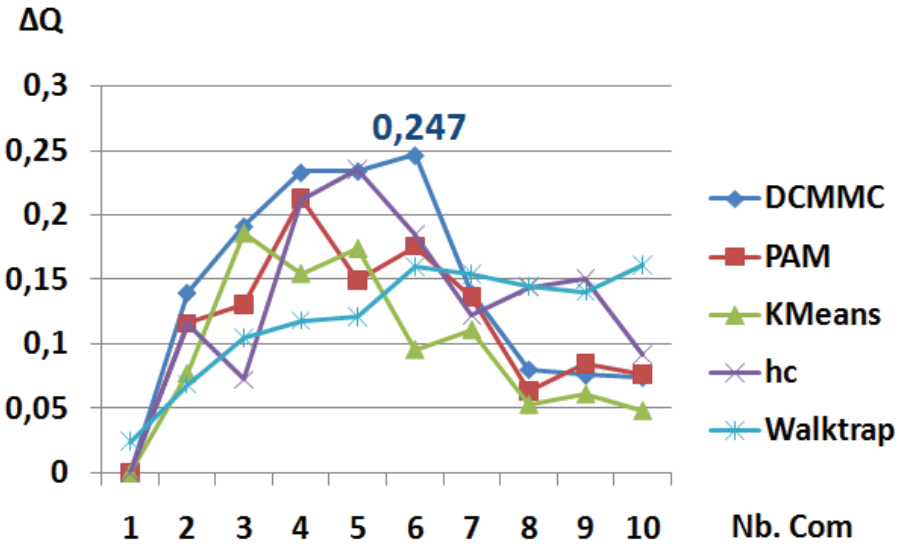


FIGURE 6.7 – Comparaison de la variation de modularité des algorithmes appliqués sur le graphe de comportement de navigation du site web de l'université de Sétif.

6.5.2 Analyse de l'identification de la structure de communautés

Une communauté est généralement considérée comme un ensemble de sommets dans un graphe qui sont fortement reliés entre eux et faiblement relié du reste du graphe [GN02]. Nous avons analysé les communautés détectées en appliquons CDMMC, Ward, PAM, k-means et walktrap. Le tableau 6.2 contient les résultats et présente le total des liens internes et externes pour chaque partition en communautés. Nous constatons que les partitions représentent une bonne structure de communauté lorsque le nombre de liens internes au sein d'une communauté est supérieure au nombre de ses liens externes. D'après les statistiques qui qualifient la qualité d'une division du réseau donnée dans le tableau 6.3, nous constatons que l'algorithme CDMMC maximise la modularité pour la partition en six communautés.

Les résultats présentés dans le tableau 6.4 comparent le nombre de liens des communautés dans la partition de six communautés détectées par CDMMC, Ward, PAM, k-means et Walktrap. Nous comparons le rapport entre la somme des poids des liens intra-communautaires (liens reliant les nœuds dans la même communauté) et la somme des poids des liens inter-communautaires (liens reliant les nœuds des différentes communautés) afin d'illustrer si la communauté présente une forte ou faible propriété de la structure de communauté selon la définition de Radicchi et al [RCC+04].

Radicchi et al [RCC+04] ont défini deux définitions quantitatives de ce qui est une communauté ; *Communauté au sens fort* et *Communauté au sens faible*. En conséquence, un sous-graphe $C \subset G$ serait une communauté au sens fort si chacun de ses nœuds a plus de liens le reliant aux nœuds appartenant à C que ceux le reliant à d'autres nœuds qui n'appartiennent pas à C. De la même façon, $C \subset G$ serait une communauté au sens faible si la somme

TABLE 6.2 – Les résultats de la détection des commuanutés en appliquant les algorithmes : CDMMC, Ward, PAM, k-means et walktrap. Comparaison entre les liens intra-communautaires et les liens inter-communautaires pour les différentes partitions du réseau étudié.

Nb Comm.	CDMMC		PAM		KMeans		HC(Ward)		Walktrap	
	outer	inner	outer	inner	outer	inner	outer	inner	outer	inner
2	15 101	10 575	12 991	12 685	12 991	12 685	12 991	12 685	10 545	5 305
	59%	41%	51%	49%	51%	49%	51%	49%	67%	33%
3	9 388	16 288	8 633	17 043	9 574	16 102	9 694	15 982	1 980	1 945
	37%	63%	34%	66%	37%	63%	38%	62%	50%	50%
4	6 683	18 993	6 705	18 971	6 554	19 122	7 210	18 466	1 966	2 546
	26%	74%	26%	74%	26%	74%	28%	72%	44%	56%
5	5 360	20 316	5 976	19 700	6 216	19 460	6 930	18 746	709	1 208
	21%	79%	23%	77%	24%	76%	27%	73%	37%	63%
6	4 873	20 803	5 651	20 025	5 061	20 615	5240	20 436	3 195	2 101
	19%	81%	22%	78%	20%	80%	20%	80%	60%	40%
7	4 949	20 727	4 999	20 677	5 146	20 530	5 019	20 657	2 006	2007
	19%	81%	19%	81%	20%	80%	20%	80%	50%	50%
8	4 518	21 158	4 359	21 317	4 323	21 353	4 628	21 048	15 854	6 306
	18%	82%	17%	83%	17%	83%	18%	82%	72%	28%
9	4 306	21 370	4 121	21 555	4 266	21 410	3 982	21 694	17 766	6 541
	17%	83%	16%	84%	17%	83%	16%	84%	73%	27%
10	4 215	21 461	3 118	22 558	4 183	21 493	3 420	22 256	13 559	8 741
	16%	84%	12%	88%	16%	84%	13%	87%	61%	39%

TABLE 6.3 – La qualité de la partition en communautés identifiée par les méthodes : CDMMC, Ward, PAM, k-means et walktrap. Notre approche CDMMC maximise la moduolarité pour la partition en 06 commuanutés.

Nb Comm.	CDMMC	PAM	KMeans	HC(Ward)	Walktrap
2	0,140	0,116	0,078	0,116	0,069
3	0,192	0,131	0,186	0,073	0,105
4	0,234	0,213	0,155	0,212	0,118
5	0,235	0,149	**0,174**	**0,236**	0,121
6	**0,247**	**0,176**	0,096	0,185	**0,160**
7	0,140	0,137	0,111	0,123	0,154
8	0,080	0,064	0,053	0,145	0,145
9	0,076	0,085	0,061	0,151	0,140
10	0,074	0,077	0,049	0,092	0,161

du nombre de liens qui relient les nœuds à l'intérieur de C est plus grande que la somme de tous les liens qui relient les nœuds appartenant à C avec des nœuds qui n'appartiennent pas à C.

Par exemple, prenons la quatrième communauté (voir tableau 6.4), en appliquant CDMMC, le poids des liens internes est égal à 84% alors que le poids des liens externes est égal à 16%. Lorsque nous appliquons PAM, le poids des liens internes est égal à 74% alors que le poids des liens externes est égal à 26%. L'algorithme k-means donne également des résultats signi-

TABLE 6.4 – Analyse des liens des communautés dans la partition de 6 communautés détectées par CDMMC, Ward, PAM, k-means et walktrap.

#Comm.	CDMMC		PAM		KMeans		HC(Ward)		Walktrap	
	outer	inner	outer	inner	outer	inner	outer	inner	outer	inner
1	8 345	0	1 415	2721	731	1 798	1 616	2 369	1335	242
	100%	0%	34%	66%	29%	71%	41%	59%	85%	15%
2	1 230	2 127	2 541	3138	2 141	3 039	2 164	3 391	669	343
	37%	63%	45%	55%	41%	59%	39%	61%	66%	34%
3	218	472	8 345	0	853	2 630	128	1 899	264	64
	32%	68%	100%	0%	24%	76%	6%	94%	80%	20%
4	454	2 405	359	1 018	1 336	2 748	8 345	0	20 094	1 559
	16%	84%	26%	74%	33%	67%	100%	0%	93%	7%
5	1 359	3 302	1 336	2 748	8 345	0	1 332	2 377	781	227
	29%	71%	33%	67%	100%	0%	36%	64%	77%	23%
6	1 612	4 152	2 055	0	2 055	0	2 055	0	74	24
	28%	72%	100%	0%	100%	0%	100%	0%	76%	24%
Total	25676									

ficatifs. Cela signifie que ces algorithmes découvre une structure de communauté cohérente. Cependant, pour cet exemple, l'algorithme Walktrap identifie une structure de communauté non cohérente.

TABLE 6.5 – La taille des communautés détectées par l'algorithme CDMMC et les autres algorithmes pour le site Web de l'UFAS.

#Comm.	CDMMC	PAM	KMeans	HC(Ward)	Walktrap
1	1	54	58	80	8
2	52	29	24	25	8
3	29	1	1	22	4
4	27	43	8	1	100
5	19	8	1	7	14
6	8	1	1	1	2
Total	136				

Le tableau 6.5 présente la taille es communautés détectées par l'algorithme CDMMC et les autres algorithmes pour le site Web de l'UFAS. L'algorithme CDMMC a détecté un cluster composé de nœud unique qui représente la page de départ, celle-là est classée dans la quatrième communauté. L'algorithme Ward a également détecté cette page racine qui correspond à la page d'accueil du site, elle est ainsi détectée dans la troisième communauté. Ce résultat est obtenu parce que l'algorithme de Ward crée une structure hiérarchique qui reflète l'ordre dans lequel les groupes sont fusionnés. L'algorithme CDMMC détecte une communauté qui contient les utilisateurs qui s'intéressent aux manifestations scientifiques (celle contenant 19 nœuds), cette communauté est à la sixième communauté identifiée en utilisant la méthode PAM. En fait, la taille des communautés dépend de la façon de partitionnement de chaque méthode. Par exemple, la méthode PAM regroupe les visiteurs consultant les manifestations scientifiques, les activités scientifiques et les ressources électroniques dans une

communauté de 29 nœuds. De ce fait, nous constatons qu'il est plus significatif de faire une comparaison selon la partition optimale de chaque algorithme. Le tableau 6.3 montre que l'algorithme PAM a une meilleure division avec trois communautés, k-means a une modularité optimale avec deux partitions, et la méthode de Ward détecte cinq communautés.

La figure (6.8) illustre la partition en commuantés découverte par notre approche. Le graphe contient six communautés identifiant des groupes d'utilisateurs ayant un comportement similaire. Ce pattern est très utile pour créer une version personnalisée et améliorée du site Web de l'Université de Ferhat Abbas (Algérie).

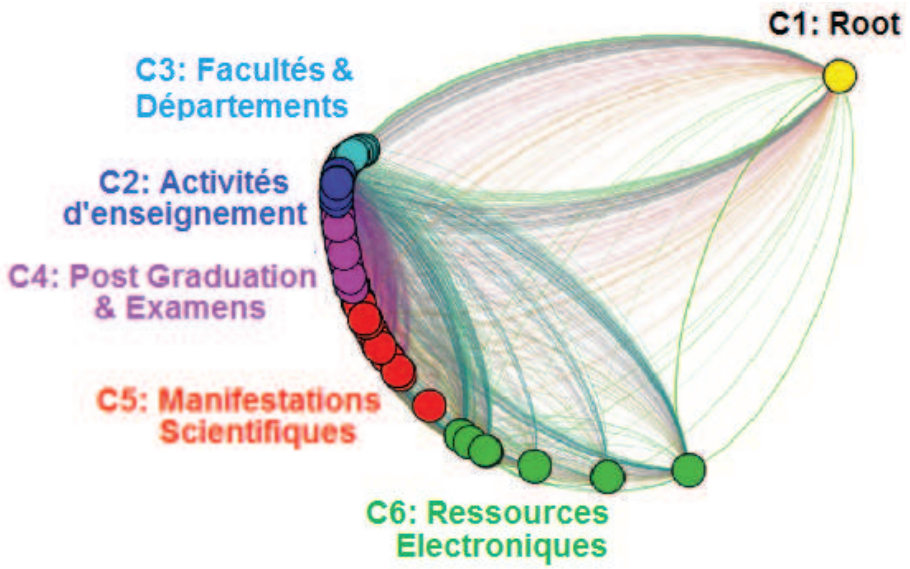


FIGURE 6.8 – La structure des communautés détectée par la méthode CDMMC.

Nous décrivons les communautés découvertes :

- Le noeud jaune illustre la communauté de la page Web racine.
- La deuxième communauté présente les utilisateurs qui accèdent aux activités d'enseignement (noeuds bleu).
- Les nœuds bleu clair représentent la troisième communauté qui regroupe les accès aux pages des différentes facultés et départements. (29 nœuds).
- La quatrième communauté illustre les visites des pages web de la post-graduation et accède aux pages du concours. Les nœuds mauves appartiennent à cette communauté.
- La cinquième communauté identifie les accès aux pages des manifestations scientifiques (la communauté contient 19 nœuds rouges).
- Une communauté identifie les visiteurs consultant les activités scientifiques de l'Université Ferhat Abbas et les pages de ressources électroniques. La figure (6.8) montre qu'elle contient 8 nœuds verts.

6.6 Conclusion

En résumé, notre travail consiste en l'amélioration des tâches d'extraction du processus WUM pour extraire le comportement des utilisateurs lors de leurs accès aux sites Web. Ce qui aide à améliorer le design du système Web. A cet effet, nous avons proposé une technique efficace dans la phase de découverte de patterns dans laquelle nous modélisons les séquences de navigation des utilisateurs par un processus stochastique qui converge vers une distribution de probabilité stable. Notre approche maximise la modularité et résulte d'une partition sous-optimale. Le modèle d'accès obtenus montrent une description significative des communautés découverte permettant ainsi une meilleure personnalisation du système Web.

Chapitre 7

Conclusion Générale

Le nombre d'accès aux pages Web ne cesse de croître. Le Web est devenu l'une des plates-formes les plus répandues pour la diffusion et la recherche d'information. Par conséquence, beaucoup d'opérateurs de sites Web sont incités à analyser l'usage de leurs sites afin d'améliorer leur réponse vis-à-vis des attentes des internautes. Or, la manière dont un site Web est visité peut changer en fonction de divers facteurs. Les modèles d'usage doivent ainsi être mis à jour continuellement afin de refléter fidèlement le comportement des visiteurs. Dans ce travail de thèse, nous avons proposé une méthodologie pour le processus du Web Usage Mining. Dans la phase de prétraitement, nous avons appliqué dans le module de nettoyage un algorithme heurestique définit dans [TK04] afin de maintenir seulement les données d'usagers qui peuvent être effectivement exploitées pour identifier le comportement de navigation des utilisateurs. Ensuite, la structuration des requêtes en un ensemble de sessions des utilisateurs est effectuée dans le module de structuration suivie par une étape de filtrage des données. Après l'achèvement de l'étape du prétraitement, l'algorithme Aprioi a été utilisé pour extraire des règles d'associations entre les sessions des utilisateurs et identifier leurs comportements. Bien que la méthode des règles d'association permette de découvrir les fréquence d'accès à une ressource Web, elle n'est pas optimale pour analyser le comportements d'utilisateurs d'un site Web. A cet effet, nous avons employé des approches de découverte de communautés dans la phase d'extraction du modèle pour produire une description très significatives des comportements des utilisateurs d'un système Web.

Dans un premier temps, nous avons adapté l'algorithme fast [NG04] de détection de communautés à nos besoin d'analyse pour modéliser le graphe d'une communautés Web et identifier la corrélation entre les utilisateurs et les différentes ressources Web. Cet aspect de communautés du Web , qui s'appuient sur les profils d'utilisateurs similaires, évalue leurs activités d'exploration par des poids représentant l'occurrence des séquences des pages Web visitées au cours des sessions des utilisateurs. Le modèle d'accès au Web que nous avons obtenu identifie les communautés des utilisateurs selon leurs intérêts d'accès au ressources du Web.

Dans un second temps, nous avons conçu un algorithme agglomératif pour pallier à la non pertinence des informations dû à l'absence de l'ordre des pages vues dans une transaction. Notre méthode analyse chaque transaction d'utilisateur comme une probabilité de transitions et définit un coefficient de pertinence pour la quantification de comportements de l'utilisateur. Le modèle d'accès au Web que nous avons obtenu permet de découvrir des communautés Web et améliore l'efficacité des tâches d'extraction du processus WUM. Notre approche est totalement indépendante des données d'usage du Web et peut être appliquée sur différents types de données.

Guidés par les deux approches précédentes, nous avons proposé une nouvelle approche basée sur les chaînes de Markov pour identifier le comportement des utilisateurs d'un système Web. Dans cette troisième contribution, notre motivation est de prendre en compte la situation présente d'accès aux ressources Web par un utilisateur pour prévoir ses comportements futurs. De ce fait, nous avons modélisé les séquences de navigation des utilisateurs par un processus stochastique et nous avons calculé davantage la similarité entre les différentes sessions de navigation des utilisateurs. De plus, nous avons définit une nouvelle mesure de la fonction de qualité considérant que l'état stable corresponde à des partitions optimales du graphe étudié. En conséquence, l'algorithme de découverte de communautés que nous avons définit DCMMC (Detection Community Method based on Markov Chain) induit l'extraction d'un modèle servant à mieux comprendre le comportement des visiteurs du site Web, et ainsi offrir de nouvelles connaissances utiles.

Enfin, nous décrivons quelques pistes pouvant inspirer des futurs travaux dans la continuation de ceux présentés ici.

- La fouille des données distribuées gagne en popularité car elle implique l'extraction d'une grande quantité d'informations stockées dans différents sites de l'entreprise ou dans différentes organisations. Récemment, plusieurs algorithmes sont développés pour extraire des modèles à partir de différents systèmes et fournir des connaissances appropriées. De ce fait, nous envisageons à proposer une extension de notre méthode DCMMC pour qu'elle puisse s'exécuter sur des systèmes distribués (exemple : des machines GPU).
- Les systèmes de recommandations regroupent tous les systèmes capables de fournir des recommandations adaptées aux goûts, aux besoins ou aux moyens des utilisateurs, et cela afin de les aider à accéder à des ressources utiles ou intéressantes au sein d'un espace de données important. Nous pouvons donc envisager de compléter l'étude expérimentale de notre approche de détection de communautés basé sur les chaînes de Markov par son implémentation dans un système de recommandation.
- De nos jours, nous pouvons penser que le processus de WUM deviendra aussi omniprésent que certaines des technologies les plus utilisées aujourd'hui. Ce type de fouille de données peut révéler diverses aspects topologiques et comportementaux. Il consiste

à extraire des données à partir d'appareils mobiles pour obtenir des informations des utilisateurs depuis différents systèmes (web blog, sites web, média sociaux, systèmes de localisations, GPS..ect). En dépit d'avoir des défis dans ce type de fouille de données tels que les difficulté de prévoir les futurs positions géographiques, risque d'atteinte à la vie privés, coût, etc, il serait donc très intéressant de se focaliser sur ce volet de recherche et développer des méthodes performants pour l'extraction des modèles de mobilité humaine.

Bibliographie

[AB02] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1) :47, 2002.

[ACJ+03] David Auber, Yves Chiricota, Fabien Jourdan, Guy Melançon, and others. Multiscale Visualization of Small World Networks. In *InfoVis03*, volume 3, pages 75–81. IEEE Computer Society, 2003.

[ADFG07] Alex Arenas, Jordi Duch, Alberto Fernández, and Sergio Gómez. Size reduction of complex networks preserving modularity. *New Journal of Physics*, 9(6) :176, 2007.

[ADGPV06] Alex Arenas, Albert Diaz-Guilera, and Conrad J Pérez-Vicente. Synchronization reveals topological scales in complex networks. *Physical review letters*, 96(11) :114102, 2006.

[AFFG08] A Arenas, A Fernandez, S Fortunato, and S Gomez. Motif-based communities in complex networks. *Journal of Physics A : Mathematical and Theoretical*, 41(22) :224001, June 2008.

[AJAL99] Réka Albert, Hawoong Jeong, and Barabási Albert-László. Internet : Diameter of the world-wide web. *Nature*, 401(6749) :130–131, 1999.

[AJAL00] Réka Albert, Hawoong Jeong, and Barabási Albert-László. Attack and error tolerance of complex networks. *Nature*, 406(6794) :378–382, 2000.

[AK05] Bruno Agard and Andrew Kusiak. Exploration des bases de données industrielles à l'aide du datamining–Perspectives. In *9ème colloque national AIP PRIMECA*, pages 1–9, 2005.

[AL02] Barbási Albert-Laszlo. *Linked : the new science of networks*. Perseus, 2002.

[AMO94] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. Network flows : Theory, algorithms, and applications. *Journal of the Operational Research Society*, 45(11) :1340–1340, 1994.

[ANTT02] Arvind Arasu, Jasmine Novak, Andrew Tomkins, and John Tomlin. Pagerank computation and the structure of the web : Experiments and algorithms. In *Proceedings of the Eleventh International World Wide Web Conference, Poster Track*, pages 107—-117, 2002.

[AS94] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, volume 1215, pages 487—-499, 1994.

[ASBS00] Luıs A. Nunes Amaral, Antonio Scala, Marc Barthelemy, and H. Eugene Stanley. Classes of small-world networks. *Proceedings of the national academy of sciences*, 97(21) :11149–11152, 2000.

[BA99] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439) :509–512, 1999.

[BB05] James P. Bagrow and Erik M. Bollt. Local method for detecting communities. *Physical Review E*, 72(4) :046108, 2005.

[BGLL08] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics : theory and experiment*, 2008(10) :10008, 2008.

[BIL+07a] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda. Detecting complex network modularity by dynamical clustering. *Physical Review E*, 75(4) :045102, 2007.

[BIL+07b] S Boccaletti, M Ivanchenko, V Latora, A Pluchino, and A Rapisarda. Detecting complex network modularity by dynamical clustering. *Physical Review E*, 75(4) :045102, 2007.

[BK73] Coenraad Bron and Joep Kerbosch. Finding all cliques of an undirected graph (algorithm 457). *Commun. ACM*, 16(9) :575–576, 1973.

[BLM+06] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D.-U. Hwang. Complex networks : Structure and dynamics. *Physics reports*, 424(4) :175–308, 2006.

[BLO+06] Ranieri Baraglia, Claudio Lucchese, Salvatore Orlando, Massimo Serrano', and Fabrizio Silvestri. A privacy preserving web recommender system. In *Proceedings of the 2006 ACM symposium on Applied computing*, SAC '06, pages 559—-563, New York, NY, USA, 2006. ACM.

[Bol98] Béla Bollobás. Random graphs. In *Modern Graph Theory*, pages 215–252. Springer, 1998.

[Bon08] Ugo Bontognali. *AWI, Applicazione Web Intelligente*. SUPSI, 2008.

[BP01] Sven Bilke and Carsten Peterson. Topological properties of citation and metabolic networks. *Physical Review E*, 64(3) :036106, 2001.

[Bro99] Annie Brooking. *Corporate memory : Strategies for knowledge management*. Cengage Learning EMEA, 1999.

[BS02] Vincent D. Blondel and Pierre P. Senellart. Automatic extraction of synonyms in a dictionary. In *the SIAM Workshop on Text Mining*, volume 1. Vertex, 2002.

[Buc03] Mark Buchanan. *Nexus : small worlds and the groundbreaking theory of networks*. WW Norton & Company, 2003.

[Bur76] Ronald S. Burt. Positions in networks. *Social forces*, 55(1) :93–122, 1976.

[BWD96] Marcelo Blatt, Shai Wiseman, and Eytan Domany. Super paramagnetic clustering of data. *Physical review letters*, 76(18) :3251, 1996.

[CCB02] Karine Chevalier, Vincent Corruble, and Cécile Bothorel. Surfminer : Connaître les utilisateurs d'un site. In *Documents Virtuels Personnalisables (DVP 2002)*, pages 85–96, 2002.

[Cha00] Soumen Chakrabarti. Data mining for hypertext : A tutorial survey. *ACM SIGKDD Explorations Newsletter*, 1(2) :1–11, 2000.

[CL02] Marie Chavent and Yves Lechevallier. Dynamical clustering of interval data : optimization of an adequacy criterion based on hausdorff distance. In *Classification, clustering, and data analysis*, pages 53–60. Springer, 2002.

[CLRS01] Thomas H. Cormen, Charles Eric Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*, volume 6. MIT press Cambridge, 2001.

[CLSA08] Malika Charrad, Yves Lechevallier, Gilbert Saporta, and Mohamed Ben Ahmed. Web content data mining : la classification croisée pour l'analyse textuelle d'un site web. In *EGC*, pages 43–54, 2008.

[CMP+98] Mark Craven, Andrew McCallum, Dan PiPasquo, Tom Mitchell, and Dayne Freitag. Learning to extract symbolic knowledge from the world wide web. Technical report, Carnegie-mellon univ pittsburgh pa school of computer Science, 1998.

[CNM04] Aaron Clauset, Mark E.J Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6) :066111, 2004.

[Coo00] Robert Walker Cooley. *Web usage mining : discovery and application of interesting patterns from web data*. PhD thesis, University of Minnesota, 2000.

[CS00] Robert Walker Cooley and Jaideep Srivastava. *Web usage mining : discovery and application of interesting patterns from web data*. Citeseer, 2000.

[DA05] Jordi Duch and Alex Arenas. Community detection in complex networks using extremal optimization. *Physical review E*, 72(2) :027104, 2005.

[DBS13] Ahlem Drif, Abdallah Boukerram, and Yacine Slimani. Découverte d'une structure de communauté des usagers du web. In *4ème conférence sur les modèles et l'analyse des réseaux : approches mathématiques et informatiques, MARAMI'13,Saint-Etienne, France*, 2013.

[DBS14] Ahlem Drif, Abdallah Boukerram, and Yacine Slimani. Community Discovery Topology Construction for Ad Hoc Networks. In *International Wireless Internet Conference*, volume 146, pages 197–208, Lisbon (Portugal), 2014. Springer.

[DBSG17] Ahlem Drif, Abdellah Boukerram, Yacine Slimani, and Silvia Giordano. Discovering interest based mobile communities. *Mobile Networks and Applications*, 22(2) :344–355, 2017.

[DBSM16] Ahlem Drif, Abdallah Boukerram, Yacine Slimani, and Abdelouaheb Moussaoui. Découverte de communautés dans les réseaux complexes. *hal.archives-ouvertes.fr*, 2016.

[DDGDA05] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics : Theory and Experiment*, 2005(09) :09008, 2005.

[DM02] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of Networks : from Biological Nets to the Internet and WWW. 2003. *Oxford University Press*, 2002.

[DM04] Luca Donetti and Miguel A Munoz. Detecting network communities : a new systematic and efficient algorithm. *Journal of Statistical Mechanics : Theory and Experiment*, 2004(10) :10012, 2004.

[DM05] Luca Donetti and Miguel A. Muñoz. Improved spectral algorithm for the detection of network communities. In *Modeling Cooperative Behavior in the Social Sciences*, volume 779, pages 104–107, 2005.

[DPV05] Imre Derényi, Gergely Palla, and Tamás Vicsek. Clique percolation in random networks. *Physical review letters*, 94(16) :160–202, 2005.

[DR00] Peter Sheridan Dodds and Daniel H. Rothman. Geometry of river networks. *Physical Review E*, 63(1) :016115–016117, 2000.

[ECB+96] Jean-Louis Ermine, Mathias Chaillot, Philippe Bigeon, Boris Charreton, and D.-MKSM Malavieille. Méthode pour la gestion des connaissances. *Ingénierie des systèmes d'information, AFCET-Hermès*, 4(4) :541–575, 1996.

[EM02a] Jean-Pierre Eckmann and Elisha Moses. Curvature of co-links uncovers hidden thematic layers in the world wide web. *Proceedings of the national academy of sciences*, 99(9) :5825–5829, 2002.

[EM02b] Jean-Pierre Eckmann and Elisha Moses. Curvature of co-links uncovers hidden thematic layers in the world wide web. *Proceedings of the national academy of sciences*, 99(9) :5825–5829, 2002.

[EMB02] Holger Ebel, Lutz-Ingo Mielsch, and Stefan Bornholdt. Scale-free topology of e-mail networks. *Physical review E*, 66(3) :035103, 2002.

[ER59] Paul Erdös and Alfréd Rényi. On random graphs. *Publicationes Mathematicae (Debrecen)*, 6 :290–297, 1959.

[ER90] Leo Egghe and Ronald Rousseau. Introduction to informetrics : Quantitative methods in library, documentation and information science. *Elsevier*, 1990.

[ESMS03] Kasper Astrup Eriksen, Ingve Simonsen, Sergei Maslov, and Kim Sneppen. Modularity and extreme edges of the internet. *Physical review letters*, 90(14) :148701, 2003.

[FA86] Yaotian Fu and Philip W. Anderson. Application of statistical mechanics to NP-complete problems in combinatorial optimisation. *Journal of Physics A : Mathematical and General*, 19(9) :1605, 1986.

[FB07] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1) :36–41, 2007.

[FbPV07] Illés Farkas, Dániel Ábel, Gergely Palla, and Tamás Vicsek. Weighted network modules. *New Journal of Physics*, 9(6) :180, 2007.

[FC08] S. Fortunato and C. Castellano. *Community structure in graphs. Encyclopedia of Complexity and System Science*. Springer, 2008.

[FDPP06] Peter Fine, Ezequiel Di Paolo, and Andrew Philippides. Spatially constrained networks and the evolution of modular control systems. In *International Conference on Simulation of Adaptive Behavior*, pages 546–557. Springer, 2006.

[Fel08a] William Feller. *An Introduction to Probability : Theory and Its Application*, volume 1. Wiley India Pvt. Limited, 3rd edition, August 2008.

[Fel08b] Willliam Feller. *An Introduction to Probability : Theory and Its Application*, volume 2. Wiley India Pvt. Limited, 2nd edition, August 2008.

[FFF99] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM computer communication review*, volume 29, pages 251–262. ACM, 1999.

[FL05] Federico Michele Facca and Pier Luca Lanzi. Mining interesting knowledge from weblogs : a survey. *Data and Knowledge Engineering*, 53(3) :225—-241, 2005.

[FLGC02] Gary William Flake, Steve Lawrence, C. Lee Giles, and Frans M. Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35(3) :66–70, 2002.

[FLM04] Santo Fortunato, Vito Latora, and Massimo Marchiori. Method to find community structures based on information centrality. *Physical review E*, 70(5) :056104, 2004.

[FPSS96] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. Knowledge Discovery and Data Mining : Towards a Unifying Framework. In *KDD-96*, volume 96, pages 82–88, 1996.

[Fre77] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[GA05] Roger Guimera and Luis A. Nunes Amaral. Functional cartography of complex metabolic networks. *nature*, 433(7028) :895, 2005.

[GDDG+03] Roger Guimera, Leon Danon, Albert Diaz-Guilera, Francesc Giralt, and Alex Arenas. Self-similar community structure in a network of human interactions. *Physical review E*, 68(6) :065103, 2003.

[GHL06] Mika Gustafsson, Michael Hörnquist, and Anna Lombardi. Comparison and validation of community structures in complex networks. *Physica A : Statistical Mechanics and its Applications*, 367 :559–576, 2006.

[GLLB04] Jean-Loup Guillaume, Matthieu Latapy, and Stevens Le-Blond. Statistical analysis of a p2p query graph based on degrees and their time-evolution. In *International Workshop on Distributed Computing*, pages 126–137. Springer, 2004.

[GN02] Michelle Girvan and Mark E.J Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America,*, 99(12) :7821–7826, 2002.

[Gra73] Mark S. Granovetter. The strength of weak ties. *American journal of sociology*, pages 1360–1380, 1973.

[GSPA04a] Roger Guimera, Marta Sales-Pardo, and Luís A Nunes Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2) :025101, 2004.

[GSPA04b] Roger Guimera, Marta Sales-Pardo, and Luís A. Nunes Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2) :025101, 2004.

[GSPA07] Roger Guimerà, Marta Sales-Pardo, and Luís A. Nunes Amaral. Module identification in bipartite and directed networks. *Physical Review E*, 76(3) :036102, 2007.

[HHJ03] Petter Holme, Mikael Huss, and Hawoong Jeong. Subnetwork hierarchies of biochemical pathways. *Bioinformatics*, 19(4) :532–538, 2003.

[HKP00] Jiawei Han, Micheline Kamber, and Jian Pei. Data mining : concepts and techniques. *the Morgan Kaufmann Series in data management systems*, 2000.

[Jam98] Michel Jambu. *INTRODUCTION AU DATA MINING. Analyse intelligente des données.* Eyrolles edition, 1998.

[JCY13] Caiyan Jia, Matthew B Carson, and Jian Yu. A fast weak motif-finding algorithm based on community detection in graphs. *BMC Bioinformatics*, 14(1) :227, 2013.

[JD88] Anil K. Jain and Richard C. Dubes. *Algorithms for clustering data.* Prentice-Hall, Inc., 1988.

[JDY09] Jeffrey Q. Jiang, Andreas WM Dress, and Genke Yang. A spectral clustering-based framework for detecting community structures in complex networks. *Applied Mathematics Letters*, 22(9) :1479–1482, 2009.

[JT98] Michel Jaczynski and Brigitte Trousse. WWW assisted browsing by reusing past navigations of a group of users. In *Advances in Case-Based Reasoning*, pages 160—-171. Springer, 1998.

[JTA+00] Hawoong Jeong, Bálint Tombor, Réka Albert, Zoltan N. Oltvai, and Barabási Albert-László. The large-scale organization of metabolic networks. *Nature*, 407(6804) :651–654, 2000.

[Jul11] Andreea Maria Julea. *Extraction de motifs spatio-temporels dans des séries d'images de télédétection : application à des données optiques et radar*. PhD Thesis, Université de Grenoble, 2011.

[Kai99] Jocelyn Kaiser. It'sa small Web after all. *Science*, 285(1815), 1999.

[KF69] PW Kasteleyn and CM Fortuin. Physica (utrecht) 57, 536 (1972); pw kasteleyn and cm fortuin. *Physical Society of Japan Journal Supplement*, 26(11), 1969.

[KFM+03] Ann E. Krause, Kenneth A. Frank, Doran M. Mason, Robert E. Ulanowicz, and William W. Taylor. Compartments revealed in food-web structure. *Nature*, 426(6964) :282–285, 2003.

[KHMG03] Sepandar Kamvar, Taher Haveliwala, Christopher Manning, and Gene Golub. Exploiting the block structure of the web for computing pagerank. *Stanford University Technical Report*, 2003.

[Kle00] Jon Kleinberg. The small-world phenomenon : An algorithmic perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 163–170. ACM, 2000.

[KSM03] V. K. Kalapala, V. Sanwalani, and C. Moore. The structure of the United States road network. *Preprint, University of New Mexico*, 2003.

[KV09] Vassiliki A Koutsonikola and Athena I Vakali. A fuzzy bi-clustering approach to correlate web users and pages. *International Journal of Knowledge and Web Intelligence*, 1(1-2) :3–23, 2009.

[Lar14] Daniel T. Larose. *Discovering knowledge in data : an introduction to data mining*. John Wiley & Sons, 2014.

[LB11] Gordon S. Linoff and Michael JA Berry. *Data mining techniques : for marketing, sales, and customer relationship management*. John Wiley & Sons, 2011.

[LGH05] Pedro G. Lind, Marta C. González, and Hans J. Herrmann. Cycles and clustering in bipartite networks. *Physical review E*, 72(5) :056127, 2005.

[LHY05] Xiangwei Liu, Pilian He, and Qian Yang. Mining user access patterns based on web logs. In *Electrical and Computer Engineering, 2005. Canadian Conference on*, pages 2280–2283. IEEE, 2005.

[LM02] Vito Latora and Massimo Marchiori. Is the Boston subway a small-world network ? *Physica A : Statistical Mechanics and its Applications*, 314(1) :109–113, 2002.

[LMN11] Bing Liu, Bamshad Mobasher, and Olfa Nasraoui. Web usage mining. In *Web Usage Mining*, Data-Centric Systems and Applications, pages 527–603. Springer Berlin Heidelberg, 2011.

[LN04] David Lusseau and Mark E.J Newman. Identifying the role that animals play in their social networks. *Proceedings of the Royal Society of London. Series B : Biological Sciences*, 271(Suppl 6) :S477–S481, 2004.

[LN08] Elizabeth A. Leicht and Mark E.J Newman. Community structure in directed networks. *Physical review letters*, 100(11) :118703, 2008.

[LP49] R. Duncan Luce and Albert D. Perry. A method of matrix analysis of group structure. *Psychometrika*, 14(2) :95–116, 1949.

[LSH08] Sune Lehmann, Martin Schwartz, and Lars Kai Hansen. Biclique communities. *Physical Review E*, 78(1) :016108, 2008.

[LWFD07] Menghui Li, Jinshan Wu, Ying Fan, and Zengru Di. Econophysicists Collaboration Networks : Empirical Studies and Evolutionary Model. In *Econophysics of Markets and Business Networks*, pages 173–182. Springer, 2007.

[LZG14] Hongwei Lu, Qian Zhao, and Zaobin Gan. A Community Detection Algorithm Based on the Similarity Sequence. In *Web Information Systems Engineering – WISE 2014*, volume 8786, pages 63–78. Springer International Publishing, Cham, 2014.

[Mar91] Neo D. Martinez. Artifacts or attributes ? Effects of resolution on the Little Rock Lake food web. *Ecological Monographs*, 61(4) :367–392, 1991.

[MBNL99] Sanjay Kumar Madria, Sourav S Bhowmick, W-K Ng, and Ee-Peng Lim. Research issues in web data mining. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 303–312. Springer, 1999.

[Mil67] Stanley Milgram. The small world problem. *Psychology today*, 2(1) :60–67, 1967.

[MKNG06] Alireza Mahdian, Hamid Khalili, Ehsan Nourbakhsh, and Mohammad Ghodsi. Web graph compression by edge elimination. In *Data Compression Conference (DCC'06)*, pages 1–pp. IEEE, 2006.

[MM09] Frank McSherry and Ilya Mironov. Differentially private recommender systems : building privacy into the net. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 627—-636, 2009.

[MMS09] Noureddine Mekroud, Abdelouahab Moussaoui, and Yacine Slimani. Intégration des techniques du Datamining et des bases de données avancées dans le processus de gestion des connaissances : proposition d'un processus hybride basé sur le raisonnement à partir de cas. In *Conférence Internationale des technologies de l'information et de la Communication*, Sétif (Algérie), 2009.

[MNLM06] Bamshad Mobasher, Olfa Nasraoui, Bing Liu, and Brij Masand, editors. *Advances in Web Mining and Web Usage Analysis : 6th International Workshop on Knowledge Discovery on the Web, WEBKDD 2004, Seattle, WA, USA, August 2004 / Lecture Notes in Artificial Intelligence)*. Springer, 2006 edition, November 2006.

[Mob04] Bamshad Mobasher. Web usage mining and personalization., 2004.

[Moo01] James Moody. Race, school integration, and friendship segregation in america1. *American Journal of Sociology*, 107(3) :679—-716, 2001.

[MRAALAN07] Sales-Pardo Marta, Guimer Roger, Moreira Andr A., and Amaral Lus A. Nunes. Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences*, 104(39) :15224–15229, 2007.

[MRC05] Stefanie Muff, Francesco Rao, and Amedeo Caflisch. Local modularity measure for network clusterizations. *Physical Review E*, 72(5) :056107, 2005.

[MS02] Sergei Maslov and Kim Sneppen. Specificity and stability in topology of protein networks. *Science*, 296(5569) :910–913, 2002.

[MSZ04] Sergei Maslov, Kim Sneppen, and Alexei Zaliznyak. Pattern detection in complex networks : Correlation profile of the Internet. *Physica A*, 333 :529–540, 2004.

[Nas05] Olfa Nasraoui. World wide web personalization. *Encyclopedia of Data Mining and Data Warehousing, Idea Group*, 2005.

[New01] Mark E.J Newman. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical review E*, 64(1) :016132, 2001.

[New03a] Mark E.J Newman. Mixing patterns in networks. *Physical Review E*, 67(2) :026126, 2003.

[New03b] Mark E.J Newman. The structure and function of complex networks. *SIAM review*, 45(2) :167–256, 2003.

[New04a] Mark E.J Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6) :066133, 2004.

[New04b] M.E.J. Newman. Analysis of weighted networks. *Physical Review E*, 70(5) :056131, 2004.

[New06] Mark E.J Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23) :8577–8582, 2006.

[New09] Mark E.J Newman. *Networks : an introduction*. OUP Oxford, 2009.

[NG04] Mark E.J Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2) :026113, 2004.

[NL07] Mark E.J Newman and Elizabeth A. Leicht. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences*, 104(23) :9564–9569, 2007.

[NMCM09] Vincenzo Nicosia, Giuseppe Mangioni, Vincenza Carchiolo, and Michele Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics : Theory and Experiment*, 2009(03) :P03024, 2009.

[PDFV05] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043) :814–818, 2005.

[PFP+07] Gergely Palla, Illes J. Farkas, Peter Pollner, Imre Derenyi, and Tamás Vicsek. Directed network modules. *New journal of physics*, 9(6) :186, 2007.

[Pim79] Stuart L. Pimm. The structure of food webs. *Theoretical population biology*, 16(2) :144–158, 1979.

[PL06] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *J. Graph Algorithms Appl.*, 10(2) :191–218, 2006.

[PMV87] Giorgi Parisi, M. Mézard, and M. A. Virasoro. Spin glass theory and beyond. *World Scientific, Singapore*, 187 :202, 1987.

[PPK+00] Georgios Paliouras, Christos Papatheodorou, Vangelis Karkaletsis, Panayotis Tzitziras, and Constantine D. Spyropoulos. Large-scale mining of usage data on web sites. In *AAAI Spring Symposium on Adaptive User Interfaces*, pages 92–97, 2000.

[PPPS03] Dimitrios Pierrakos, Georgios Paliouras, Christos Papatheodorou, and Constantine D Spyropoulos. Web usage mining as a tool for personalization : A survey. *User modeling and user-adapted interaction*, 13(4) :311–372, 2003.

[RB04] Jörg Reichardt and Stefan Bornholdt. Detecting fuzzy community structures in complex networks with a Potts model. *Physical Review Letters*, 93(21) :218701, 2004.

[RB06] Jörg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(1) :016110, 2006.

[RB08] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4) :1118–1123, 2008.

[RCC+04] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9) :2658–2663, 2004.

[Res00] Mauricio GC Resende. Detecting dense subgraphs in massive graphs. In *17th International Symposium on Mathematical Programming*, 2000.

[RRdlIRS06] Alan P. Reynolds, Graeme Richards, Beatriz de la Iglesia, and Victor J. Rayward-Smith. Clustering rules : a comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms*, 5(4) :475—-504, 2006.

[RS01] Ferrer-i-Cancho Ramon and Richard V. Solé. The small world of human language. *Proceedings of the Royal Society of London B : Biological Sciences*, 268(1482) :2261–2265, 2001.

[SCCH09] Huawei Shen, Xueqi Cheng, Kai Cai, and Mao-Bin Hu. Detect overlapping and hierarchical community structure in networks. *Physica A : Statistical Mechanics and its Applications*, 388(8) :1706–1712, 2009.

[SCDT00] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web usage mining : Discovery and applications of usage patterns from web data. *Acm Sigkdd Explorations Newsletter*, 1(2) :12–23, 2000.

[Sco12] John Scott. *Social network analysis : A Handbook*. SAGE publications, 2012.

[SDC+03] Parongama Sen, Subinay Dasgupta, Arnab Chatterjee, P. A. Sreeram, G. Mukherjee, and S. S. Manna. Small-world properties of the Indian railway network. *Physical Review E*, 67(3) :036106, 2003.

[SM10] Yacine Slimani and Abdelouahab Moussaoui. La fouille des usagers du web par application de l'algorithme Apriori sur les fichiers logs. *Revue d'information Scientifique & Technique*, 18(1) :18–34, 2010.

[SM12a] Yacine Slimani and Abdelouahab Moussaoui. A social network algorithm for detecting communities from weighted graph in Web Usage Mining system. In *The International Conference on Information & Communication Technology*, Al-Quds Open University, Ramallah (Palestine), 2012.

[SM12b] Yacine Slimani and Abdelouaheb Moussaoui. Analyse des sessions de navigations du site web de l'Université Ferhat Abbes de Sétif par les algorithmes de réseaux sociaux. In *9 ème Colloque sur l'Optimisation et les Systèmes d'Information*, page 40, Tlemcen (Algérie), 2012.

[SMD14] Yacine Slimani, Abdelouahab Moussaoui, and Ahlem Drif. Using weighted graph for discovery of usage patterns from web data. In *The First International Symposium on Informatics and its Applications*, M'sila (Algeria), 2014.

[SMD15] Yacine Slimani, Abdelouahab Moussaoui, and Ahlem Drif. Discovery and Analysis of Usage Patterns for Web Personalization. *International Journal on Recent and Innovation Trends in Computing and Communication*, 3(2) :578 – 582, 2015.

[SMDL18] Yacine Slimani, Abdelouaheb Moussaoui, Ahlem Drif, and Yves Lechevallier. Discovering Communities for Web Usage Mining Systems. *International Journal of Advanced Intelligence Paradigms*, 2018.

[SMG09a] Yacine Slimani, Abdelouahab Moussaoui, and Abdelmalek Guessoum. Extraction des règles d'associations depuis les fichiers logs dans un processus de fouille des usagers du web. In *Conférence Internationale des technologies de l'information et de la Communication*, Sétif (Algérie), 2009.

[SMG09b] Yacine Slimani, Abdelouahab Moussaoui, and Abdelmalek Guessoum. Prétraitement des fichiers journaux d'accès dans le processus de la fouille des usagers du web. In *Les 1ères Journées Scientifiques sur l'Informatique et ses Applications*, Guelma (Algérie), 2009.

[SMLD11] Yacine Slimani, Abdelouahab Moussaoui, Yves Lechevallier, and Ahlem Drif. A community detection algorithm for Web Usage Mining Systems. In *Fourth International Symposium on Innovation in Information & Communication Technology*, pages 112–117, Philadelphia University, Amman (Jordan), 2011. IEEE.

[SMLD12] Yacine Slimani, Abdelouahab Moussaoui, Yves Lechevallier, and Ahlem Drif. Identification de communautés d'usage du web depuis un graphe issu des fichiers d'accès. In *12ème Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances*, volume E23, pages 525–530, Bordeaux (France), 2012. Hermann.

[SOBB03] Daby M Sow, David P Olshefski, Mandis Beigi, and Guruduth Banavar. Prefetching based on web usage mining. In *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 262–281. Springer, 2003.

[SPGMA07] Marta Sales-Pardo, Roger Guimera, André A. Moreira, and Luís A. Nunes Amaral. Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences*, 104(39) :15224–15229, 2007.

[SPW+08] Yi Shen, Wenjiang Pei, Kai Wang, Tao Li, and Shaoping Wang. Recursive filtration method for detecting community structure in networks. *Physica A : Statistical Mechanics and its Applications*, 387(26) :6663–6670, 2008.

[SR05] Gabor J. Szekely and Maria L. Rizzo. Hierarchical clustering via joint between-within distances : Extending ward's minimum variance method. *Journal of Classification*, 22(2) :151–183, 2005.

[SSM05] B.S. Suryavanshi, N. Shiri, and S.P. Mudur. A fuzzy hybrid collaborative filtering technique for web personalization. *Intelligent Techniques for Web Personalization*, page 1, 2005.

[Str01] Steven H. Strogatz. Exploring complex networks. *Nature*, 410(6825) :268–276, 2001.

[Tan05] Doru Tanasa. *Web usage mining : Contributions to intersites logs preprocessing and sequential pattern extraction with low support*. PhD thesis, University of Nice Sophia Antipolis, 2005.

[TJK99] Brigitte Trousse, Michel Jaczynski, and Rushed Kanawati. Using user behaviour similarity for recommendation computation : the broadway approach. In *Proceedings of the HCI International*, volume 99, pages 85—-89, 1999.

[TK02] Pang-Ning Tan and Vipin Kumar. Discovery of web robot sessions based on their navigational patterns. *Data Mining and Knowledge Discovery*, 6(1) :9–35, January 2002.

[TK04] Pang-Ning Tan and Vipin Kumar. Discovery of web robot sessions based on their navigational patterns. In *Intelligent Technologies for Information Analysis*, pages 193–222. Springer, 2004.

[Tsu93] Shigehisa Tsuchiya. Improving knowledge creation ability through organizational learning. In *ISMICK'93 Proceedings, International Symposium on the Management of Industrial and Corporate Knowledge*, pages 87–95, 1993.

[TT04] Doru Tanasa and Brigitte Trousse. Advanced data preprocessing for intersites web usage mining. *Intelligent Systems, IEEE*, 19(2) :59–65, 2004.

[VD00] S. Van Dongen. *Graph clustering by flow simulation.* phd thesis, University of Utrecht, The Netherlands, 2000.

[Wat99] D. J. Watts. Small Worlds. *Princeton University Press*, 1999.

[Wat04] Duncan J. Watts. *Six degrees : The science of a connected age*. WW Norton & Company, 2004.

[WF94] Stanley Wasserman and Katherine Faust. *Social network analysis : Methods and applications*, volume 8. Cambridge university press, 1994.

[WM00] Richard J. Williams and Neo D. Martinez. Simple rules yield complex food webs. *Nature*, 404(6774) :180–183, 2000.

[WS98] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world'networks. *Nature*, 393(6684) :440–442, 1998.

[WS05] Scott White and Padhraic Smyth. A spectral clustering approach to finding communities in graphs. In *Proceedings of the 2005 SIAM international conference on data mining*, pages 274–285. SIAM, 2005.

[XZL10] Guandong Xu, Yanchun Zhang, and Lin Li. *Web mining and social networking : techniques and applications*, volume 6. Springer Science & Business Media, 2010.

[YJGMD96] Tak Woon Yan, Matthew Jacobsen, Hector Garcia-Molina, and Umeshwar Dayal. From user access patterns to dynamic hypertext linking. *Computer Networks and ISDN Systems*, 28(7-11) :1007–1014, 1996.

[YLK+16] Xiuming Yu, Meijing Li, Kyung Ah Kim, Jimoon Chung, and Keun Ho Ryu. Emerging pattern-based clustering of web users utilizing a simple page-linked graph. *Sustainability*, 8(3) :239, 2016.

[Zac77] Wayne W. Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.

[ZGH03] Tingshao Zhu, Russ Greiner, and Gerald Häubl. An effective complete-web recommender system. In *The Twelfth International World Wide Web Conference (WWW2003)*, 2003.

[Zho03a] Haijun Zhou. Distance, dissimilarity index, and network community structure. *Physical review e*, 67(6) :061901, 2003.

[Zho03b] Haijun Zhou. Network landscape from a Brownian particle's perspective. *Physical Review E*, 67(4) :041908, 2003.

[ZL04] Haijun Zhou and Reinhard Lipowsky. Network brownian motion : A new method to measure vertex-vertex proximity and to identify communities and subcommunities. In *International conference on computational science*, pages 1062–1069. Springer, 2004.

[ZR02] Djamel Abdelkader Zighed and Ricco Rakotomalala. Extraction de connaissances à partir de données (ECD). *Techniques de l'Ingénieur, H*, 3 :744, 2002.

[ZWL+08] Peng Zhang, Jinliang Wang, Xiaojia Li, Menghui Li, Zengru Di, and Ying Fan. Clustering coefficient and community structure of bipartite networks. *Physica A : Statistical Mechanics and its Applications*, 387(27) :6869–6875, 2008.

[ZZZ08] Junhua Zhang, Shihua Zhang, and Xiang-Sun Zhang. Detecting community structure in complex networks based on a measure of information discrepancy. *Physica A : Statistical Mechanics and its Applications*, 387(7) :1675–1682, 2008.

Résumé : La fouille du Web est l'application de la fouille de données sur les données Web. Le but de la fouille de l'usage du Web est de comprendre le comportement des utilisateurs des sites Web à travers le processus de fouille des données d'accès au Web. Les connaissances issues de la fouille de l'usage du Web peuvent être utilisées pour améliorer la conception du des sites Web, introduire un service de personnalisation du site et offrir une navigation plus efficace. La dissertation propose trois contributions principales; dans la première contribution, nous proposons un processus de fouille d'usage du Web. Le processus implique le prétraitement des données, l'intégration des données provenant de sources multiples et la transformation des données intégrées en une forme appropriée pour les opérations de fouille de données spécifiques. Notre deuxième contribution modélise les communautés Web et analyse la corrélation entre le comportement des utilisateurs dans la phase de découverte de modèles. Dans notre troisième contribution, on propose une nouvelle approche basée sur un processus stochastique qui converge vers une partition optimale du graphe du Web.

Mots clés : Fouille des usagers du web, sessions de navigation, motifs fréquents, extraction de connaissances, partitionnement du graphe, méthode de découverte de communautés, méthode de marche aléatoire, optimisation de la modularité, chaîne de Markov.

Abstract: Web mining is the application of data mining on the web data,. The purpose of the Web usage mining is to understand the behavior of website users through the process of searching web access data. The knowledge gained from Web browsing can be used to improve web design, introduce a personalization service and provide more efficient navigation. The dissertation proposes three main contributions. In the first contribution, we propose a web mining process which involves preprocessing data, integrating data from multiple sources, and transforming embedded data into a form suitable for specific data mining operations. Our second contribution models web communities and analyzes the correlation between user behavior in the model discovery phase. In our third contribution, we propose a new approach based on the stochastic process that converges to an optimal partition of the web graph.

Keywords: Web usage mining, browsing sessions, frequent patterns, information retrieval, graph partitioning, community discovery method, random walk method, modularity optimization, Markov chain.

الخلاصة: التنقيب الواب هو تطبيق تقنيات التنقيب في البيانات على البيانات المستخرجة. الهدف من التنقيب عن استخدام الويب هو فهم سلوك مستخدمي موقع الويب من خلال عملية تصفح بيانات الوصول إلى الويب. يمكن استخدام المعرفة المتعلقة باستخدام الواب لتحسين تصميم الموقع، وإدخال خدمة التخصيص، وتوفير ملاحة أكثر فعالية. تقترح الرسالة ثلاثة مساهمات رئيسية؛ في المساهمة الأولى، نقترح نمطية جديدة من أجل إجراء عملية التنقيب في الواب. تشتمل العملية على المعالجة الأولية للبيانات، ودمج البيانات من مصادر متعددة، وتحويل البيانات إلى شكل يتناسب مع عمليات التنقيب. المساهمة الثانية نقوم بقياس المجتمعات المحلية وتحليل العلاقة بين سلوك المستخدمين في مرحلة اكتشاف النماذج. في مساهمتنا الثالثة، نقترح نهج جديد يقوم على سلاسل ماركوف من أجل التقسيم الأمثل لمخطط البياني الخاص بإستخدام الواب.

الكلمات الرئيسية : التنقيب عن استخدام الويب ، جلسة التصفح ، الأسباب المتكررة ، استرجاع المعلومات ، تقسيم الرسم البياني، طريقة اكتشاف تجمع ، طريقة المشي العشوائي، سلسلة ماركوف.