Opinions Mining from Posters' Users in Social Networks

Radhia Toujani

Université de Tunis

Institut Supérieur de Gestion

THESE DE DOCTORAT

en vue de l'obtention du titre de docteur en

Informatique de Gestion

Opinions Mining from Posters' Users in Social Networks

Radhia Toujani

Soutenu le 25 Février 2021, devant le jury composé de :

Lamjed BEN SAID	Professeur, ISG Tunis	Président
Nahla BEN AMOR	Professeur, ISG Tunis	Rapporteur
Imen BOUKHRIS	Maître de Conférences, ENSI Tunis	Rapporteur
Raouia AYACHI	Maître de Conférences, ESSEC Tunis	Membre
Jalel Akaichi	Professeur, ISG Tunis	Directeur de thèse

Laboratoire/Unité de recherche : BEST MOD

Abstract

Nowadays, the analysis of social networks is a rich field of research and the analysis of opinions and the detection of communities is becoming a problem because of the considerable development of social media While existing sentiment analysis methods focus only on the extraction of positive and negative opinions, in our work we aim to extract fuzzy opinions. That is why we propose to follow Fuzzy Support Vector Machine during our sentiment analysis process. We also present a new hierarchical classification approach for the detection of communities sharing the same opinions. Our mixed hierarchical clustering technique, based on the assumption that there exists an initial solution composed of k partitions and the combination of ascendants and descendants methods, does not the change of the number of partitions, modifes the repartition of the initial structure. At the end of the introduced clustering process, a fixed point, representing a local optimum of the cost function which measures the degree of importance between two partitions, is obtained. Consequently, the introduced combined model leads to the emergence of local community structure. To avoid this local optimum and detect community structure converged to the global optimum of the cost function, the detection of community structures, in this study, is not considered only as a clustering problem, but as an optimization issue. In addition, it is essential not only to determine communities but also to identify the modifications in the network structure over the time. Additionally, we track the evolution of events over time. We develop, in this study, an approach for natural hazard events news detection and danger citizen' groups clustering . Analyzing the ambiguity and the vagueness of similarity of social publications plays a key role in event detection. This matter was ignored in traditional event detection techniques. To this end, we apply fuzzy sets techniques on the extracted events to enhance the clustering quality and remove the vagueness of the extracted information. Then, the defined degree of citizens' danger is injected as input to the introduced citizens clustering method in order to detect citizens' communities with close disaster degrees. The experimental analysis shows promising results made a set of comparisons between our proposed contributions and the existing state of the art.
Keywords : social network analysis, sentiment analysis, machine learning , bottom-up clustering, top-down clustering, mixed hierarchical clustering, optimization

Resumé

L'analyse des réseaux sociaux représente, de nos jours, un domaine de recherche riche et l'analyse des sentiments et la détection des communautés deviennent de plus une problématique à cause de développement considérable des médias sociaux. Cette thèse se concentre sur l'analyse et la modélisation des réseaux sociaux par l'extraction des opinions floues en utilisant Fuzzy Support Vector Machine et par la classification hiérarchique pour la détection des communautés partageant les mêmes opinions. Notre technique de classification hiérarchique mixte, basée sur l'hypothèse qu'il existe une solution initiale composée de k partitions et sur la combinaison des méthodes ascendantes et descendantes, ne modifie pas le nombre de classes, mais plutôt la distribution de la structure initiale des partitions. À la fin du processus de regroupement introduit, un point fixe, représentant un optimum local de la fonction de coût mesurant le degré d'importance entre deux partitions, a été obtenu. Par conséquent, le modèle combiné introduit a produit une structure communautaire locale. Pour éviter cet optimum local et détecter la convergence de la structure de communauté vers l'optimum global de la fonction de coût, la détection des structures de communauté, dans cette étude, n'a pas été considérée uniquement comme un problème de regroupement, mais aussi en tant que problème d'optimisation. Nous avons intégré les métaheuristiques dans notre processus de classification. De plus, il est essentiel non seulement d'identifier les communautés, mais également d'identifier les modifications apportées Ã la structure du réseau est de suivre l'évolution des évènements au cours du temps. Nous avons développé aussi, dans cette étude, une approche pour la détection des événements lors de danger naturel et le regroupement des communautés des citoyens en danger. L'analyse de l'ambiguité et du caractère vague de la similarité des publications socials joue un rôle clé dans la détection des événements. Cette problématique a été ignorée dans les techniques traditionnelles de détection des événements. À cette fin, nous avons appliqué les techniques flous sur les événements extraits pour améliorer la qualité du regroupement et supprimer le caractère vague des informations extraites. Ensuite, le degré de danger défini des citoyens a été injecté comme entrée de la méthode introduite de regroupement des citoyens afin de détecter les communautés de citoyens en situation de catastrophe proche. L'analyse expérimentale montre que les bons résultats obtenus ont permis de comparer nos contributions avec celles présentées dans l'état de l'art.

Mots clés : {analyse des réseaux sociaux, analyse des sentiments, apprentissage automa-

tique, clustering ascendant, clustering descendant, clustering hiÃ©rarchique mixte, optimisation}

Dédicaces

je dédie mes travaux de thè se à :

l'âme de mon père Sahbi ,
l'âme de ma mère Rachida ,
mon mari AbdErrahmen,
mes enfants Mohamed Mooemen et Mariem,
mes soeurs Houda, Henda et Fouzia,
ma soeurette Farah,
toute ma famille,
tous mes amis
et à tous les gens que j'aime

Remerciements

Avant tout, je remercie mon Dieu pour m'avoir donné le courage, la volonté et la patience pour réaliser ce présent travail.
Je tiens à remercier toute personne ayant contribué de prés ou de loin à l'aboutissement de ce travail.
Je tiens à adresser mes vifs remerciements et ma profonde reconnaissance à Monsieur Jalel Akaichi, Professeur à l'Université de Tunis, pour la confiance, pour avoir dirigé mon travail, pour son assistance et sa disponibilité, pour tous les encouragements, les remarques, les suggestions et les soutiens continus et pour l'intér êt qu'il a manifesté pour mon travail.
Ma profonde gratitude s'adresse également à tous les membres du laboratoire BESTMOD, qui ont dépensé tant d'efforts pour nous inculquer le savoir.

Table des matières

Table des figures

Liste des tableaux

Introduction Générale

Contexte de la thèse

Web 2.0 a apparu suite au développement des technologies et des méthodes numériques. Il permet à l'utilisateur d'accéder à des applications et de multi-plateformes qui facilitent la création et le partage du contenu et des données à grande échelle. Ce développement a donné naissance à un outil appelé "médias sociaux". Ce dernier, représentant des phases du développement du web, a changé radicalement les manières d'utilisation du web modernes ainsi que la personnalité des internautes à cause des nouvelles fonctionnalités "sociales". Se référant à de nouvelles activités, comme le partage des vidéos, des histoires et des images, les utilisateurs du web peuvent communiquer instantanément en temps réel en faisant des commentaires sur les documents connexes dans le réseau. En plus, les internautes peuvent accéder à leurs identités réelles plutôt que leurs pseudonymes, tel qu'il est le cas dans les anciennes plateformes en ligne.
En fait, les médias sociaux contiennent plusieurs sites (réseaux sociaux), applications et fonctionnalités qui sont strictement liés au progression des interactions conversationnelles et sociales entre les utilisateurs. A titre d'exemple, on peut citer Facebook, Twitter et linkedin qui sont les réseaux sociaux les plus employés par plusieurs catégories d'internautes : étudiants, chercheurs, fonctionnaires, marketeurs, etc. Ces réseaux, intégrant une masse volumineuse de données dont l'analyse permet de concevoir des comportements sociaux et quelques phénomènes sociaux. Les acteurs représentent l'élément principal dans la structure des réseaux sociaux. Les comportements humains et toutes nos activités sont généralement influencés par nos opinions et points de vue. En outre, nous ne pouvons pas prendre une décision que si on connait bien les opinions des autres individus concernant un sujet précis. La connaissance des opinions des consommateurs ou du public sur leurs produits et services est aussi importante pour les entreprises et les organisations. De leur part, les consommateurs veulent savoir les points de vue des autres individus concernant un produit ou les opinions des citoyens sur les candidats politiques avant les élections.

L'importance des opinions s'est manifestée au paravent par les enquêtes, les sondages d'opinion ainsi que par les groupes de discussion. De ce fait, connaître les avis du public et des consommateurs est un facteur décisif dans les domaines suivants : le marketing, les relations publiques et le secteur politique. Néanmoins, avec le développement consi-

dérable des médias sociaux (critiques, discussions de forum, blogs, micro-blogs, Twitter, commentaires et publications sur des sites de réseaux sociaux) sur le Web, le contenu des pages de ces médias devient de plus en plus utilisé pour la prise de décision. Lorsque nous désirons acheter un tel produit, nous ne consultons pas seulement l'avis de nos proches, mais aussi nous prenons en considération plusieurs critiques et discussions sur les forums publics sur le Web. Le travail d'une organisation devient plus facile puisqu'il existe de nombreuses informations sur les opinions du public accessibles sur les pages Internet. Par conséquent, l'analyse d'opinion sur le web constitue une activité primordiale dans tous les secteurs (économique, politique, santé, services financiers, etc.), d'où l'importance de l'analyse des sentiments dans les réseaux sociaux. Grâce à la simplicité de la communication et l'échange des opinions sur le Web, il devient nécessaire de participer ensemble dans des petits groupes dans les réseaux sociaux, nommés communautés, qui permettent l'échange rapide et efficace des opinions. En fait, les participants d'une communauté ont des points de vue et des intérêts communs. La structure de la communauté facilite l'extraction efficiente de l'information et garantit le partage des connaissances entre les membres d'une communauté spécifique. On peut donc définir la communauté comme étant un groupe humain qui a les mêmes opinions et intérêts. Ce partage est dû à l'homophilie montrant la tendance naturelle des humains à se réunir avec des individus ayants des caractères similaires.

En fait, une communauté peut être modélisée par un graph G= (V, E), où V dénote l'ensemble des noeuds et E représente l'ensemble des liens entre les noeuds de G. En conséquence, la détection de communauté consiste à trouver une partition P = C_1, ..., C_k de k communautés de l'ensemble des noeuds de V. Une communauté C_i est un sous-groupe de noeuds fortement liées plus qu'ailleurs dans le graphe G. Dans le monde réel, ces communautés ont généralement des structures imbriquées. Subséquemment, c'est plus logique que la modélisation d'une structure de communauté soit sous la forme d'un dendrogramme que sous la forme d'une partition. Pour investiguer la structure de la communauté à diverses granularités, il est nécessaire de faire un zoom avant ou arrière.

L'importance d'identification des communautés est due au fait que ce problème est souvent rencontré dans divers domaines d'application et dans plusieurs situations réelles. Far exemple, des communautés qui sont constituées par des individus, ayant les mêmes intérêts ou des fermes relations entre eux, sont présentes dans les réseaux sociaux. Par conséquent, il est possible de prévoir la consommation ainsi que le comportement des personnes en examinant les achats et les comportements de ceux appartenant à la même communauté. En identifiant des communautés, nous pouvons aussi déterminer le rôle que jouent plusieurs acteurs dans les communautés et, plus généralement, dans le réseau social. Le noeud ayant une position central au sein de sa communauté et partageant un grand nombre de liens avec les autres noeuds de la même communauté a un pouvoir de contrôle important et peut garantir la stabilité du réseau.

En fait, les noeuds localisées sur les frontières entre les communautés méditent bien et contrôle fortement les communications entre les communautés. En outre, de nombreuses techniques de détection de communautés ont été proposées. La première catégorie inclut les techniques de classification hiérarchiques permettant de sélectionner la structure de

communauté parmi divers niveaux hiérarchiques démontrant plusieurs structures possibles. La deuxième catégorie contient les techniques d'optimisation d'une fonction objective qui spécifient les communautés en optimisant la fonction de qualité. Ces méthodes maximisent localement la représentation statistique d'une communauté. Irrémédiablement, la dernière classe contient les techniques à base du modèle.
En plus, il est non seulement nécessaire de détecter la communauté, mais aussi de contrôler le développement de la structure des réseaux sociaux qui progressent au fil du temps. Pour mieux interpréter ce dynamique, il est primordial de déterminer comment les groupes d'une communauté persévèrent et se développent. Une communauté peut être définie comme une chaîne cohérente de groupes contemplés dans différents jours. Par conséquent, le suivi des communautés est basé sur les liens entre divers groupes du même jour et des différentes communautés.

Des informations concernant des divers évènements sont partagées, discutés et retransmises par les utilisateurs des médias sociaux en temps réel. Ce volume, constamment grandissant, fait des messages publiés sur les médias sociaux des sources informationnelles riches et réactives. Pour cela, les médias traditionnels perdent leur popularité.

Cependant, la croissance de ce volume résulte en une surcharge informationnelle, ce qui rend l'identification des éléments d'information pertinents en relation avec des évènements importants plus difficile. En fait, le terme « évènement important » désigne un fait réel et probable d'être couvert par les médias traditionnels. Dans ce contexte, il est nécessaire de poser la question suivante : comment peut-on exploiter les médias sociaux pour détecter automatiquement les évènements importants ? En répondant à cette question, nous pourrons surtout analyser les évènements qui provoquent plus l'attention des utilisateurs des médias sociaux. Cette analyse est profitable dans le cadre de la veille d'information, du journalisme de données, du marketing, etc. En fait, la détection automatique des évènements dominants à partir des médias sociaux est une tâche difficile car les messages décrivant ces évènements sont la plupart sans rapport. La procédure de détection d'événement consiste principalement à transformer les informations non-structurées qui n'ont pas des données pertinentes dans le format original. Par conséquent, le contrôle de la vague et l'incertitude de la connaissance réelle demeure primordial. En plus, le sens du contenu textuel des événements change selon la variation des domaines d'application.
De plus, les significations objectives et les indications de subjectivité de l'actualité des événements employés représentent une source importante pour la détection des événements. De ce fait, les bénéfices des approches floues permettent d'introduire des techniques plus performantes pour la représentation des événements.

Problématiques de la thèse
Les problématiques abordés dans cette thèse sont les suivants :

— L'utilisation des médias sociaux par les utilisateurs est fondamentale, car elle leur permet de partager des opinions et rechercher des informations sur divers sujets. Pour mieux comprendre la perception, les opinions et les émotions exprimées dans une mention en ligne, il est primordial d'analyser l'opinion de chaque utilisateur des réseaux sociaux. Cependant, les méthodes d'analyse des sentiments existantes se concentrent uniquement sur l'extraction des opinions positives et négatives et

négligent l'extraction des opinions floues. En fait, la signification du mot sentiment varie selon les domaines d'application. Comme dans les phrases interrogatives et conditionnelles, les mots de sentiment ne peuvent exprimer aucune émotion et les phrase présentant objectivement des faits n'expriment aucun sentiment.

— Analyser l'opinion des groupes dans les réseaux sociaux et détecter les communautés partageant les mêmes opinions est un problème principal dans la modélisation et l'analyse des réseaux sociaux. En fait, les réseaux montrent la structure et les relations hiérarchiques des communautés. Cette hiérarchie simplifie la division du réseau en quelques grandes communautés qui peut être subdivisées en unités plus petites. En fait, la définition de la structure hiérarchique et modulaire appropriée des réseaux complexes joue un rôle primordial pour la compréhension des systèmes complexes où la détection de communauté est considérée comme un problème de partitionnement de graphe NP-complet.

— La détection des structures communautaires localement optimales est un problème très compliqué à traiter. Le fait de considérer le processus de classification comme un problème d'optimisation accentue davantage la difficulté de l'identification d'une strcuture de communauté globalement optimale. L'analyse des communautés d'optima locaux nécessite des corrélations intéressantes entre les caractéristiques du communauté et la difficulté de recherche connue des problèmes combinatoires.

— L'interaction dynamique des utilisateurs des réseaux sociaux pendant les évènements provoque la nécessité de suivre l'évolution des communautés au fils du temps. Il est important non seulement de détecter les communautés, mais également de suivre l'évolution de la structure du réseau au fil du temps. Naturellement, les réseaux sociaux évoluent fréquemment. Suivre et comprendre l'évolution sont deux tâches difficiles qui nécessitent l'identification des événements importants pouvant survenir aux communautés et aux individus. La plupart des réseaux sociaux évoluent avec le temps en raison des changements fréquents : des noeuds peuvent rejoindre ou quitter des communautés ; des nouveaux liens peuvent être créés et d'autres peuvent disparaître et probablement former des nouvelles communautés. En outre, les utilisateurs des réseaux sociaux se joignent à divers les événements. Pour cette raison, la participation à des événements communs peut développer des nouvelles communautés. En d'autres termes, les informations sur les événements permettent de déduire des sous-groupes, d'où la nécessité de suivre les variations des événements.

Toutefois, les techniques de détection de la communauté proposées se sont focalisées particulièrement sur le graphe statique et elles ont abandonné les caractéristiques temporelles du graphe. Par conséquence, une attention importante et une littérature riche ont été présentées pour étudier les mécanismes par lesquels le réseau s'évolue et un lien apparaît.

Objectifs de la thèse

Dans cette thèse, nous introduisons une approche de modélisation et d'analyse des réseaux sociaux. Nous visons, dans cette thèse, à :

- examiner les opinions exprimées dans les réseaux sociaux en développant une tech-

nique efficiente pour classifier les sentiments ;
- présenter les réseaux sociaux sous forme de graphe et investiguer les communautés ayant les mêmes opinions en développant une nouvelle approche de classification hiérarchique. Subséquemment, notre étude traite en plus de l'analyse des opinions et des sentiments existants dans les réseaux sociaux la classification hiérarchique pour la détection de communautés partageant les mêmes opinions ;
- incorporer les méta-heuristiques dans notre procédure de classification pour crée une structure communautaire qui se rapproche de l'optimum globale vue la convergence de la structure communautaire obtenue par la classification hiérarchique proposée vers un optimum local ;
- étudier la progression de la structure des communautés dans le temps réel. Nous utilisons les méthodes d'apprentissage automatique (arbre de décision) pour investiguer l'évolution des opinions au cours du temps. Nous suggérons aussi une méthode de prédiction des liens d'amitié ;
- introduire une technique pour l'analyse des évènements et la détection automatique des évènements primordiaux lies aux catastrophes naturelles.

Structure de la thèse

Ce rapport est constitué de deux parties :

- La première partie présente l'état de l'art contenant trois chapitres.

  Chapitre 1, intitulé « Analyse des sentiments et opinions exprimés dans les réseaux sociaux », détaille les grandes classes d'approches d'analyse des sentiments dans les réseaux sociaux. Dans les années récentes, ce sujet est a été intensivement discuté par les chercheurs. Par conséquent, plusieurs méthodes liées à l'analyse des sentiments ont été proposée dans la littérature.

  Dans le deuxième chapitre, intitulé « Détection de communautés dans les réseaux sociaux », nous examinons les réseaux et les communautés du monde réel. D'une part, cette étude se focalise sur les diverses stratégies de partitionnement et les approches utilisées pour détecter les communautés et, d'autre part, elle présente la structure communautaire ainsi que le piège d'optimum local.

  Chapitre 3, intitulé « détection et suivi de l'évolution des structures communautaires et d'évènements dans les réseaux sociaux », s'intéresse au suivi de la progression du structure communautaire en fonction du temps. Il présente l'état de l'art des approches de suivi des structures communautaires et de prédiction des liens (« la structure » du réseau). De plus, ce chapitre décrit les techniques appliquées pour la détection des événements en temps réel.

- Dans la deuxième partie, nous présentons nos contributions. Cette partie est composée de six chapitres.

  Dans le quatrième chapitre, nous introduisons une approche pour analyser les

sentiments dans les réseaux sociaux. Cette approche est constituée de trois étapes importantes. Après l'assemblage des statues des réseaux sociaux, nous traitons automatiquement le langage. Ultérieurement, nous effectuons le processus de classification floue en développant une machine à vecteur de support.

Dans le chapitre 5, nous développons une approche de classification mixte consistant à pratiquer une classification ascendante hiérarchique à fin de modéliser le problème de détection de communauté en forme de graphe pondéré. Après, nous proposons deux algorithmes heuristiques de classification qui sont combinés pour créer un troisième algorithme ayant une optimalité locale.

Dans le chapitre 6, nous nous concentrons plutôt sur l'intégration des métaheuristiques dans l'approche de classification hiérarchique proposée afin d'éviter la convergence vers l'optimum locale et se rapprocher de l'optimum global.

Nous suggérons, dans le chapitre 7, une technique pour contrôler la mobilité des communautés. Nous proposons aussi une approche de détection automatique des événements importants. Nous décrivons spécifiquement les événements qui attirent l'attention des utilisateurs des médias sociaux dans les périodes des catastrophes naturelles.

Le chapitre 8 décrit l'expérimentation réalisée pour valider les contributions introduites. Ces expérimentations ont été performées via des différents jeux de données assemblés sur les média sociaux prouvent la pertinence de nos contributions.

- Dans la conclusion générale, nous présentons un bilan détaillé de la thèse et nous évoquons nos diverses contributions. Nous discutons aussi des nouvelles perspectives pour la recherche future.

Liste des publications

Les travaux entrepris par la doctorante Radhia Toujani ont conduit à la production de deux papiers impactés dans des journaux internationaux :

1) Toujani, R., and Akaichi, J. (2019a). An approach based on mixed hierarchical clustering and optimization for graph analysis in social media network : toward globally hierarchical community structure. Knowledge and Information Systems, 1-41.

2) Toujani, R., and Akaichi, J. (2019b). Event news detection and citizens community structure for disaster management in social networks. Online Information Review, 43(1), 113-132.

En plus, des articles sont publiés dans des conférences soit indexés soit classées : 1) Toujani Radhia, D. Z., and Jalel, A. (2015). Machine learning and metaheuristic for sentiment anaysis in social networks. In proceedings of the metaheuristic internatianal conference(mic'15).

2) Toujani, R., and Akaichi, J. (2016). Fuzzy sentiment classification in social network facebook'statuses mining. In 2016 7th international conference on sciences of electronics, technologies of information and telecommunications (setit) (pp. 393-397).

3) Toujani, R., and Akaichi, J. (2017). Optimal initial partitionning for high quality hybrid hierarchical community detection in social networks. In 4th international conference on control, decision and information178 Références bibliographiques technologies (codit)

(pp.0395-0403).
4) Toujani, R., and Akaichi, J. (2017). Sentiment Classification Method for Identification of Influential Learners in Social Networks Communities. In LPKM.
5) Toujani, R., and Akaichi, J. (2017). Hybrid Hierarchical Clustering Approach for Community Detection in Social Network. World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 11(6), 654-660.
6) Toujani, R., and Akaichi, J. (2017). A Model Based Metaheuristic for Hybrid Hierarchical Community Structure in Social Networks. ISi, 1, 1.
7) Toujani, R., and Akaichi, J. (2018). Ghhp : Genetic hybrid hierarchical partitioning for community structure in social medias networks. In 2018 ieee smartworld, ubiquitous intelligence and computing, advanced and trusted computing, scalable computing and communications, cloud and big data computing, internet of people and smart city innovation (smartworld/scalcom/uic/atc/cbdcom/iop/sci) (pp. 1146-1153)
8) Toujani, R., Chaabani, Y., Dhouioui, Z., and Bouali, H. (2018). The next generation of disaster management and relief planning : Immersive analytics based approach. In International conference on immersive learning (pp.80-93).
9) Toujani, R., Dhouioui, Z., and Akaichi, J. (2018). Mobility based machine learning modeling for event mining in social networks. In International conference on intelligent interactive multimedia systems and services (pp. 311-322).
10) Zeineb Dhouioui, Radhia Toujani and Jalel Akaichi. Tracking Dynamic Community Evolution and Events Mobility in Social Networks. Encyclopedia of Social Network Analysis and Mining. 2nd Ed. 2018 11) Chaabani Y., Toujani R., Akaichi J. (2018) Sentiment Analysis Method for Tracking Touristics Reviews in Social Media Network. In : De Pietro G., Gallo L., Howlett R., Jain L. (eds) Intelligent Interactive Multimedia Systems and Services 2017. KES-IIMSS 2017. Smart Innovation, Systems and Technologies, vol 76. Springer, Cham. Vilamoura, Portugal.

Contexte et Etat de l'Art

Chapitre 1

Analyse des sentiments et opinions exprimés dans les réseaux sociaux

1.1 Introduction

Vu l'utilisation massive des réseaux sociaux, les opinions sont de plus en plus façonnées par les interactions sociales et les phénomènes sociaux qui sont étroitement liés à la vie quotidienne. La croissance explosive des données dans les réseaux sociaux apporte plus d'opportunités et de défis à l'analyse des sentiments. A ce propos, il y a un besoin d'outils qui peuvent analyser ces discussions et en extraire les opinions, les attitudes et les émotions des utilisateurs des réseaux sociaux. Dans le premier chapitre de notre thèse, nous exposons une étude bibliographique, non-exhaustif concernant l'analyse des sentiments nommée en anglais *« Opinion Mining »*. Pendant les dernières années, cette analyse a connu un progrès considérable grâce à son application dans plusieurs domaines. Dans la section 1.2, nous montrons l'importance des opinions dans les réseaux sociaux. Nous décrivons dans la section 1.3 les diverses approches existantes d'analyse des sentiments. Dans la section 1.4 nous mettons en évidence la litérature des méthodes flous d'analyse de sentiments. Nous terminons par mettre l'accent sur les mesures de performances pour évaluer les méthodes d'analyse de sentiments dans la section 1.5.

1.2 Analyse des émotions, sentiments et opinions exprimés dans les réseaux sociaux

Les différents aspects de l'analyse des sentiments, à savoir la fouille d'opinions, la complexité de notation d'opinion, le besoin de connaitre les sentiments des autres ainsi que la polarité et l'intensité de l'opinion sont illustrés dans cette partie.

1.2.1 Fouille d'opinions et analyse des sentiments

La fouille d'opinions (opinion mining), «analyse des sentiments » (sentiment analysis), représente un sous-domaine informatique dans lequel plusieurs disciplines, comme la fouille du texte, le traitement automatique du langage (TAL), la recherche d'information, fouille d'opinion et l'apprentissage automatique, sont considérés(Haccianella et al., 2010, accianella et al., 2010). En fait, l'analyse des données textuelles importantes dans les réseaux assure une compréhension approfondie des comportements des individus ainsi que une étude de quelques évolutions sociales. La première étape de l'évaluation des sentiments dans les réseaux sociaux consiste à rechercher et extraire les opinions exprimées.

De plus, une nouvelle problématique qui se pose pour le TAL, aussi connu par la catégorisation de texte (TC), est l'étude des messages échangés qui sont généralement complexes. Cette technique est utilisée pour indexer les documents basés sur un lexique spécifique, filtrer les documents, produire systématiquement des méta-données, clarifier le sens des mots, former des catalogues hiérarchiques de ressources Web, etc. La TC met souvent en évidence les éléments essentiels pour organiser, traiter sélectivement et adapter des documents (Haccianella et al., 2010, accianella et al., 2010)(Nakov et al., 2016, akov et al., 2016).

Ainsi, la catégorisation des textes est une méthode qui combine deux autres techniques : l'Apprentissage Automatique (ang : Machine Learning - ML) et la Recherche d'Information (IR). En fait, un classificateur de texte par l'apprentissage est construit automatiquement à partir d'un ensemble de documents pré- classifiés ou de caractéristiques de catégories d'intérêts en appliquant la "ML". Par conséquent, la Fouille d'opinions Textes peut être considérée comme une série de traitements informatiques qui consistent à extraire des connaissances en se basant sur des critères de nouveauté ou de similarité spécifiés dans des textes écrites pardes humains(Joachims and Sebastiani, 2002, oachims and Sebastiani, 2002).

1.2.2 Les besoins de connaitre les sentiments des autres

Les médias sociaux ont une très grande importance car chaque utilisateur présente lui-même un auteur potentiel et le langage avec lequel il exprime ces idées, sentiments et opinions reflète sa réalité (Zhou et al., 2015, hou et al., 2015). En fait, la connaissance des

points de vue des autres individus était régulièrement un élément essentiel d'information dans la procédure de décision puisque, avant que les personnes prennent une décision, ils consultent d'autres individus pour savoir leurs opinions.

Secteur industriel

Dans le secteur industriel, le contrôle des données extraites des médias sociaux est très important. Ces données jouent un rôle primordial pour améliorer considérablement l'efficacité de la veille stratégique. Leur intégration dans les systèmes de veille stratégique rend la réalisation des objectifs des entreprises plus facile, particulièrement pour ce qui concerne la stratégie de marque et la notoriété, la gestion des clients actuels et potentiels et l'amélioration du service.

Défense et sécurité nationale

Ce secteur de défense et sécurité nationale se concentre spécifiquement sur l'étude des données dans les médias sociaux pour mieux concevoir les diverses situations, analyser les sentiments d'un groupe de personnes ayant les mêmes intérêts et être attentif aux menaces probables dans les domaines cibles. Dans ce contexte, de nombreuses techniques ont été introduites pour extraire des informations (par exemple l'extraction des entités nommées et des liens entre ces dernières) à partir du Web 2.0 et afin d'analyser le contenu des réseaux sociaux dans lesquels des utilisateurs et même des organisations s'évoluent. Ces données fournissent des importants renseignements pour la sécurité nationale.

Soins de santé

En outre, les médias sociaux sont intensivement utilisés par les malades pour discuter des sujets liés à certaines maladies (les traitements, les médicaments et même les recommandations à l'intention des professionnels), ce qui montre leur pertinence dans ce secteur. De plus, ces forums de discussion représentent une source d'après laquelle les professionnels de santé et, plus particulièrement, les médecins peuvent mieux comprendre les perceptions des patients de leurs maladies(Nzali et al., , zali et al.,).

Politique

Le contrôle des médias sociaux garantit le suivi des mentions effectuées par de nombreux citoyens et leurs points de vue envers un parti politique. D'après l'extraction, le suivi et l'étude de ces opinions publiées, un parti politique peut mieux observer la valeur de certains événements. Ces trois procédures lui donne l'occasion d'améliorer ses positionnements politiques (Bakliwal et al., 2013, akliwal et al., 2013). Par conséquence, les médias sociaux jouent un rôle important dans le déroulement de la campagne électorale.

En se basant sur les idées susmentionnées, on peut conclure que l'utilisation de l'internet nous permet de savoir les points de vue et les expériences de plusieurs personnes que nous ne connaissons pas, mais nous partageons avec eux les mêmes goûts. Les opinions de ces individus nous aident à prendre une décision, orienter nos choix, changez nos préjugés sur un sujet donné, etc.

1.2.3 La complexité de notation d'opinion

La problématique majeure qui se pose, dans l'étape du traitement des données extraites à partir des réseaux sociaux, est « Big Data » avec ses trois V : volume, variété et vélocité. En fait, les réseaux sociaux sont constitués des acteurs liés par des liens ou des interactions, d'où la nécessité de modéliser la structure d'un groupe social pour préciser son effet sur d'autres variables, et de suivre son évolution. Par conséquent, l'étude des données volumineuses et hétérogènes issues des médias sociaux en temps réel montre que le concept d'opinion est complexe.

Volume

En 2013, e-Marketer a publié, dans New Media Trend Watch(Commission et al., 2013, ommission et al., 2013), un rapport dans lequel il a estimé qu'à l'échelle mondiale une personne sur quatre utilisait les médias sociaux en 2013. Pour l'année 2012, des études statistiques sur les médias sociaux ont prouvé que le nombre des utilisateurs actifs du Facebook a dépassé huit cents millions ; parmi eux, deux cents millions sont des nouveaux adhérents au cours d'une seule année. De plus, la plate-forme Twitter contient cent millions d'utilisateurs et LinkedIn (soixante-quatre millions de ces utilisateurs habitent en Amérique du Nord (Farzindar and Roche, 2013, arzindar and Roche, 2013). Les statistiques ont montré aussi que plus de trois cent millions de tweets ont été envoyés à Twitter chaque jour (Tang et al., 2014, ang et al., 2014).

L'analyse de ce contenu riche régulièrement renouvelé nous permet d'accéder à une source d'information précieuse que les médias traditionnels ne peut offrir (Melville et al., 2009, elville et al., 2009). L'analyse sémantique des médias sociaux a ouvert la voie à l'analyse de données volumineuses, discipline émergente inspirée de l'apprentissage automatique, de l'exploration de données, de la recherche documentaire, de la traduction automatique et du résumé automatique.

Vélocité

Les messages écrits sur les réseaux sociaux sont généralement produits en temps réel. Ceux qui traitent un sujet commun transmettent des émotions, des néologismes ou des rumeurs. Puisque ces messages peuvent découler de diverses localisations, il est essentiel de considérer la vélocité de production des données.

Les médias sociaux mettent en relief l'utilité de la recherche des événements en temps réel et l'importance de les détecter (Atefeh and Khreich, 2015, tefeh and Khreich, 2015). Ces deux procédures (recherche et détection) exigent l'application des stratégies de recherche efficients à partir de plusieurs fonctionnalités qui considèrent de nombreuses dimensions telles que les liens spatiaux et temporels (Moncla et al., 2014, oncla et al., 2014). De plus, les discussions liées à un événement spécifique peuvent combiner, pendant une durée de temps courte, divers sujets. Ceci montre la problématique de l'hétérogénéité des données.

Variété

Les informations accessibles dans les médias sociaux forment une source de renseignements. Cependant, les textes rédigés par plusieurs auteurs en diverses langues et différents styles n'ont aucune structure précise. Ils sont présentés sous plusieurs formats : blogues, microblogues, forums de discussion, clavardages, jeux en ligne, annotations, classements, commentaires et FAQ créés par des utilisateurs, etc. L'existence des nombreux plans, contenus et styles font de l'analyse globale une tâche difficile.

1.2.4 La polarité et l'intensité de l'opinion

Une opinion contient deux composants majeurs : une cible g et un sentiment s sur la cible, c'est-à-dire (g, s) où g dénote toute entité et tout aspect de l'entité sur lesquels cette opinion est exprimée, et s correspond à une opinion positive, un sentiment négatif ou neutre, ou une note numérique montrant la force /l'intensité du sentiment (par exemple, 1 à 5 étoiles). En fait, les deux approches importantes de la représentation des documents sont basées sur le modèle du sac de mots (en ang the Bag of Words Model (BOW)) (Sebastiani, 2002, ebastiani, 2002) et celui de l'espace vectoriel (Mitra et al., 2016, itra et al., 2016)(en ang the Vector Space Model (VSM)).

La représentation du texte, dans le premier modèle BOW, est faite par un vecteur de caractéristiques comprenant tous les mots qui y figurent. En conséquence, la dimension de l'espace de représentation du document est égale au nombre de mots différents dans tout le texte. L'extraction de chaque mot est effectuée à partir du texte en tenant compte des séparateurs tels que l'espace, la tabulation et la ponctuation. Dans ce cas et si le nombre de mots caractérisant le corpus de documents est assez élevé, il est obligatoire de garder un sous ensemble de ces mots. Ce filtrage est basé sur les fréquences d'occurrences des mots dans le corpus.

De nombreuses possibilités ont été proposées pour calculer l'orientation sémantique des mots. La technique de l'orientation sémantique des associations (SO-A) est calculée en soustraire une mesure de l'association des mots positifs d'une mesure de l'association des mots négatifs :

$$SO{-}A(mot) = \sum pmotsA \in pmotsA(mot, pmots) - \sum nmotsA \in nmotsA(mot, nmots) \tag{1.1}$$

$A(mot, nmots)$ désigne l'association du mot étudié avec le mot négatif. Si la somme est positive, le mot est orienté positivement, sinon, l'orientation est négative. La valeur absolue de la somme montre le degré d'intensité de l'orientation.

La mesure de l'association entre les mots A peut être calculé par plusieurs méthodes telles que the Pointwise Mutual Information - SO-PMI.

$$PMI(mot1, mot2) = \log_2 \frac{p(mot1 \& mot2)}{p(mot1)p(mot2)} \tag{1.2}$$

Le $p(mot1\&mot2)$ spécifie la probabilité de co-existence de deux mots.

La deuxième possibilité consiste à analyser la relation statistique entre les mots dans le corpus appliquée la méthode Singular Value Decomposition (SVD). La technique qui utilise SVD est appelée Latent Semantic Analysis - SO-LSA dans laquelle la matrice contenant, en ligne et en colonnes, les pondérations des mots et des parties du texte telles que les phrases ou les paragraphes, est décomposée. Cette pondération est souvent calculée par rapport au tf-idf (Term Frequency Inverse Document Frequency)(Alfaro et al., 2016, lfaro et al., 2016).

L'application de la procédure de classification du texte dj en une représentation compacte de son contenu doit être uniforme aux documents d'apprentissage et de validation ainsi que aux documents des tests. Le choix d'une représentation du texte est dépendent des unités linguistiques exprimant le sens du texte (le problème de sémantique lexicale). Les approches d'indexation sont classées en deux catégories :

— celles basées sur l'étude des divers moyens utilisés pour mieux comprendre le concept d'unité linguistique,
— celles basées sur plusieurs méthodes de calcul des poids des unités.

Il a été prouvé que l'utilisation comme unité linguistique de représentations plus sophistiquée que le mot ne donne pas des résultats beaucoup plus fiables (Atefeh and Khreich, 2015, tefeh and Khreich, 2015)(Alfaro et al., 2016, lfaro et al., 2016). En fait, ses résultats sont dus au fait que le traitement statistique est moins important que l'indexation basée sur les mots tandis que l'indexation reposant sur les phrases est caractérisée par une sémantique de qualité supérieure. Puisque, dans l'indexation des mots composés, il y a plus d'unités, plus de synonymes, une plus faible cohérence de la correspondance (comme les synonymes ne sont pas affectés aux mêmes documents), et une fréquence inférieure d'unités par document. La meilleure solution qui peut être adoptée ici pour améliorer les résultats est de combiner ces deux approches.

Le poids des unités varie généralement entre 0 et 1. Il peut être binaire (1 révèle la présence du terme dans le document et 0 montre son absence). Pour ce qui concerne l'indexation non-binaire, toutes les méthodes d'indexation de IR, représentant un document comme un vecteur de termes pondérés, peuvent être appliquées pour déterminer le poids de w_{kj} de l'unité t_k dans le document d_j. La fonction $tf - idf$ souvent employée est définie comme suit :

$$tf - idf(t_k; d_j) = *(t_k; d_j).log\frac{|T_r|}{*T_r(T_k)} \tag{1.3}$$

où $*(t_k; d_j)$ correspond au nombre de fois que t_k se produit en d_j, et $*T_r(T_k)$ représente la fréquence du document de terme t_k. Autrement dit le nombre des documents d'ensemble d'apprentissage (EP) dans lesquels t_k est formée. Cette fonction montre que, plus un terme apparait souvent dans un document, plus il est représentatif ; plus le nombre de documents contenant un terme est important moins ce terme est discriminatoire.

1.3 Les méthodes d'analyse des sentiments

Dans cette section, nous décrivons les approches d'analyse des sentiments largement appliquées dans la littérature. Elles peuvent être catégorisées en deux classes : celles basées sur l'apprentissage automatique et celles basées sur les dictionnaires(Ravi and Ravi, 2015, avi and Ravi, 2015)(Yadollahi et al., 2017, adollahi et al., 2017).

1.3.1 Les approches basées sur l'apprentissage automatique

Les approches d'apprentissage automatique sont intensivement employées pour la classification des textes. Elles sont généralement classées en deux types majeurs : Apprentissage supervisé et apprentissage non-supervisé. Pour le premier type, la connaissance des étiquettes de classe se passe avant l'apprentissage. Mais, ces étiquettes sont inconnues dès le début pour les méthodes d'apprentissage non supervisé.

Pang et al.(Pang et al., 2002, ang et al., 2002) ont utilisé trois techniques pour classer les critiques cinématographiques en se basant sur les classificateurs suivants : « naïf Bayes », entropie maximale et classificateur SVM (Support Vector Machine). Les résultats obtenus par SVM sont les meilleurs avec un taux de pertinence égal à 83% en employant les unigrammes. Pang et Lee (Pang and Lee, 2004, ang and Lee, 2004) ont aussi développé une autre approche pour classer la polarité des critiques cinématographiques. Cette approche peut être divisée en de deux étapes. La première consiste à détecter les parties subjectives des documents. Ultérieurement, ils ont appliqué le même classificateur statistique pour la détection de polarité uniquement sur les fragments subjectifs qui sont détectés précédemment. Pang et Lee ont prouvé qu'il y a un certain degré de continuité dans la subjectivité des phrases car un auteur est généralement subjectif ou objectif. Ils ont essayé d'attribuer, aux phrases à proximité, le même degré de subjectivité. Ensuite, pendant la procédure de classification collective, toutes les phrases du document ont été classées comme subjectives ou objectives.

En 2002, Turney a introduit un algorithme d'apprentissage non supervisé, nommé « algorithme d'information mutuelle et de recherche d'informations », pour étiqueter les textes comme recommandés ou non recommandés(Turney, 2002, urney, 2002).

Dave et al.(Dave et al., 2003, ave et al., 2003) ont suggéré une autre approche pour montrer si la critique est positive ou négative. Les auteurs ont sélectionné, dans une première étape, un ensemble de caractéristiques f_1,...,f_n. Ultérieurement, les notes ont été attribuées aux caractéristiques afin de classer les documents de test en tant que critiques positives (c) ou négatives ($C\prime$). Après, la fréquence d'occurrence normalisée $-p(f_i|C)$ a été déterminée en considérant le nombre des occurrences d'une caractéristique f_i dans C et en divisant par le nombre total de tokens dans C. La valeur de la note attribuée aux caractéristiques varie entre -1 à 1.

$$score(f_i) = \frac{p(f_i|C) - p(f_i|C\prime)}{p(f_i|C) + p(f_i|C\prime)} \tag{1.4}$$

Après avoir noté chaque caractéristique, les notes des mots d'un document inconnu ont été additionnées et le signe de cette somme a été employé pour spécifier la classe C ou $C\prime$. Par conséquent, pour un document d_j=f_1,..., f_n.

$$class(d_i) = (Csieval(d_j) > 0 ou C\prime sieval(d_j) < 0) \tag{1.5}$$

Cette approche a produit une pertinence de 76%.

Une approche basée sur le bootstrapping a été aussi appliquée pour détecter la subjectivité des phrases. Dans cette approche, la sortie d'un classificateur initial a été utilisée pour étiqueter les données sur lesquelles on peut appliquer l'algorithme d'apprentissage. Cette technique a été employée par Riloff et Wiebe (Riloff and Wiebe, 2003, iloff and Wiebe, 2003) avec un classificateur initial ayant une importante précision pour la préparation de la phase d'apprentissage qui consiste à extraire les occurrences pour les expressions subjectives. Des comportements intéressants ont été obtenus. A titre d'exemple, dans le contexte « The fact is... », le mot « fact » est caractérisé par une corrélation robuste avec la subjectivité. Deux classificateurs de haute précision et faible rappel ont été utilisés dans cette approche : un classificateur de subjectivité et un classificateur d'objectivité. Les classificateurs appliqués sont basés sur un groupe de mots uniques ainsi que sur un ensemble de n-grammes ou d'unités lexicales extraits manuellement et révélant une relation importante de subjectivité. Ultérieurement, à la fin de l'étape d'apprentissage les phrases retrouvées ont été réintroduites dans le classificateur afin d'améliore leur étiquetage comme des phrases subjectives ou objectives. La répétition de cette procédure fait accroître la précision et réduit, par conséquence, la valeur de rappel. La méthode de Riloff et Wiebe a donné un taux de précision inférieur à 90% et une valeur de rappel égale à 40%.

Dans le domaine de l'analyse des sentiments, plusieurs approches d'apprentissage automatique ont été introduites. Parmi ces approches, on peut citer les modèles de régression qui permettent d'anticiper l'utilité d'une revue (Zhu and Zhang, 2006, hu and Zhang, 2006) ainsi que la technique semi-supervisée utilisée pour la classification binaire des textes comme positifs ou négatifs (Esuli and Sebastiani, 2005, suli and Sebastiani, 2005). L'effet de Naïve Bayes et de Support Vector Machine sur les journaux Web politiques a été examiné par Durant et Smith (Durant and Smith, 2006, urant and Smith, 2006) qui ont montré également que les résultats obtenus par le classificateur Naïve Bayes et plus importants que ceux fournis par Support Vector Machine.

En 2004, le simple classificateur en ligne Winnow a été appliqué par Hurst et Nigam afin de préciser la polarité des documents. Les auteurs ont constaté que le taux de rappel et de précision de l'accord humain varie entre 75% et 80% sur la prévision de polarité. La valeur de rappel obtenue en utilisant Winnow était négligeable ; elle n'a pas dépassé 43%,

pour les critiques positives, et 16% pour les critiques négatives (Hurst and Nigam, 2003, urst and Nigam, 2003).

Une étude comparative pour la classification des sentiments a été réalisée par Hang et al.(Cui et al., 2006, ui et al., 2006) afin d'analyser les produits en ligne en utilisant les classificateurs suivants : classificateur à algorithme passif-agressif (Shalev-Shwartz et al., 2004, halev-Shwartz et al., 2004), classificateur à modélisation de langage (Manning et al., 1999, anning et al., 1999)et le classificateur Winnow. Les résultats obtenus ont prouvé que le taux de précision de l'algorithme passif-agressif est la plus élevé (90,07%), comparé à ceux obtenus par les autres techniques.

En 2012, une analyse des sentiments liée aux critiques de restaurants a été menée par Kang et al.(Kang et al., 2012, ang et al., 2012) en utilisant un senti-lexicon. Deux versions améliorées de l'algorithme Naïve Bayes ont été aussi développées. Les performances de ces deux versions ont été ultérieurement évaluées en les comparant à celles des algorithmes originaux de Naïve Bayes et Support Vector Machine. L'efficacité importante de Naïve Bayes était claire d'après les bons résultats obtenus par les deux versions.

La classification des sentiments est souvent traitée par les techniques d'apprentissage automatique comme un problème de classification de texte reposant sur un sujet précis ou les mots tels que «sport», «politique», «science», etc. Tous les algorithmes de classification de texte comme Naïve Bayes, Support Vector Machine ou Maximum Entropy, etc peuvent être utilisé pour la classification des sentiments Mais pour la classification des sentiments, seuls les mots comme « génial », « bon », « meilleur », « parfait », etc. sont généralement considérés. Bien que les techniques d'apprentissage automatique sont largement utilisées et prouvent une grande performance, elles dépendent fortement des fonctionnalités définies manuellement. Pour cette raison, les techniques d'apprentissage en profondeur ont récemment attiré l'attention, car elles peuvent réduire l'effort de définition des caractéristiques et atteindre des performances relativement élevées (par exemple, la précision).
Le travail (Gutiérrez-Esparza et al., 2019, utiérrez-Esparza et al., 2019), intitulé «Classification des cas de cyber-agression » appliquant l'apprentissage automatique traite la détection de la cyberagression. Le corpus développé peut favoriser la recherche dans ce domaine, compte tenu de la rareté des ressources lexicales dans des langues différentes de l'anglais. Kim et Jeong (Kim and Jeong, 2019, im and Jeong, 2019)traitent du problème de la classification des sentiments textuels. Ils proposent un modèle CNN (Convolutional Neural Network), qui est un type d'apprentissage profond. Ce modele est composé d'une couche d'integration, de deux couches convolutives, d'une couche de regroupement et d'une couche entièrement connectée.
Ce modèle vise a compléter les vecteurs de phrases pour créer une taille fixe. Autrement dit, des phrases trop longues ont été coupées à une certaine longueur, et des phrases trop courtes ont été ajoutées au jeton. la longueur fixe S est defini comme étant la longueur maximale des phrases. Une couche d'incorporation qui mappe chaque mot d'une phrase à un vecteur d'entités E-dimensionnel génère une matrice S $\times$ E, où E désigne la taille d'incorporation. En d'autres termes, la couche d'intégration est un processus consistant à placer les mots reçus en entrée dans un espace sémantiquement bien conçu, où les mots avec des significations similaires sont situés à proximité et les mots avec des significations

opposées sont éloignés, les numérisant dans un vecteur. La première couche convolutive est utilisée pour identifier des informations contextuelles simples tout en se référant a la matrice S × E, et la deuxième couche convolutive est utilisée pour capturer les caractéristiques clés, puis les extraire (par exemple, pire, excellent) qui contiennent des sentiments affectant la classification.
Jabreel et Moreno (Jabreel and Moreno, 2019, abreel and Moreno, 2019) abordent le problème de la classification multi-classes des émotions basée sur des techniques de Deep Learning. L'approche la plus populaire pour ce problème est de le transformer en plusieurs problèmes de classification binaire, un pour chaque classe d'émotion. Cet article propose une nouvelle approche de transformation, appelée ensemble de paires xy, qui transforme le problème d'origine en un seul problème de classification binaire. Ce système se compose de trois modules : un module d'intégration qui utilise trois modèles d'intégration et une fonction d'attention, un module d'encodage basé sur les réseaux neuronaux récurrents (RNN) et un module de classification qui utilise deux couches de rétroaction avec la fonction d'activation ReLU suivie d'unité sigmoide.

1.3.2 Les méthodes basées sur des lexiques

Les techniques basées sur un dictionnaire reposent sur l'extraction de la polarité de chaque phrase d'un document. Le sens des mots d'opinion présents est, ensuite, analysé pour la classification des sentiments dans le texte. Les méthodes qui utilisent cette approche sont généralement basées sur des lexiques et emploient un dictionnaire de mots mappés sur leur valeur sémantique (Denecke, 2008, enecke, 2008). Dans ce sens, on peut définir le vocabulaire comme le lexique d'une langue précise. En fait, la version la plus connue du vocabulaire est WordNet (Miller, 1998, iller, 1998) qui représente un lexique sémantique formé des groupes de synonymes nommés Synsets.

Un autre exemple de lexique bien connu est SentiWordNet (Esuli and Sebastiani, 2006b, suli and Sebastiani, 2006b). C'est un lexique de sentiment représentant un index des mots de sentiment. Ce lexique comprend les informations de polarité du mot pertinent que se soit porteur d'un sentiment positif ou négatif. Il peut être considéré comme une extension de WordNet. Il assigne à WordNet Synsets une mesure graduée selon deux échelles : une échelle positive / négative et une échelle subjective / objective. Puisque le mot peut avoir diverses significations, la classification des sentiments est souvent basée sur des Synsets plutôt que des mots (Esuli and Sebastiani, 2006a, suli and Sebastiani, 2006a).

Malgré l'importance du lexique dans les systèmes TAL, les ressources disponibles ne sont pas assez nombreuses. Pour la langue anglaise, COMLEX Syntax (Cavnar et al., 1994, avnar et al., 1994) comprend une information détaillée contenant 38.000 mots ; le nombre des verbes dans cette information est 6.000. À partir de 191 classes sémantiques et 52 cadres syntaxiques, VerbNet présente 4.000 sens verbaux. Cependant, la langue française contient plusieurs lexiques. La majorité d'eux concernent la morphologie et pas la syntaxe.

En fait, le lexique LEFFF (Lexique des Formes Fléchies du Français) contient 5.000 verbes et 200.000 formes fléchies. Cependant, l'information associée est totalement flexion-

nelle. Comme nous l'avons précisé, le lexique-grammaire de Gross contient flexionnelle. Ce lexique a été numérisé par le Laboratoire d'Automatique Documentaire et Linguistique (LADL). Aujourd'hui, il est proportionnellement disponible sous une licence LGPL-LR. Tout ces caractéristiques rend la constitution d'une ressource lexicale appropriée au TAL plus facile.

Ha et al. (Ha et al., 2019, a et al., 2019) proposent une méthode de visualisation des sentiments dans les médias sociaux massifs. A cette fin, ils conçoivent un mécanisme de visualisation de réseau de sentiments à plusieurs niveaux basé sur des mots émotionnels dans le domaine de la critique de films. Ils proposent trois techniques de visualisation : une visualisation par carte thermique des mots sémantiques de chaque noeud, une carte de mise à l'échelle bidimensionnelle des données de mots sémantiques et une visualisation de constellation utilisant des images d'astérisme pour chaque cluster du réseau. Les visualisations proposées ont été utilisées comme système de recommandation qui suggèrent des films avec des émotions similaires à celles précédemment regardées. Cette nouvelle idée de recommander des contenus basés sur des schémas émotionnels similaires peut être appliquée à d'autres réseaux sociaux.
Mao et al .(Mao et al., 2019, ao et al., 2019) suggérer l'utilisation d'un mot incorporant des sentiments pour améliorer l'analyse émotionnelle. La méthode proposée construit une représentation hybride qui combine des embeddings de mots émotionnels basés sur un lexique émotionnel avec des incorporations sémantique de mot basé sur Word2Vec(Mikolov et al., 2013, ikolov et al., 2013). Ils utilisent le lexique émotionnel DUTIR, qui est une ressource d'ontologie chinoise rassemblée et étiquetée par l'Université de Dalian du Laboratoire de recherche d'informations technologiques(Chen, 2008, hen, 2008). Cette ressource annote les entrées de lexique avec un modèle de sept émotions (bonheur, confiance, colère, tristesse, peur, dégoût et surprise). L'évaluation de cette technique se réfère aux deux méthodes (combinaison directe et addition) pour construire la représentation hybride dans plusieurs ensembles de données.Les expérimentations prouvent que l'utilisation de vecteurs de mots hybrides est efficace pour la classification des émotions supervisée, améliorant considérablement la précision de la classification.
Dans (van den Broek-Altenburg and Atherly, 2019, an den Broek-Altenburg and Atherly, 2019), les auteurs vise à comprendre les sentiments des consommateurs envers les assurances maladie. Pour cela, ils ont exploité les discussions sur Twitter et les ont analysées en utilisant une approche basée sur un dictionnaire utilisant le Lexique des émotions du CNRC (Mohammad et al., 2013, ohammad et al., 2013), qui fournit à chaque mot sa polarité ainsi que son émotion associée (colère, anticipation, dégoût, peur, joie, tristesse, surprise et confiance). La principale conclusion de cette étude est que les consommateurs s'inquiètent des réseaux de fournisseurs, des avantages des médicaments sur ordonnance et des préférences politiques. De plus, les consommateurs font confiance aux prestataires médicaux mais craignent les événements imprévus. Ces résultats suggèrent que des recherches supplémentaires sont nécessaires pour comprendre l'origine des sentiments qui motivent les consommateurs afin que les assureurs puissent offrir de meilleurs régimes d'assurance.
L'analyse des sentiments peut également être calculée à l'aide du analyse des émoticônes (Wang and Castanon, 2015, ang and Castanon, 2015). Les émoticones ont aussi signification textuelle exacte où l'analyse des sentiments peut être fait sur la base de ces significations textuelles. Ces données seront téléchargées à partir de divers réseaux sociaux médias

(Wolny, 2016b, olny, 2016b),(Wolny, 2016a, olny, 2016a),(Shiha and Ayvaz, 2017, hiha and Ayvaz, 2017) et les émoticônes peuvent être traités de la même manière que les textes. De plus, le les emojis seront également considérés comme des émoticônes où le les émoticônes et emojis seront classés au cours du processus en fonction des polarités de sentiment. La classification binaire fait l'objet des approches d'analyse de sentiments. Toutefois, l'extraction des opinions floues constitue le centre d'intéret des travaux de recherche. En fait, la signification du mot sentiment varie selon les domaines d'application. Comme dans les phrases interrogatives et conditionnelles, les mots de sentiment ne peuvent exprimer aucune émotion et les phrases présentant objectivement des faits n'expriment aucun sentiment. Dans la section suivante nous mettons l'accent sur la litérature d'approches floues pour l'analyse des sentiments.

1.4 Approches floues pour l'analyse des sentiments

Les systèmes basés sur la logique floue peuvent faire face à l'imprécision et à l'ambiguité (Zadeh, 2015, adeh, 2015). Une contribution importante de la logique floue est la technique de calcul avec des mots, c'est-à-dire que les mots peuvent être transformés en valeurs numériques pour un calcul ultérieur. La logique floue nous offre une manière souhaitable de traiter les problèmes linguistiques (Ross, 2010). Dans (Dragoni et al., 2014, ragoni et al., 2014) les auteurs présentent un système dont les objectifs sont la mise en oeuvre d'une approche d'apprentissage capable de modéliser des fonctions floues utilisées pour construire le graphe de relations représentant l'adéquation entre les concepts de sentiments et différents domaines et le développement d'une ressource sémantique basée sur la connexion entre une version étendue de WordNet, SenticNet et ConceptNet, qui a été utilisée à la fois pour l'extraction de concepts et pour classer les phrases dans des domaines spécifiques.
Dans le papier de Saoud et al.(Saoud et al., 2014, aoud et al., 2014), la notion de sévérité est proposée pour moins pénaliser les utilisateurs crédibles mais hélas assez éloignés de l'opinion majoritaire. L'approche proposée utilise l'algorithme de clustering flou (fuzzy cmeans).
Dans un autre travail récent (Montoro et al., 2018, ontoro et al., 2018), une liste nommée : Affective Norms for English Words (ANEW) qui est un ensemble de mots anglais avec des mesures d'émotion : valence, excitation et dominance pour chaque terme est utilisée pour construire une classification modèle. Ce modèle basé sur flou est construit en utilisant le clustering k-means, l'analyse en composantes principales (ACP) et la fonction d'appartenance trapézoïdale floue et enfin les données textuelles de Twitter sont classées en cinq catégories d'opinions floues (très négative, négative, neutre, positive et très positive).
Le système basé sur des règles floues de Tsukamoto a été utilisé dans (Liu and Cocea, 2017, iu and Cocea, 2017) (Jefferson et al., 2017, efferson et al., 2017) pour l'analyse des sentiments. L'attribut d'entrée de ce système utilise la fonction d'appartenance floue trapézoïdale pour convertir les valeurs numériques en termes linguistiques flous. Ce système fournit deux sorties : une sortie double avec des valeurs à la fois pour la classe positive et

négative et une sortie indiquant différentes intensités de sentiment (Jefferson et al., 2017, efferson et al., 2017).
Siddiqua et al. a intégré un classificateur basé sur des règles basé sur des émoticônes et des mots porteurs de sentiments avec un classificateur Naïve Bayes supervisé pour classer les sentiments des tweets. Ce classifieur Naïve Bayes est formé à l'aide de plusieurs lexiques de sentiments (Siddiqua et al., 2016, iddiqua et al., 2016).
En 1975, les travaux influents de Mamdani et Assilian (Mamdani and Assilian, 1975, amdani and Assilian, 1975) ont introduit le premier contrôleur basé sur des règles alimenté par un mécanisme d'inférence floue. Un tel système est généralement appelé système basé sur des règles floues. Inspiré par le Mamdani , les auteurs dans (Vashishtha and Susan, 2019, ashishtha and Susan, 2019) ont développé un système de classification des sentiments non supervisé basé sur des règles floues en utilisant le système les règles de mamdani. Une approche basée sur la logique floue développée par (Vashishtha and Susan, 2018, ashishtha and Susan, 2018) trace les sautes d'humeur dynamiques des tweets au fil du temps. Cette approche analyse les tweets des fans de cricket en déterminant la polarité des tweets et en traçant leur humeur en fonction du temps.

1.5 Mesures de performance

L'efficience de classification est souvent mesurée en utilisant les paramètres classiques spécifiques au domaine de la recherche d'information : la précision et le rappel. La précision mesure aussi l'exactitude d'un classificateur en mettant l'accent sur l'exactitude d'une classe spécifique prédite :

$$Précision = \frac{Nombred'identifiantsassignéscorrectementàleursclasses}{Nombretotald'identifiant} \tag{1.6}$$

En fait, la précision est le rapport du nombre de documents pertinents trouvés au nombre total de documents sélectionnés. Elle est prise en considération pour mesurer le bruit. Plus sa valeur est proche de 100%, moins de bruit sera produit. Ceci résulte en une meilleure réponse. Pour ce qui concerne le rappel, il est défini comme suit :

$$Rappel = \frac{lenombred'itemspertinentsretrouvés}{touslesitemspertinentssélectionnables} \tag{1.7}$$

Cela indique qu'une décision était prise si un document quelconque dx a été classé sous c_i. Le rappel est le rapport du nombre de documents pertinents trouvés au nombre total de documents pertinents. Plus la valeur du rappel est proche de 100%, plus de bruit est produit et plus la réponse est satisfaisante. Le rappel est communément appelé sensibilité. Il consiste à mesurer la capacité d'un modèle de prédiction à sélectionner des instances d'une certaine classe à partir d'un ensemble de données. Il représente le nombre des classifications correctes pénalisées par le nombre des éléments manqués

La précision et le rappel représentent respectivement le "degré de solidité" du classificateur et son « degré d'exhaustivité ». La pertinence a été souvent évaluée dans les

recherches documentaires pour quantifier le rappel et la précision en utilisant les indices de bruit et de silence. Le bruit désigne le rapport du nombre de documents non pertinents ramenés en réponse au nombre total de documents.
Par conséquent,

$Bruit = 1 - Précision$ ´ Le silence représente le taux entre le nombre de documents pertinents qui n'apparaissent pas dans le résultat de la recherche et le nombre total de documents. Par conséquent,
$Silence = 1 - Rappel$

En se basant sur la définition mentionnée au dessus, la précision et le rappel sont considérés en tant que des probabilités subjectives, qui savoir l'intensité de l'attente de l'utilisateur d'un système qui se comporte correctement lors de la classification d'un document inconnu pour une catégorie ci.

De plus, on peut obtenir des niveaux plus élevés de précision au prix de faibles valeurs de rappel. De ce fait, l'évaluation de la classification doit être réalisée en combinant des valeurs de précision et de rappel. Dans ce contexte, de nombreuses mesures ont été introduites. Les plus fréquentes sont citées au dessous.

- seuil de rentabilité : Le seuil de rentabilité est la valeur à laquelle la précision est égale au rappel. En fait, ce résultat est fourni par le tracé de la précision en fonction du rappel pour différents seuils ; le seuil de rentabilité représente la valeur de la précision ou de rappel pour lesquelles la courbe coupe la ligne precision = rappel. Cette idée est basée sur le fait que le rappel accroît toujours de 0 à 1 et la précision amoindrit monotonement d'une valeur proche de 1 en diminuant le seuil de 1 à 0.
- function F-mesure (Fscore) :La F mesure est une mesure d'efficacité dérivée. C'est une moyenne pondérée de la précision et du rappel. La meilleure valeur de F-mesure est 1 et la pire valeur est 0. Elle est calculée comme suit :

$$F - measure = \frac{2 * Précision * Rappel}{Précision + Rappel} \tag{1.8}$$

1.6 Conclusion

La notion des sentiments et opinions dans les réseaux sociaux a été introduite dans ce chapitre. Les approches principales d'analyse des sentiments ont été aussi présentées ultérieurement. De plus, nous avons mis l'accent sur les mesures de performance des approches d'analyse des sentiments. Dans le chapitre suivant, nous allons décrire les différentes techniques de détection des communautés dans les réseaux sociaux.

Chapitre 2

Détection de communautés dans les réseaux sociaux

2.1 Introduction

L'augmentation du nombre des postes qui expriment des points de vue provoque un accroissement du volume des données créées. En fait, la volumétrie des données est la cause majeure de l'application des algorithmiques relatives aux médias sociaux. L'analyse des réseaux sociaux est compliquée en raison de leur grande taille. De nombreuses techniques, telles que le partitionnement des réseaux sociaux en éléments plus petits nommés communautés, ont été efficacement employées pour résoudre ce problème. En employant ce cloisonnement, il est donc possible de comprendre le comportement général du système avec l'analyse locale des éléments qui le forment ainsi que leurs comportements.

Nous exposons, dans la section 2.2 la représentation usuelle des réseaux sociaux, notamment le graphe. Ensuite, un état de l'art des principales techniques de détection des communautés dans les réseaux sociaux seront décrites dans la section 2.3. Dans la section 2.4 nous mettons l'accent sur les critéres d'évaluation des algorithmes de détection de communautés.

2.2 Théorie des graphes et modélisation des réseaux sociaux

Dans cette section, nous présentons les notions fondamentales de la théorie des graphes appliqués pour la représentation et la manipulation des réseaux sociaux. Un réseau social

peut être défini comme celui établi par des interactions sociales (Dutta et al., 2015, utta et al., 2015). Dans ce réseau, les individus et leurs interactions sont représentés respectivement par des noeuds et des arêtes.

2.2.1 Définitions

Nous considérons un graphe $G = (V, E)$ où V désigne l'ensemble des sommets et $E \subseteq V \times V$ représente l'ensemble des arêtes. On étudiera, dans ce travail de recherche, les graphes non-orientés qui montrent une relation symétrique entre les sommets qui correspondent aux objets liés dans le graphe. On parle aussi dse noeuds ou des acteurs. Soit N le nombre des sommets de G avec $N = |V|$.

On parle de deux sommets v et $v^{'}$ si ces derniers représentent les extrémités d'une même arête du graphe, autrement dit si $(v, v^{'}) \in E$.

Les relations entre les sommets du graphe sont décrites par les arêtes. Chaque arête unie deux sommets (éventuellement confondus) du graphe. Il est possible de valuer les arêtes en leur donnant une valeur. La valuation élevée montre ainsi une relation ayant une forte intensité. On affirme que le graphe est valué. Dans le cas contraire, uniquement les graphes ne contiennent que des valuations positives sur leurs arêtes sont considérés. Si le sommet établit une (ou deux) de ses extrémités, on peut dire que une arête est incidente au sommet.

La matrice d'adjacence A du graphe G représente la matrice carrée du côté $|V|$ où le terme A $[v, v']$ réfère à la valuation, notée simplement Av,v', de l'arête éventuelle reliant les sommets v et $v^{'}$, ou 0 si v et $v^{'}$ ne sont pas adjacents. M est la somme des valuations des arêtes de G :

$$M = \sum_{(v,v' \in V * V)} A_{v,v'} \tag{2.1}$$

Le degré deg(v) d'un sommet $v \in V$ correspond au nombre des arêtes adjacentes à v.
On préfère généralement, dans un graphe valué, d'employer le degré valué qui considère la valuation des arêtes :

$$K(v) = \sum_{v' \in V} A_{v,v'} \tag{2.2}$$

Une arête liant un sommet v avec lui-même est nommée boucle. Par conséquence, l'influence de la valuation de l'arête est du à ses deux extrémités. La contribution de l'arête au degré du sommet se multiplie dans le degré de v.

Un graphe est considéré complet dans le cas où tous ses sommets sont adjacents deux à deux : $\forall v, v^{'} \in V * V, v, v^{'} \in E$

Un sous-graphe $G' = (V^{'}, E^{'})$ de G, où $V^{'} \subset V$, $E^{'} \subset E$ contient des sommets de $V^{'}$ ainsi des arêtes de E qui ont deux extrémités dans $V^{'}$.

Une clique $G' = (V^{'}, E^{'})$ représente un sous-graphe de G dans lequel tous les couples de sommets de $V^{'}$ sont liés avec une arête. Une clique est alors un sous-graphe complet de G.

Un graphe biparti peut être considéré comme un graphe ayant un ensemble de sommets divisé en deux sous-ensembles disjoints, V_1 et V_2, où chaque arête relie un sommet de V_1 à un sommet de V_2.

Une composante connexe est constituée d'un ensemble optimal des sommets entre chaque couple d'eux (v_1, v_n), il y'a un chemin ou une succession de sommets $v_2, v_3, ..., v_{n-1}$ de V avec $(v_i, v_{i+1}) \in E$, $\forall\ i = 1, ..., n-1$.

Un graphe est considéré comme non-orienté en cas où $\forall (v, v') \in E, (v', v) \in E$. Autrement dit, si les arêtes sont formées à partir des paires de sommets non ordonnées. En cas où les arêtes ayant la forme de couples de sommets sont caractérisées par une origine et une destination, le graphe est orienté.

2.2.2 Distances dans un graphe

Le concept de distance entre objets a une grande importance dans toutes les étapes de classement. Les mesures appliquées aux sommets d'un graphe vont être représentées dans cette sous-section.

La longueur de plus courte distance séparant deux sommets v et v' de V, dans un graphe non-valué, est le nombre des arêtes qu'on doit traverser au minimum afin de joindre v et v'. Elle est nommée distance géodésique entre v et v'.

L'intervalle de la plus petite distance, dans un graphe valué, est la somme minimale des valuations des arêtes essentielles pour joindre les sommets v et v'.

En termes de calcul, cette distance est moins coûteuse puisqu'il y a aucune valuation positives sur chaque arête. En outre, des cycles infinis ayant un coût négatif peuvent apparaître dans cette structure. Pratiquement, on pourra ainsi limiter les distances minimales. En revanche, on peut borner la distance maximum séparant deux sommets si la liaison de deux sommets est impossible(Combe, 2013, ombe, 2013).

2.2.3 Mesures de centralité

Le terme indicateur réfère à toute mesure ayant une nature quantitative. Le calculé de centralité est fait à partir des sommets, des arêtes, des assemblages de sommets ou d'arêtes ou aussi du graphe lui-même. Comme exemples de ces indicateurs, on peut citer les mesures de centralité utilisées pour l'évaluation des propriétés généralement abstraites des entités constituant un réseau social. On peut également mentionner la centralité de proximité, de prestige, de pouvoir et de cohésion, etc. (Combe, 2013, ombe, 2013).

Plusieurs auteurs ont discuté les centralités sans donner une définition consensuelle. Pourtant, il est possible d'étudier la nomenclature de Koschutzki et al. qui ont introduit une typologie de ces mesures en considérant les axes décrits au dessous (Koschützki et al., 2005, oschützki et al., 2005) :

— l'accessibilité basée sur un concept de distance séparant les sommets (degré, excentricité, proximité) ;
— l'écoulement basé sur le concept de flux qui circule entre les sommets du graphe. Nous donnons, comme exemples, la centralité d'intermédiarité et les mesures qui utilisent une marche aléatoire.
— la vitalité montrant l'importance d'un sommet ou d'une arête dans un graphe en calculant la différence entre f(G) et f(G $\setminus v_x$), où la function f() correspond à la mesure quantitative qui caractérise G et G $\setminus v_x$ dénote le graphe G privé du sommet v_x.
— la réaction dont le score d'un sommet dépend des scores d'autres sommets dans le réseau, tel qu'il est le cas dans l'indice de Katz.

2.2.4 Centralité de degré

Le degré est la mesure plus simple de la centralité. En fait, la somme de l'intensité de la connexion d'un sommet est généralement mesurée par ses voisins directs.

$$C_D(v) = \sum_{v \in V} A_{v,v'} \tag{2.3}$$

Nieminen a suggéré une version normalisée pour laquelle le score est égal à 1 pour les sommets connectés à tous les autres sommets.

$$C_D^{'} = \frac{deg(v)}{|V| - 1} \tag{2.4}$$

Cette mesure est employée dans les conditions où l'importance d'un sommet est assimilée à son activité potentielle de communication.

2.2.5 Centralité d'intermédiarité

C'est une mesure de centralité d'un sommet dans un graphe. En fait, l'intermédiarité d'un sommet u ∈ V est obtenue comme suit :

$$C_B(u) = \sum_{v,v' \in V, v \neq v'} \frac{\phi(v, v^{'} \setminus u)}{\phi(v, v^{'})} \tag{2.5}$$

où ($\phi(v, v^{'})$ représente le nombre des plus courtes distance entre le sommet v le sommet $v^{'}$ et $\phi(v, v'\backslash u)$ correspond au nombre des plus petites distances séparant le sommet v du sommet $v^{'}$ en passant par u. L'intermédiarité des sommets localisés sur les plus courts chemins séparant deux autres sommets est plus grande que celles des autres sommets (Brandes, 2008, randes, 2008).

Nous pouvons définir l'intermédiarité aussi pour une arête e :

$$C_{EB}(e) = \sum_{v,v^{'} \in V, v \neq v^{'}} \frac{\phi(v, v^{'}\backslash e)}{\phi(v, v^{'})} \tag{2.6}$$

où $\phi E(v, v^{'}\backslash e)$ désigne le nombre des plus courtes distances séparant le sommet v du sommet $v^{'}$ en passant par l'arête e.

2.2.6 Centralité de proximité

Concernant les graphes connexes, la centralité de proximité peut être définie comme l'inverse de la distance moyenne à tous les autres sommets. En fait, la centralité de proximité des sommets, qui sont à distance limité de tous les autres sommets, est plus grande.

$$Cc(v) = \frac{1}{\sum_{v' \in V \backslash v} dist(v, v')} \tag{2.7}$$

où $dist(v, v^{'})$ dénote une distance, telle que le nombre d'arêtes dans le plus court chemin entre deux sommets ou la somme des valuations de ces arêtes pour les graphes valués.

2.3 Algorithmes de detection de communautés

Nous décrivons, dans cette section, les algorithmes de détection des communautés.

2.3.1 Formalisation

On considère un graphe G = (V, E). Ensuite, on cherche une partition P de V de telle façon que chaque classe C de P contienne les deux extrémités de plusieurs arêtes Cependant, les arêtes possédant des extrémités dans deux catégories diverses ne sont pas assez nombreuses.

Soit un ensemble d'éléments V = $\{v1, ..., vn\}$ est décrit par leur représentation, le but sera donc de définir une partition P = $\{C1, ..., Cr\}$ de V en r classes de telle sorte que les éléments montrés dans la même classe soient proches à l'égard de leur représentation et d'un critère antérieurement sélectionné pendant que des éléments différents soient attribués à des classes diverses.

2.3.2 Les strategies de partitionnement

Nous décrivons, dans cette sous-section, les diverses stratégies utilisées par les algorithmes de détection des communautés(Creusefond, 2017, reusefond, 2017). Les techniques de partitionnement des données sont inspirées du problème éponyme. Dans ce problème, on vise à grouper les objets génériques. Ce domaine a pour stratégie spécifique d'introduire une fonction de similarité entre les objets. Par la suite, un algorithme est exécuté pour trouver des « clusters » où les similarités internes sont robustes. Des méthodes de partitionnement hiérarchique ont été aussi introduites (les communautés fusionnent ou se divisent durant l'exécution) et des techniques de coupure de graphe ont été proposées (en réduisant le nombre des arêtes entre les parties). Une autre approche consiste à intégrer des graphes dans un espace métrique en utilisant une mesure de dissimilarité entre les noeuds .

Une hiérarchie peut être définie comme une famille parfaitement ordonnée de partitions H = $\{P1, ..., Pn\}$ où P1 représente la partition discrète, PN désigne la partition grossière et, pour i =1,..., N-1, Pi est plus fine que Pi+1 en comparant les partitions.

Selon le résultat obtenu, les méthodes de classification non supervisées peuvent être classifiées en deux types : les techniques hiérarchiques et non-hiérarchiques. Celles appartenant à la première classe peuvent être : i) ascendantes, en cas où elles aboutissent par agglomérations successives à la partition grossière, ou ii) décroissantes si elles procèdent par diviser la partition grossière jusqu'à l'obtention d'une partition discrète. La deuxième classe enferme les méthodes qui sont capables de produire immédiatement une partition. Les méthodes non-hiérarchiques sont généralement itératives. En fait, leur application exige la pré-connaissance du nombre de classes à fournir.

Les méthodes divisives sont les techniques de partitionnement hiérarchique. Elles sont basées sur la spécification et la suppression consécutives des arêtes supposées être entre les « clusters ». Le graphe est déconnecté par ces suppressions et les composantes connexes représentent les communautés.

La hiérarchique ascendante est une technique de classification utilisée pour le regroupement des classes les plus proches à partir de la partition discrète, en employant une distance séparant les éléments et une fonction nommée mesure d'agrégation qui permet la comparaison des groupes d'éléments. Pour ce qui concerne la classification hiérarchique ascendante, le fait de choisir la distance est à la discrétion de l'utilisateur. Elle dépend de la nature de la représentation des éléments. Pour les vecteurs numériques, on emploie généralement la distance euclidienne, tandis que dans le cas des documents décrits par

des sacs de mots, on considère la distance du cosinus. En outre, cette technique nécessite l'utilisation d'un critère d'agrégation. Au fil du temps, de nombreux critères d'agrégation ont été suggérés.

Le lien minimum est une mesure d'agrégation basée sur l'association d'un minimum de distances entre les paires d'éléments formées par un élément de chaque classe à deux classes C_k et C_l. Cependant, le lien maximum repose sur l'association du maximum de ces distances à ces classes. La mesure initiale s'agit d'agréger les deux classes qui possèdent les deux éléments les plus proches. La deuxième mesure associe les deux classes entre lesquelles les deux éléments les plus éloignés sont les plus proches.

$$S_{min}(C_k, C_l) = min_{v \in C_k, v' \in C_l} d(v, v')$$
$$S_{max}(C_k, C_l) = max_{v \in C_k, v' \in C_l} d(v, v')$$

Le lien moyen peut être défini comme la mesure d'agrégation qui applique la moyenne arithmétique des distances :

$$S_{moy}(C_k, C_l) = \frac{1}{|C_k| \cdot |C_l|} \cdot \sum_{v \in C_k} \sum_{v' \in C_l} d(v, v')$$

La mesure de Ward, également nommée construction hiérarchique du moment d'ordre deux, est obtenue comme suit :

$$S_{ward}(C_k, C_l) = \frac{m_k . m_l}{m_k + m_l} . d(gC_k, gC_l)$$

où m_k et m_l correspondent aux masses des deux classes ou le nombre des éléments qu'elles contiennent.

Cette mesure peut être interprétée en matière d'optimisation d'inertie. En fait, elle résulte en l'optimisation de l'inertie inter-classes. Pour une partition précise, il y a moins de possibilités pour sélectionner deux classes à fusionner ($Nx(N-1)$) de telle façon que l'une des classes puisse être divisée en deux.

Newman et Girvan ont présenté la modularité (Newman and Girvan, 2004, ewman and Girvan, 2004) inspirant une série de techniques fondées sur celle-ci. Brandes et al.(Brandes, 2008, randes, 2008) ont considéré la partition optimale en modularité comme un problème NP-Complet. Par conséquent, le passage à l'échelle devient extrêmement coûteux. Les approches d'optimisation gloutonne, telles que l'algorithme de Louvain (Blondel et al., 2008, londel et al., 2008), la méthode du recuit simulé (Guimera et al., 2004, uimera et al., 2004) ou l'analyse spectrale (Mitrović and Tadić, 2009, itrović and Tadić, 2009), ont été proposées afin d'aborder une alternative acceptable en temps de calcul.

En fait, plusieurs algorithmes reposent sur des techniques spectrales. Dans ces méthodes, le spectre d'une matrice précise, définissant une notion proximale entre les noeuds y, est étudié. Les vecteurs propres liés aux valeurs propres les plus faibles (sauf le premier, à valeur propre nulle) représentent des « clusters » à forte similarité interne (Mitrović and Tadić, 2009, itrović and Tadić, 2009). La considération des k premiers vecteurs propres (en ignorant le premier) rend la projection des n noeuds dans un espace k-dimensionnel possible. Dans ce cas, conséquence, l'exécution d'un partitionnement classique (type k-means) est suffisante pour trouver des « clusters ».

Généralement, la matrice Laplacienne est utilisée comme matrice de similarité entre les noeuds. En revanche, on peut employer autres matrices, telles que la matrice de modularité où la proximité réfère au profit en modularité fourni par la réunification des noeuds voisins dans le même « cluster » (Newman and Girvan, 2004, ewman and Girvan, 2004).

Le processus qui se déroule sur le graphe est simulé par les techniques dynamiques. Le modèle de Potts (Wu, 1982, u, 1982) représente un ensemble de particules, caractérisées par un état, et une notion de proximité avec les autres particules. Les particules qui s'unissent entre elles. Les proches particules partagent le même état. Il est donc nécessaire de connaitre les paramètres du modèle qui correspondent le mieux aux données. Une autre procédure est la synchronisation où tous ses éléments sont unifiés progressivement au même état par le système simulé. Les noeuds d'une communauté proches sont localement synchronisés, ce qui autorise leur identification(Arenas et al., 2006, renas et al., 2006). D'autre part, les marches aléatoires sont plus nombreuses dans les sous-graphes denses(Su et al., 2017, u et al., 2017).

Les techniques d'inférence statistique sont basées sur l'hypothèse que le graphe a été construit en appliquant un modèle et que ce modèle montre l'appartenance des noeuds aux diverses communautés en tant que paramètre. Il consiste donc de connaitre la les paramètres du modèle qui produisent les données observées avec la probabilité la plus importante. On suppose, par exemple, que les noeuds, à l'intérieur des communautés, sont liés avec une probabilité p_{in}, et, à l'extérieur, ils sont connectés par une probabilité p_{in} (Hastings, 2006, astings, 2006). Le but, ici, est de savoir la partition qui aurait le plus de chances de construire le graphe obtenu à partir de ce modèle.

2.3.3 Algorithmes de détection de communautés dans les graphes

Les algorithmes de détection de communautés dans les graphes ou dans les réseaux sociaux sont souvent appliqués afin de former une partition des sommets en considérant les liaisons entre les sommets dans le graphe de façon que les communautés soient composées de sommets strictement connectés et peu reliées entre elles (Yang et al., 2016, ang et al., 2016).

Les techniques de détection de communautés les plus importantes sont celles qui améliorent une fonction de qualité pour l'évaluation de la qualité d'une partition précise, comme la modularité, la coupe ratio, la coupe min-max ou la coupe normalisée, les techniques hiérarchiques, telles que les algorithmes de division et les techniques spectrales ou l'algorithme de Markov et ses extensions (Dutta et al., 2015, utta et al., 2015)(Clauset et al., 2008, lauset et al., 2008)(Combe, 2013, ombe, 2013).

Ces méthodes de partitionnement des graphes sont considérablement efficaces pour la détection des composantes strictement connectées dans un graphe. La détection de telles communautés est décrite dans cette section. Nous exposons aussi les techniques de classification hiérarchique. Nous focalisons sur les techniques utilisées pour optimiser un

critère de qualité de la partition, particulièrement la modularité.

Modularité

La classification des sommets d'un graphe est plus récente que la classification non supervisée des données vectorielles. Puisque les données considérées sont différentes, aucune équivalence existe entre deux problèmes, ce qui permet l'utiliser toutes les techniques efficients sur les vecteurs vers les graphes et vice versa.

Fortunato a développé un panorama large, néanmoins défini à partir des solutions qui y sont apportées (Fortunato and Hric, 2016, ortunato and Hric, 2016). Porter et al., ont réalisé eux aussi une étude approfondie (Porter et al., 2009, orter et al., 2009). Dans plusieurs travaux, la classification des sommets dans un graphe a été traitée comme un problème d'optimisation(Newman, 2016, ewman, 2016). Une fonction associée à une partition des sommets du graphe fournit, de ce fait, un score mesurant la qualité de la partition. L'un des critères employés est celui de la modularité(Newman and Girvan, 2004, ewman and Girvan, 2004).

Nous décrivons maintenant le critère de modularité Q_{GN} comme défini par Newman et Girvan (Newman and Girvan, 2004, ewman and Girvan, 2004). En fait, il est plus judicieux de partitionner un graphe d'une façon ou d'une autre.

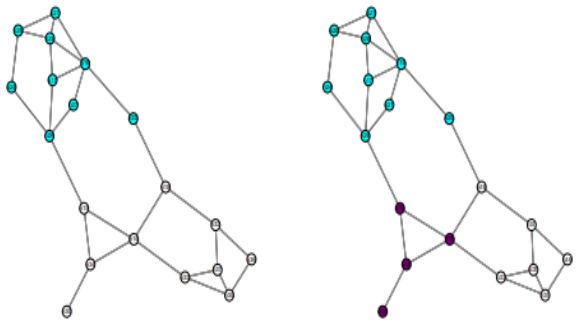


FIGURE 2.1: Pourquoi la partition (a) est-elle la plus mauvaise ?

La modularité Q_{GN}, développée par Newman et Girvan (Newman and Girvan, 2004, ewman and Girvan, 2004) (Newman, 2006b, ewman, 2006b) peut être appliquée pour répondre à cette question. Elle s'agit d'une mesure employée pour caractériser l'efficacité d'une partition des sommets d'un graphe vis-à-vis à la densité des liens dans des groupes et du nombre de liens entre ces eux ainsi que au regard du degré des sommets considéré. Cette modularité est basée sur la comparaison de la proportion des arêtes du réseau reliant des sommets issus de la même communauté à celle attendue dans un graphe équivalent en cas où les sommets sont caractérisés par la même distribution des degrés et les arêtes sont placées aléatoirement entre tous les sommets. Q_{GN} est calculée par :

$$Q_{GN} = \frac{1}{2M} \sum_{v,v'} [A_{v,v'} - \frac{k(v).k(v^{'})}{2M} .\delta(c(v).c(v))] \tag{2.8}$$

où $v, v^{'}$ traverse toutes les paires de sommets de VxV (de manière que les boucles soient aussi considérées). M représente la somme des valuations des arêtes, A correspond à la matrice d'adjacence, $c(v)$ est la classe du sommet v et δ désigne la fonction de Kronecker qui vaut 1 en cas où ses arguments sont égaux et 0 sinon.

La valeur de modularité varie entre -1 (si toutes les arêtes sont intercommunautaires) et 1 (si toutes les arêtes sont intracommunautaires). La modularité vaut 1 dans un graphe partitionné de manière à ne pas avoir d'arête placée entre 2 sommets de communautés différentes (mais au moins une dans une communauté). La modularité vaut zéro pour la partition grossière. Elle représente également la valeur vers laquelle tend la modularité d'une partition aléatoire. Comme montré par Newman et al.(Newman, 2006b, ewman, 2006b), les valeurs de modularité pour des réseaux avec des structures de communauté varient souvent entre 0,3 et 0,7. Des valeurs élevées sont parfois obtenues. En cas où le nombre des arêtes intra-groupes (connectant deux arêtes qui appartiennent au même groupe) est inférieur à celui des arêtes attendues dans un graphe qui fait l'objet d'une distribution aléatoire, la modularité QNG sera donc négative.

La modularité est, enfin, indéfinie sur un graphe dans lequel n'existe aucun lien (graphe vide). Les fondateurs de la notion de modularité ont proposé d'utiliser plusieurs variations du graphe aléatoire, nommé modèle nul. Malgré cela, le graphe basé sur la distribution des degrés a été vite utilisé dans les travaux de recherche. Nicosia et al. (Nicosia, 2009, icosia, 2009) ont formalisé, ensuite, une variante de la modularité pour les graphes orientés. Cette variante est basée sur l'application d'une matrice d'adjacence non-symétrique ainsi que sur la considération du sens des connexions pendant le calcul du produit des degrés des sommets. En cas où elle influence considérablement l'étude de la détection de communautés dans les graphes, la modularité a des défauts. Elle présente également une limite de résolution qui affaiblit la possibilité d'obtenir des petites communautés bien définies (?, ?).

Pour dépasser cette limite de résolution, plusieurs variantes ont été proposées, particulièrement en employant un paramètre pour obtenir des petites ou des grandes classes. Il a été prouvé par Lancichinetti et Fortunato qu'en appliquant une extension de la modularité montrant un facteur de variation de la taille des communautés à construire, la résolution de quelques configurations n'est pas bien faite (Lancichinetti and Fortunato, 2011, ancichinetti and Fortunato, 2011). De leur part, Gautier et al. ont suggéré de résoudre ce problème en utilisant un facteur qui limite le nombre de classes à produire (Krings and Blondel, 2011, rings and Blondel, 2011).

Montgolfier et al. ont illustré que des graphes particuliers (tores, hypercubes) n'ayant pas a priori de structure de communauté ont, pourtant, des partitions à modularité forte (De Montgolfier et al., 2011, e Montgolfier et al., 2011).

Dans la technique de Louvain, reposant sur la modularité, Blondel et al.(Blondel et al., 2008, londel et al., 2008) ont spécifié le problème posé en faisant référence à l'exemple du « Collier de perles » (Aynaud et al., 2011, ynaud et al., 2011). En fait, dans cette méthode, une boucle de cliques connectées par des ponts forms (vérifiez ce mot) d'une arête, au lieu d'être considérées comme des communautés distinctes, sont finalement agglomérées entre elles. Le regroupement de ces communautés ne peut être effectué seulement si leur taille est inférieure à $\sqrt{2M}$.

Ye et al. ont mentionné un inconvénient de l'emploie du rapport entre le nombre des arêtes internes et externes aux communautés (Zhen-Qing et al., 2012, hen-Qing et al., 2012). Par conséquent, les deux communautés proposées ont la même contribution à la modularité (elles ont le même nombre d'arêtes internes) malgré que la communauté située à gauche soit évidemment meilleure. Plusieurs autre études ont focalisé sur la paramétrisation de la résolution. Par exemple, Arenas et al(Arenas et al., 2008, renas et al., 2008). ont proposé d'adjoindre une boucle sur tous les sommets et d'ajuster sa valuation pour obtenir une résistance qui empêche le regroupement des petites communautés. Reichardt et al. ont paramétrisé également un critère de modularité. Cette paramétrisation concerne le calcul du graphe aléatoire (Reichardt and Bornholdt, 2006, eichardt and Bornholdt, 2006).

Les algorithmes de classification hiérarchique

Les réseaux sociaux sont caractérisés par leur capacité de présenter une structure hiérarchique avec des communautés plus petites incluses dans des communautés plus grandes. Ceci fait justifie un groupe spécial d'algorithmes appelés algorithmes de détection de communauté hiérarchique qui peuvent révéler la hiérarchie des communautés dans les réseaux sociaux. Les algorithmes de classification hiérarchique sont utilisés afin de regrouper les noeuds d'un réseau dans des communautés diverses de manière que les noeuds d'une même communauté soient semblables le plus possible, alors que ceux appartenant à des communautés différentes soient distinctes les uns des autres. Nous pouvons ainsi distinguer les algorithmes ascendants (algorithmes agglomératifs) et les algorithmes descendants (algorithmes divisifs)(Xi et al., 2016, i et al., 2016).

- **Algorithmes agglomératifs**

Au début de l'application des algorithmes agglomératifs, chaque noeud du graphe représente une communauté. En fin, nous avons initialement n communautés (où n est le nombre de noeuds).Ensuite, les distances séparant les communautés sont calculées. Les deux communautés les plus proches sont ultérieurement fusionnées pour construire une nouvelle communauté. Toutes les distances entre les communautés sont calculées et deux communautés sont fusionnées à chaque étape. Le processus de calcul s'arrête s'il y a une seule communauté représentant le tout graphe.

L'algorithme Walktrap (Pons and Latapy, 2006, ons and Latapy, 2006) est un d'algorithme agglomératif qui emploie les caractéristiques des marches aléatoires afin de définir une notion de distance entre les noeuds. Une partition $P_i = \{\{v_i\},\ v_i \in V,\ i = 1...n\}$

du graphe G, représentant n communautés, est considérée dans cet algorithme. En fait, chaque communauté comprend seulement un noeud. Toutes les distances séparant les n communautés sont initialement calculées. Subséquemment, les étapes suivantes sont répétées jusqu'à la création d'une seule communauté :

— Etape 1 : consiste à choisir les deux communautés C_1 et C_2 appartenant à la partition Pk et séparées par une distance minimale. En fait, cette distance mesure un poids $P_i(i,j)$, représentant la probabilité qu'une marche aléatoire aille de i à j en t étapes, pour chaque paire de sommets (i,j). La distance sur les paires de noeuds est calculée comme suit :

$$dist_t(i,j) = \sqrt{\sum_{v \in V} \frac{|w_t(v,i), w_t(v,j)|^2}{k_v}} \tag{2.9}$$

— Étape 2 : il s'agit du fusionnement de deux communautés en une seule $C_3 = C_1 \cup C_2$ et la création d'un nouveau partitionnement $Pk+1 = (Pk \backslash \{C_1, C_2\}) \cup \{C_3\}$.
— Étape 3 : dans cette étape, les distances entre les communautés sont calculées et on retourne à l'étape 1.

Après $n-1$ itérations, l'algorithme a comme partition Pn = $\{V\}$ représentant le graphe entier. Cet algorithme est basé sur l'application d'une stratégie de partitionnement hiérarchique par le fusionnement des communautés à faible distance. Le problème le plus important ici est de sélectionner la bonne structure de communautés pendant les itérations.
L'algorithme (Zhou et al., 2018, hou et al., 2018) utilise la modularité de Newman (Newman, 2006b, ewman, 2006b) afin de construire des communautés finales.

La complexité de Walktrap est de O(nmH) où m désigne le nombre des liens, n est le nombre des noeuds et H dénote la hauteur du dendrogramme. Si les sommets sont fusionnés successivement, H abouti son maximum et vaut n-1. En fait, la majeure limite de cet algorithme est le paramètre t représentant le nombre des pas ou des liens séparant les noeuds.

La distinction entre Walktrap algorithme et les autres approches agglomératives (Xi et al., 2016, i et al., 2016)(Bedi and Sharma, 2016, edi and Sharma, 2016) est faite au niveau du choix des communautés fusionnées à chaque itération.
- Les techniques de regroupement par liaisons simples sélectionnent à chaque itération les deux communautés ayant le plus petit élément xij de la matrice de similarité X, telles que le sommet i est dans une communauté et le sommet j dans l'autre communauté.
- Les techniques de regroupement complet des liens sont basées sur la procédure opposée. En fait, elle sélectionne, à chaque itération, l'élément maximum xij de X et fusionne les communautés de sommets i et j . La technique de clustering par couplage moyen est utilisée pour calculer la moyenne de toutes les valeurs de similarité.

— **Algorithmes divisifs**

Les algorithmes divisifs sont employés pour diviser un réseau en de nombreuses com-

munautés en supprimant itérativement les liens entre les noeuds. Au début, de l'exécution de cet algorithme une seule communauté (le réseau entier), en haut du dendrogramme, est considérée jusqu'à obtenir n communautés à un seul noeud montrant les feuilles du dendrogramme. Tout réseau connexe est considéré, dans chaque itération, comme une communauté. Les techniques introduites peuvent être distinguées selon le choix des liens à supprimer et selon les poids accordés) ces liens. L'algorithme de division le plus appliqué est celui de Girvan et Newman (Girvan and Newman, 2002, irvan and Newman, 2002)(Newman and Girvan, 2004, ewman and Girvan, 2004). Il contient quatre étapes principales :

1- Calcul des EBC pour tous les liens du graphe.
2- Suppression du lien le plus intermédiaire.
3- Re-calcul des EBC de tous les liens restants.
4- Répétitions des étapes 2 et 3 jusqu'à la suppression de tous les liens.

La technique Edge Betweenness débute par une seule communauté C comprenant tous les noeuds du graphe. Des nouvelles communautés C_1 et C_2 avec C=$C_1 \cup C_2$ sont crées après chaque itération et s'il existe un nouveau sous-graphe qui n'est pas connexe au graphe initial. Cette procédure se répète jusqu'à que les feuilles du dendrogramme constituées par n communautés sont obtenues.

Les résultats fournis par cet algorithme montrent son efficacité pour la détection des communautés. Néanmoins, il est caractérisé par un temps de calcul très élevé. L'étape de calcul des degrés d'intermédiarité est aussi trop lente. Ajoutons à cela qu'elle est exécutée à chaque itération.

L'algorithme 'Edge Betweenness' est défini par une complexité de l'ordre de O(m2.n) où m correspond au nombre des liens et n représente le nombre des noeuds (Newman and Girvan, 2004, ewman and Girvan, 2004). Pour résoudre ce problème, des mesures presque similaires au degré de centralité d'intermédiarité locales et qui se calculent vite ont été suggérées par de nombreux algorithmes (Lancichinetti and Fortunato, 2011, ancichinetti and Fortunato, 2011)(Aynaud et al., 2011, ynaud et al., 2011). Ceci dit, ce processus reste un peu lent.

Radicchi et al.(Radicchi et al., 2004, adicchi et al., 2004) ont suggéré un algorithme identique à celui proposé par Girvan et Newman(Newman and Girvan, 2004, ewman and Girvan, 2004). La seule différence réside dans le fait que le premier emploie le concept d'un coefficient de groupement d'arêtes plutôt qu'un concept d'arête entre deux nuances. L'algorithme de Girvan et Newman supprime, à chaque itération, le lien ayant la plus haute valeur, tandis que celui de Radicchi et al. supprime le lien ayant le coefficient de regroupement des liens le plus bas. En fait, le coefficient de groupement de lien $\{i, j\}$ est obtenu comme suit :

$$C_{i,j}^{g} = \frac{Z_{i,j}^{g}}{S_{i,j}^{g}} \tag{2.10}$$

où $Z_{i,j}^{g}$ est le nombre de cycles de longueur g que le lien i,j fait partie, et $S_{i,j}^{g}$ est le nombre maximum possible de cycles de longueur g que le lien i,j peut faire partie.
La majeure faiblesse de l'algorithme ascendant c'est qu'il est efficace seulement si appliqué dans les petits et non pas dans les grands groupes (Yang et al., 2016, ang et al., 2016). De plus, l'efficience de calcul, la robustesse et l'application simple de l'approche de classification hiérarchique divisive ont été prouvées dans plusieurs études (Chowdhary, 2017, howdhary, 2017).

Malgré que plusieurs algorithmes de détection de communauté hiérarchique aient été développés, de nombreux problèmes sont encore non résolus. Pour ce qui concerne les réseaux à grande échelle, par exemple, l'efficacité de la plupart des algorithmes proposés est faible et leur complexité temporelle est élevée. En conséquence, les algorithmes hiérarchiques appliqués pour la détection des communautés avec une qualité élevée permettent de construire des réseaux de petite ou moyenne taille, mais leur application efficace dans les grands réseaux demeure impossible à cause de leur grande complexité (Chowdhary, 2017, howdhary, 2017) (Xi et al., 2016, i et al., 2016). Une autre limite des approches de regroupement hiérarchique réside dans le fait qu'elles ne donnent aucune importance aux petites communautés qui font être fusionnées avec les plus grandes. Ce problème est nommé "la limite de résolution" (?, ?). Malgré que les méthodes descendantes soient efficaces pour la détection de grandes communautés, ces techniques ne sont pas prometteuses pour la détection des partitions plus petites. On peut conclure, par conséquent, qu'une division satisfaisante ne peut jamais être obtenue par la méthode hiérarchique (Herrmann et al., 2016, errmann et al., 2016).

Approches heuristiques visant à optimiser un critère

La recherche d'une partition qui améliore un critère précis est généralement NP complète puisque le nombre des partitions possibles d'un ensemble explose par le nombre de ses éléments. Subséquemment, les heuristiques, qui s'agissent d'une exploration contrainte, souvent plus intelligentes pour ce problème, sont très importantes dans le partitionnement de graphes. L'étude considère, par conséquent, le voisinage local des sommets quand il vise à former une nouvelle partition pouvant mieux satisfaire le critère à maximiser.

Ces techniques sont classées par Noack et al.(Noack and Rotta, 2009, oack and Rotta, 2009) en deux catégories. La première enferme les algorithmes de dégrossissement, strictement agglomératifs qui sont basée sur la fusion communautés. Dans ces méthodes, une fusion sûre est cherchée puisqu'elle est définitive.

- *Pas à pas* . La classe d'un sommet est modifiée.
- *Multi-pas* . Dans, cette approche, la classe de nombreux sommets change à la fois.
- *Priorisation de fusions.* La deuxième classe comprend les algorithmes à raffinement. Cette priorisation permet de revenir après-coup sur une fusion.
- *Complete Greedy* attribue à chaque itération la meilleure modification d'affectation. Ce changement est affecté sur tous les sommets et toutes les classes d'affectation.

— *Fast Greedy* affecte successivement tous les sommets les dans la meilleure classe possible.
— L'adaptation de Kernighan-Lin(Kernighan and Lin, 1970, ernighan and Lin, 1970) (pour la coupure minimale) a été faite par Newman pour la modularité (Newman, 2006b, ewman, 2006b) afin d'agrandir la variabilité des partitions explorées et pour ne pas changer l'affectation de chaque sommet qu'une seule fois. Cependant, il n'est plus important que le changement d'affectation résulte en un gain par rapport à l'affectation initiale.
— Le raffinement multi-niveaux : en appliquant les approches précédentes, il est impossible de fusionner deux sous-communautés denses moyennement reliées sans réaliser différents changements d'affectation qui influencent négativement la modularité. Les techniques employant le raffinement multi-niveaux permettent d'obtenir des résultats intermédiaires constitués par des graphes ayant des sommets qui forment des agglomérations des sommets des graphes des niveaux inférieurs. Si la convergence d'un niveau est réalisée, un nouveau graphe sera crée à partir des communautés courantes. Le processus peut ultérieurement être appliqué sur ce nouveau graphe (Blondel et al., 2008, londel et al., 2008).

Nous allons décrire, au-dessous, les techniques qui reposent sur l'optimisation de la modularité et présentons leurs résultats. Cependant, celles-ci étant basées sur des heuristiques, leur efficacité en temps et en mémoire ainsi que leur faculté à renvoyer une solution proche de l'optimale est propre à chaque méthode. En 2004, Clauset, Newman et Moore ont proposé FastQ pour Fast Modularity (Newman and Girvan, 2004, ewman and Girvan, 2004). Puisque il est impossible d'obtenir la modularité de toutes les partitions, FastQ emploie la modularité pour le déroulement d'un processus de recherche glouton.

La première étape de cette méthode consiste à déplacer tous les sommets dans des classes dont ils sont les seuls représentants. Des fusions successives qui fournissent les meilleurs gains de modularité sont donc cherchées. Dans la matrice employée, cette technique calcule le gain de modularité obtenu par les diverses fusions possibles.

Blondel et al. Ont prouvé que cette méthode donnent plus d'importance pour les grandes communautés (Blondel et al., 2008, londel et al., 2008). La complexité de FastQ est, selon ses auteurs, de O(n^3.log(n)) dans le pire des cas, mais de O(n.log(n)) en pratique le plus souvent.

En fait, la méthode de Louvain(Blondel et al., 2008, londel et al., 2008) est basée sur la classification des sommets dans un graphe valué. La vitesse de cette technique lui rend applicable dans les réseaux ayant une grande taille. Cette technique n'est pas supervisée (le nombre des groupes à construire n'est pas exigé avant l'exécution). Elle retourne une partition en améliorant le critère de modularité. Il est primordial de mettre en relief cet algorithme ne peut pas fournir un maximum global, mais il résulte généralement en un optimum local. La technique gloutonne de Louvain a été utilisée pour améliore la modularité sur un graphe optionnellement valué(Aynaud et al., 2011, ynaud et al., 2011). Elle contient deux phases exécutées en alternance.

Dans la première étape,chaque sommet forme une communauté. Ultérieurement, on

considère les sommets un à un (le résultat obtenu est dépendent de l'ordre selon lequel les sommets sont énumérés). Chaque sommet est mis dans l'une des communautés voisines (la sienne étant incluse) choisie puisqu'elle réduit le gain de modularité. Cette procédure est répétée jusqu'à ce que le déplacement du sommet devienne impossible. On parle, dans ce cas, d'optimum local. Dans la deuxième phase, un nouveau graphe est construit entre les communautés trouvées à la fin de l'étape 1 : Pour chaque communauté et pour deux communautés C et C', il y a respectivement un sommet dans le graphe et une arête de valuation w où w = $\sum v, v' \in CxC' poids(v, v')$.

Une boucle existe sur C de poids w =$\sum v, v' \in CxCpoids(v, v')^2$. Les phases 1 et 2 ont été exécutées alternativement jusqu'à ce que la modularité ne peut plus s'améliorer. L'effet de l'ordre de l'énumération des sommets pendant le déroulement de l'algorithme a été examiné par (Aynaud et al., 2011, ynaud et al., 2011). Les temps de calcul et la modularité de la partition finale n'ont pas été optimisés par l'énumération des sommets selon l'ordre de leurs classes d'appartenance (tous les sommets de la première communauté puis tous les sommets de la seconde communauté, etc.). La technique rapide proposée a été appliquée sur des réseaux de plusieurs millions de sommets. Elle a amélioré le critère de modularité.
Dans (Zhou et al., 2018, hou et al., 2018), la méthode Bi-Louvaina été proposée pour étudier la structure de la communauté dans les réseaux bi-partites et pour optimiser la modularité bi-partite par l'application de la technique gloutonne.

Malgré que les méthodes basées sur l'algorithme Louvain ont montré leur efficacité, il a été constaté après que ces techniques tenaient compte seulement des informations de liaison qui existent entre les noeuds et ne se concentrent pas sur les noeuds voisins, ce qui réduit le nombre des noeuds d'une même communauté et influence la précision des résultats finaux.

En outre, les auteurs qui ont développé la technique de Louvain ont focalisé sur la classification des réseaux en mouvement. Ils ont noté l'inconvénient de cette technique qui résiste bien aux changements minimes du réseau, pouvant occasionner une modification locale et globale de la partition. Comme toutes les techniques d'optimisation de la modularité, celle de Louvain a ses propres limites.

La technique de Louvain a été appliquée dans plusieurs études afin de résoudre quelques problèmes, particulièrement ceux de la modularité elle-même ou pour l'adapter ou de l'étendre. Collingsworth et al. ont proposé une technique basée sur l'entropie de sommet (Collingsworth and Menezes, 2014, ollingsworth and Menezes, 2014). Cette méthode permet de mesurer l'entropie dans le cas où on change la communauté d'affectation d'un sommet :

$$H_S = -\sum_{i=1}^{n} \frac{p_i log_2(pi)}{e^{\frac{k}{4}}}$$

où n désigne le nombre des communautés qui peuvent être rejointes potentiellement par le sommet peut rejoindre, k correspond au nombre de triades (sous-graphes complet comportant 3 sommets) intra-communautaires créer en unissant une communauté et pi représente le taux des arêtes voisines du sommet considéré. Ces arêtes relient le taux à la

communauté numéro i. En fait, le mot triade réfère au sous-graphe complet contenant 3 sommets.

Dans (Xi et al., 2016, i et al., 2016) ont optimisé la fonction de modularité de l'algorithme de Louvain en introduisant la notion de la similarité de noeud. Ils ont proposé l'algorithme SHC(Similarity based Hierarchical Community). Ensuite, Staudt et Meyerhenke (Staudt and Meyerhenke, 2015, taudt and Meyerhenke, 2015) ont suggéré, en 2016, un algorithme rapide qui emploie l'application d'une technique en deux étapes pour former une structure de communauté basée sur la propagation d'étiquettes. HAMUHI-CODE (Heuristic Algorithm for MUlti-scale Hierarchical COmmunity Detection) a été proposé dans (Castrillo et al., 2017, astrillo et al., 2017), comme nouvel algorithme heuristique rapide utilisé pour détecter les communautés hiérarchiques multi-échelles. Cet algorithme repose sur une technique de classification hiérarchique agglomérative. Une nouvelle similarité structurelle de sommets basée sur la similarité classique en cosinus en supprimant certains sommets a été définie pour accroitre la probabilité d'identifier les arêtes intergroupes. Il est clair que les algorithmes cités au dessus ne sont pas évolutives en présence d'un grand nombre de communautés. Ils résultent en un optima local de la modularité.

L'algorithme infomap (Rosvall and Bergstrom, 2008, osvall and Bergstrom, 2008) est basé sur l'associe d'un code en deux parties à chaque noeud : un préfixe représentant la communauté et un identifiant du noeud dans la communauté. On peut décrire un chemin en fonction des identifiants des noeuds qu'il parcourt et aussi du préxe des communautés traversées. Le but, ici, est de chercher les préfixes communautaires qui permettent de réduire le nombre moyen des bits nécessaires pour la représentation d'une marche aléatoire. Cette optimisation est réalisée de manière gloutonnement. Ensuite, elle est ranée avec le biais d'un algorithme de recuit simulé.

L'algorithme SCD (Prat-Pérez et al., 2014, rat-Pérez et al., 2014)(Scalable Community Detection) améliore considérablement un score qui mesure la proportion des triangles existant à l'intérieur des communautés. Le score en question est approché par cet algorithme pour ne pas calculer le nombre des triangles. Semblable à l'algorithme de Louvain, algorithme SCD modifie localement les noeuds de communautés et donne plus aux noeuds la chance de s'isoler dans d'autres nouvelles communautés. Cet algorithme se termine par la stabilisation du score de graphe. Les auteurs dans (Zhang et al., 2018, hang et al., 2018) ont développé PMAC (Partial Matrix Approximation Convergence) qui est une technique de détection de communauté hiérarchique. L'étape principale de PMAC révèle la convergence approximative locale des noeuds centraux plutôt que la convergence globale dans la matrice de transfert de probabilité.

En appliquant une seule méthode de clustering, on peut obtenir des hiérarchies de communautés non équilibrées et dominées par une grande communauté unique absorbant des sommets particuliers un par un. Ceci résulte en un déséquilibre au niveau de la hiérarchie à tous les niveaux hiérarchiques. Par conséquent, malgré le rôle important que jouent les algorithmes hiérarchiques divisifs et agglomératifs, la combinaison des ces types d'algorithmes n'a pas bien considérée dans les travaux précédents pour former une structure de communauté hiérarchique complète. Pour cela, d'après cette étude, nous visons à créer

une hiérarchie de communautés qui montre nettement et entièrement la structure de la communauté dans le réseau social.

2.4 Critères d'évaluation

Deux modes d'évaluation des techniques proposées pour la détection des communautés dans les graphes ont été présentés dans la littérature(Kim et al., 2017, im et al., 2017). Le premier type d'évaluation est celle réalisée en utilisant des critères internes pour obtenir la meilleure partition la en fonction d'un critère particulier tel que la modularité.

Le deuxième mode d'évaluation consiste en la confrontation des classes obtenues par la technique à des groupes « naturels ». Par conséquence, le résultat est évalué selon un critère externe. Le score est calculé en fonction de précision et de rappel ou de F-mesure. En fait, les indices externes sont différent des indices internes à cause de la présence des informations de catégories connues. Les indices les plus employés sont présentés dans la section suivante.

2.4.1 Critères d'évaluation interne

Les indices internes représentent des mesures de validation employées pour évaluer les résultats de la classification en prendre en considération seulement les informations intrinsèques aux données sous-jacentes. Un indice des indices les plus anciens et les plus appliqués a été introduit par Dunn (Dunn, 1973, unn, 1973) pour identifier les groupes compactes et bien séparées en optimisant la distance inter-groupes tout en réduisant la distance intra-groupes. L'indice Dunn pour k groupes peut être exprimé comme suit :

$$k^* = argmax_{k\geq 2}\{DU(K) = min_{i=1,\ldots,k}(min_{j=i+1,\ldots,k}(\frac{D(C_i,C_j}{max_{m=1,\ldots k}diam(C_m)})\}$$

où D (C_i, C_j) correspond à la distance entre deux groupes $(C_i et C_j)$ comme distance minimale entre une paire d'objets au sein des deux groupes distincts et le diamètre de la groupe C_m, diam C_m, en tant que distance maximale entre deux objets du même groupe. Le nombre maximal des groupes est obtenu en fonction de la valeur la plus grande de l'indice de Dunn sensible au bruit. Une famille d'indices de validation des groupes est formée par la redéfinition du diamètre et de la distance des groupes.

Comme l'indice Dunn, celui de Davies-Bouldin (Davies and Bouldin, 1979, avies and Bouldin, 1979) est appliqué pour obtenir des groupes avec une distance minimale intra-groupe et une distance maximale entre les centroïdes des groupes. En fait, la valeur minimale de l'index montre une partition adéquate pour l'ensemble des données. L'indice Davies-Bouldin est obtenu par :

$$k^* = argmin_{k\geq 2}\{DB(K) = \frac{1}{k}\sum max^k_{i=1,\ldots,k,i\neq j}(\frac{D(diam(C_i)+diam(C_j)}{d(z_i,z_j)})\}$$

où le diamètre d'un cluster est défini comme :

$$diam(C_i) = \sqrt{\frac{1}{n}} \sum_{o \in C_i} d(o, z_j)^2$$

L'index Silhouette (Kim et al., 2017, im et al., 2017) peut être défini comme un critère interne utilisé pour quantifier la qualité d'une partition créée sur un réseau d'information. Pour chaque objet, cet index calcule une largeur en fonction de son appartenance à un cluster donné. Pour le ième objet, ai est la distance moyenne qui sépare un objet aux autres objets qui appartiennent à son groupe et bi représente le minimum de la différence moyenne entre l'objet i et ceux qui appartiennent aux autres groupes.

La largeur de la silhouette est écrite par l'expression suivante : $(b_i - a_i)/max\{a_i, b_i\}$. L'indice Silhouette représente la moyenne largeur Silhouette de tous les points de données. La partition avec le SI le plus élevé (k) est optimale.

$$k^* = argmax_{k \geq 2}(SI(K) = \frac{1}{n} \sum_{i=1}^{n} (\frac{a_i - b_i}{max(b_i - a_i)}))$$

Récemment, l'index géométrique a été développer afin d'intégrer les données avec des clusters de diverses densités et des clusters se chevauchant. Le nombre maximal des clusters est obtenu en réduisant l'indice GE (k). On suppose que d est la dimensionnalité des données et λ_{pq} représente la valeur adéquate de la matrice de covariance à partir des données. D (C_i, C_j) désigne la distance inter-cluster séparant le groupe i et le groupe j. L'indice GE est construit comme :

$$k^* = argmin_{k \geq 2}(GE(K) = max_{1 \leq i \leq k}(\frac{2 \sum_{j=1}^{d} \sqrt{\lambda_{ji}^2}}{min_{1 \leq j \leq k, i \neq j} D(C_i, C_j)}))$$

La conductance d'une classe, aussi nommée métrique de la coupure normalisée, est employée pour la mesure du taux d'évaluations d'arêtes à l'extérieur de la classe, à la classe ou à son complément, selon quel côté de la coupure la somme des évaluations des arêtes est la moins importante (Leskovec et al., 2008, eskovec et al., 2008). Ce critère a été, au début, introduit pour l'évaluation de la qualité d'une coupure, qui consiste à scissionner les sommets d'un graphe en deux parties. Par conséquent, on peut définir la conductance d'une classe C comme montré au dessous :

$$Conductance(C) = \frac{\sum u \in C \sum v \notin C}{min(\sum u \in C \sum v \in C A_{u,v'} \sum u \in C \sum v \notin C A_{u,v'})}$$

La conductance du graphe peut être définie, à partir de cette mesure, comme la plus petite conductance de chaque classe :

$$Conductance(G) = min_{C_{k \in \varphi}}(conductance(C_k))$$

En fait, une classe avec une conductance forte est celle ayant des fortes arêtes et des faibles liaisons avec les autres classes. Un graphe avec une conductance forte a toutes ses classes denses et faiblement liées entre elles.

2.4.2 Critères d'évaluation externe

Le deuxième mode d'évaluation consiste à la confrontation des classes obtenues par la technique proposée à des groupes « naturels ». Il est clair donc que le résultat est évalue selon un critère externe.

La NMI(Normalized Mutual Information) est une technique de comparaison qui repose sur la théorie de l'information . Le MI (Informations mutuelles) entre deux partitions, C_i et C_k, est utilisée pour la mesure lu degré de dépendance entre elles. Il est défini comme suit :

$$MI(C_i, C_k) = \sum_{c_j^i} \in C_i \sum_{c_{j'}^k} \in C_k \frac{c_j^i \cap c_{j'}^k}{N} log_2(\frac{N|c_j^i \cap c_{j'}^k|}{|c_j^i||c_{j'}^k|})$$

où $c_j^i \cap c_{j'}^k$ dénote le nombre d'objets qui se trouvent dans l'intersection de deux partitions. La mesure MI est profondément liée à celle de l'entropie d'une partition (H) obtenue par :

$$H(C_i) = \sum_{c_j^i} \in C_i \frac{|c_j^i|}{N} log_2 \frac{|c_j^i|}{N}$$

De ce fait, NMI correspond au rapport entre MI et H. A représente la structure réelle du réseau et $P = \{C_1, C_2, ..., C_K\}$ est le partitionnement calculé par n'importe quel modèle. NMI est défini par l'équation suivante :

$$NMI(C_i, C_k) = \frac{MI(C_i, C_k)}{\sqrt{H(C_i, H(C_k)}}$$

De plus, pour l'évaluation des performances d'un algorithme de classification, le schéma de comptage est employé par les clusters basées sur le critère de l'entropie (CCE). La dérivation de ce critère, qui relie le critère et l'approche de classification, est faite du cadre formel des modèles de classification probabiliste. Le critère basé sur l'entropie recherche le nombre maximum des clusters par la suppression automatiquement des groupes ayant un coût de l'information négatif. $C = \{C_1, C_2, ..., C_K\}$ est un partitionnement calculé en appliquant n'importe quel modèle. Le codage des éléments de chaque groupe c_k est réalisé par une densité optimale d_i de la famille D_i.

$$CCE(c_1, d_1, ..., c_k, d_k) = \sum_{i=1}^{k} p_i.(-ln(p_i + H^x)(c_i||d_i))$$

où $p_i = c_i/c$ et $H^x(c_i||d_i)$ est la classe de densité la plus commune pour laquelle l'entropie croisée est obtenu comme suit :

$$H^x(Y||G\sum) = \frac{N}{2} ln(2\Pi) + \frac{1}{2} tr(\sum^{-1}, \sum_x) + \frac{1}{2} lndet(\sum)$$

Le critère basé sur l'entropie est dérivé du cadre formel des modèles de classification probabiliste. Il établit les liens entre le critère et l'approche de classification. Nous utilisons cette mesure comme critère de densité basé sur l'entropie pour vérifier la validité introduite de la classification hiérarchique mixte introduite. Ce critère cherche le nombre optimal des groupes en supprimant automatiquement les groupes dont le coût d'information est négatif.

Soit $C = c_1, c_2, ..., c_k$ un partitionnement calculé par n'importe quel modèle. Nous considérons que les éléments de chaque groupe c_k sont codés par une densité optimale d_i de la famille D_i.

$$CEC(c_1, d_1; ....; c_k, d_k) = \sum_{i=1}^{i=k} p_i.(-\ln(p_i + H^{\times})(C_i \| d_i)) \tag{2.11}$$

où $p_i = \frac{c_i}{c}$ et $H^{\times}(C_i \| d_i)$ indique la classe de densité la plus courante, pour laquelle l'entropie est :

$$H^{\times}(y \| G\Sigma)) = \frac{N}{2} ln(2\pi) + \frac{1}{2} tr(\Sigma^{-1}\Sigma_x) + \frac{1}{2} \ln \det(\Sigma) \tag{2.12}$$

Généralement, CEC partitionne les m membres en k groupes afin de minimiser la fonction de coût appelée énergie du regroupement E en commutant les membres entre les groupes. La fonction CEC réduit les groupes dont les cardinalités ont diminué au-dessous d'un petit niveau préfixé. En effet, la fonction énergie ou la fonction de coût E est donnée par :

$$E(c_1, D_1; ...; c_K, D_K) = \sum_{i=1}^{k} p(c_i)(-\ln(p(c_i))) + H^{\times}(c_i \| D_i) \tag{2.13}$$

où c_i désigne le $i_{ème}$ groupe, $p(c_i)$ est le rapport entre le nombre des membres du $i_{ème}$ groupe et le nombre total des membres du réseau, $H(c_i \| D_i))$ correspond à la valeur d'entropie croisée, ce qui représente la fonction énergétique interne du cluster des données c_i définies par rapport à une certaine famille de densité D_i codant pour le type de cluster considéré.

2.5 Conclusion

Les notions de base des graphes complexes, spécifiquement des réseaux sociaux, ont été présentées dans ce chapitre. Les techniques de classification hiérarchique et celles employées pour l'optimisation d'un critère ont été discutées. Nous avons aussi décrit les modes d'évaluation des techniques de détection des communautés proposées.

Nous avons montré que les interactions sociales, qui changent avec le temps, sont importantes pour la compréhension de l'évolution des réseaux sociaux. Par conséquent, la détection des communautés et des évènements importants qui attirent l'attention des utilisateurs ainsi que le suivi de leurs changements sont primordiaux. Les études précédentes qui examinent l'évolution des communautés dans les réseaux sociaux seront traitées. Ensuite, nous illustrons le problème de détection des événements.

Chapitre 3

Détection et suivi des structures communautaires dynamiques et d'événements dans les médias sociaux

3.1 Introduction

Pendant les années récentes, les médias sociaux ont largement affecté la production et la diffusion de l'information concernant des événements personnels ainsi que globaux en publiant ou retransmettant des messages, en temps réel. En fait, l'augmentation du nombre des utilisateurs des médias sociaux est accompagnée d'un accroissement du volume de messages publiés. Ces énormes quantités de données provoquent des problèmes : la taille, le bruit et le dynamisme(Dugué and Perez, 2014, ugué and Perez, 2014). Puisque l'augmentation de ce volume engendre un phénomène de surcharge informationnelle, il devient plus difficile d'identifier des éléments d'informations pertinentes liées à des évènements importants.
Ainsi, l'analyse manuelle des données des réseaux sociaux semble impossible et la détection automatique des évènements importants, à partir des médias sociaux, est une tâche complexe(Guille and Favre, 2014, uille and Favre, 2014). Pour remédier à ce problème, l'exploration des données fournit un large éventail de techniques permettant de détecter les événements et analyser l'évolution des structures communautaires. Dans ce chapitre, nous réalisons, dans la section 3.2, une étude bibliographique sur la détection des évènements suscitant l'intérêt des utilisateurs des médias sociaux. Dans la section 3.3, un état de l'art, des principales méthodes de suivi des changements, d'évolution des transitions des groupes sociaux est présenté. Nous présentons dans la section 3.4 les techniques de

prédiction des liens.

3.2 Les méthodes de détection d'événements dans les réseaux sociaux

Les méthodes de détection d'évènements à partir des médias sociaux ont été utilisées dans de nombreux travaux portant sur la détection de thématiques, de motifs saillants et d'évènements à partir de flux textuels. Dans ces techniques, un évènement peut être défini comme une thématique propre à une certaine période de temps (Bjorne et al., 2010, jorne et al., 2010).

Une distinction commune des approches d'extraction d'événements découle du domaine de modélisation(Hogenboom et al., 2016, ogenboom et al., 2016). Les approches basées sur les données visent à convertir les données à la connaissance à travers l'utilisation des statistiques, de data mining et d'apprentissage automatique. D'autres méthodes reposent sur les connaissances expertes. Elles extraient des connaissances en exploitant les connaissances spécialisées existantes. Généralement, ces techniques sont basées sur les modèles de graphes. En ce qui concerne les procédures d'extraction avancées, les chercheurs peuvent utiliser des méthodes des deux domaines (données ou connaissances) en amorçant ou en optimisant leurs algorithmes reposant sur la connaissance au moyen d'un apprentissage automatique ou inversement.

3.2.1 Approches basées sur les données

Les approches basées sur les données développent des modèles de corpus de texte proches des phénomènes linguistiques.
Ces techniques d'extraction d'événements ne se limitent pas à un raisonnement statistique basé sur la théorie des probabilités, mais englobent toutes les approches quantitatives du traitement automatique du langage, telles que la modélisation probabiliste, la théorie de l'information et l'algèbre linéaire.

Dans la littérature, il existe plusieurs méthodes de détection d'évènements à partir des médias sociaux par des méthodes qui définissent des métriques permettant de scorer les termes rencontrés dans les messages de telle sorte que les termes liés à des évènements obtiennent les scores les plus élevés. Pour faciliter l'identification des évènements, les termes sont classés selon ces scores. Nous citons, à titre d'exemple, la méthode proposée par Shamma et al(Shamma et al., 2011, hamma et al., 2011) nommée Peaky Topics qui repose sur le calcul d'une mesure de fréquence normalisée ntf_i pour chaque mot $t \in V$ en chaque tranche temporelle i $\in [1; n]$. La fréquence est normalisée par le nombre total des occurrences du mot t dans le flux de messages afin que les mots fréquents, en une tranche temporelle et rares dans le reste du flux, aient une valeur de ntf élevée et faible pour les

autres valeurs. La fréquence normalisée est définie ainsi comme suit :

$$ntf_{t,i} = \frac{tf_{t,i}}{cf_t} \tag{3.1}$$

où $tf_{t,i}$ est la fréquence du mot t à la ième tranche temporelle et cf_t désigne le nombre total des occurrences du mot t dans le corpus C. Pour classer les mots liés à des évènements, les auteurs ont défini, dans une deuxième étape, un score pour chaque mot t :

$$peakiness_t = \max_{ntft_i} \tag{3.2}$$

Chaque entrée du classement est décrite par un mot, son score et la tranche temporelle maximisant la métrique ntf. La normalisation proposée ne tient compte que de la variabilité de la fréquence du mot considéré à travers le temps. Par conséquent, les mots, dont la fréquence est uniformément distribuée (qu'ils soient toujours très fréquents ou toujours très rares), auront un score peakiness proche de $\frac{1}{n}$(où n est le nombre des tranches temporelles). De la même façon, un mot qui n'apparaît que dans une tranche temporelle, quelle que soit sa fréquence absolue dans cette tranche, aura un score égal à 1. L'autre limitation de la méthode Peaky Topics est liée au fait qu'un seul mot peut être insuffisant pour décrire un évènement complexe à cause de la possible ambiguité et du manque de contexte.

Pour pallier à ces limitations, les auteurs dans (Benhardus and Kalita, 2013, enhardus and Kalita, 2013) ont concentré sur l'étude de N-grammes des mots (i.e. séquences de N mots consécutifs) et, plus particulièrement, sur les bi-grammes et trigrammes. Pour chaque N-gramme et tranche temporelle, ils ont suggéré de calculer un score, nommé trending score (aussi noté TS), équivalent à une fréquence normalisée. La normalisation a été effectuée par rapport au nombre total d'occurrences du N-gramme dans le corpus de messages et aussi, contrairement à la méthode Peaky Topics, par rapport à la fréquence des autres N-grammes dans la même tranche temporelle. Le trending score est défini comme présenté au dessous :

$$TS_{t,i} = \frac{ntf_{t,i}}{atf_{t,i}} \tag{3.3}$$

où : $ntf_{t,i}$ est la fraction entre tf_i et $\sum_{k \in V} tf_{k,i}$

Ce qui rend la normalisation de la fréquence du mot t, par rapport à celle des autres mots du vocabulaire est possible ce qui normalise la fréquence du mot t par rapport à sa fréquence dans les autres tranches temporelles. En outre, il existe des méthodes qui décrivent chaque évènement à l'aide d'un ensemble pondéré de mots. Lau et al.(Lau et al., 2012, au et al., 2012) ont développé On-line LDA, une variante en ligne du modèle LDA (i.e. Latent Dirichlet Allocation (Blei et al., 2003, lei et al., 2003))) et une technique de mesure de l'évolution des thématiques et de détection des évènements. LDA est un modèle génératif probabiliste qui apprend un ensemble de thématiques latentes à partir d'une collection de documents ; chaque document étant considéré comme un sac de mots. Un

document est caractérisé par une distribution sur un nombre fixé (K) de thématiques. Cependant, une thématique est une distribution de probabilités sur le vocabulaire V des mots employés dans les documents. Yuheng et al.(Hu et al., 2012, u et al., 2012), constatant que le modèle LDA s'adapte mal aux documents courts (tel qu'il est le cas des messages publiés dans les médias sociaux, ont introduit ET-LDA (Event and Tweets LDA). Les auteurs ont proposé premièrement d'enrichir les messages publiés sur les médias sociaux. Dans cette approche, chaque message est utilisé comme requête sur un moteur de recherche traditionnel. Ensuite, il est enrichi par l'adjonction des mots les plus fréquents dans les résultats de la recherche. Deuxièmement, ET-LDA modélise deux corpus conjointement le premier est le corpus des messages enrichis et le second un corpus d'articles tirés des médias traditionnels afin de favoriser la distinction entre les thématiques d'arrière plan et les thématiques liées à des évènements. Les évènements détectés correspondent uniquement à ceux traités dans le corpus d'articles utilisé. Même si les messages issus des medias sociaux peuvent potentiellement apporter des informations supplémentaires, leur impact est minoré par le fait que ces messages ont été altérés par l'information apportée par un moteur de recherche traditionnel. L'étude réalisée par Aiello et al.(Aiello et al., 2013, iello et al., 2013) a montré que les méthodes de détection d'évènements, reposant sur la modélisation des thématiques latentes, souffrent de plusieurs limitations. Il apparaît aussi que ce type de techniques est particulièrement inefficace pour traiter des flux de messages dans lesquels de nombreux évènements distincts sont discutés. Pour l'extraction des événements pilotés par les données, on peut distinguer les approches d'apprentissage supervisé et les approches d'apprentissage non-supervisé. Les premières approches nécessitent des connaissances d'experts, car les données étiquetées sont fournies à l'apprentissage des algorithmes. Cependant, les dernières approches sont généralement utilisées lorsqu'aucune donnée étiquetée n'est disponible. Le regroupement des documents similaires ou apparentés (ex. : phrases, termes, etc.) est une technique non- supervisée couramment utilisée pour l'extraction d'événement. l'extraction des événements pilotés par le clustering peut être effectuée par de nombreuses méthodes. Par exemple, on pourrait utiliser le regroupement d'occurrences d'événements au fil du temps ainsi que la prédiction du type et des propriétés d'un nouvel événement (Okamoto and Kikuchi, 2009, kamoto and Kikuchi, 2009). Les options alternatives consistent à regrouper des documents contenant des événements analysés superficiellement sur le plan linguistique afin d'identifier des événements (Tanev et al., 2008, anev et al., 2008) ou des phrases faisant référence au même événement.
Dans des contextes plus complexes, le regroupement est généralement associé à des structures de graphes avancées (Hogenboom et al., 2016, ogenboom et al., 2016). Li et al. (Li et al., 2012, i et al., 2012) ont proposé la méthode TwEvent qui vise à regrouper non pas des mots mais des N-grammes de mots. Selon cette technique, les N-grammes candidats sont d'abord identifiés et sélectionnés sur la base d'informations statistiques fournies par Microsoft Web NGram service (3) ainsi que leur fréquence d'apparition sur Wikipedia en tant que labels (étiquettes) de liens pointant vers d'autres articles. Les auteurs ont défini ensuite paramétriquement la probabilité qu'un N-gramme soit saillant en une tranche temporelle donnée en fonction d'une mesure de fréquence normalisée. Dans l'étape suivante, les N-grammes ont été regroupés selon une stratégie du type « k plus proches voisins », à l'aide de l'algorithme décrit par Jarvis et Patrick (Jarvis and Patrick, 1973, arvis and Patrick, 1973), pour une fenêtre de taille fixe. Constatant que l'extraction des N-grammes

par TwEvent est à la fois coûteuse et grandement influencée par Microsoft Web N-Gram et Wikipédia. Parikh et Karlapalem (Parikh and Karlapalem, 2013, arikh and Karlapalem, 2013) ont développé la méthode ET. Cette technique ne considère que les bi-grammes et, plus particulièrement, les bi-grammes saillants détectés à l'aide d'une mesure de fréquence normalisée par rapport au temps. Dans (Shi et al., 2017, hi et al., 2017), les auteurs ont décrit une nouvelle méthode de détection de similarité des événements basée sur la mesure en cosinus afin d'évaluer la corrélation entre les événements. Évidemment, les techniques bassées sur le clustering présentent certains inconvénients. Notamment, elles ont tendance à produire des clusters de grande taille. Valkanas et Gunopulos (Valkanas and Gunopulos, 2013, alkanas and Gunopulos, 2013) ont noté, à ce propos, que les méthodes à base de clustering, en regroupant de manière « aggressive » certains termes, incorporent du bruit dans les descriptions des évènements. L'utilisation des approches basées sur les données pour l'extraction d'événements présente un avantage principal. En fait, il n'est pas nécessaire de disposer des connaissances spécialisées ni de ressources linguistiques. Cependant, les approches basées sur les données nécessitent un énorme corpus de texte afin de développer des modèles qui se rapprochent des phénomènes linguistiques. Un autre inconvénient est que les méthodes basées sur les données ne traitent pas de la signification du texte. Pour remédier à ce problème, les chercheurs ont recours à des approches fondées sur les connaissances basées sur des schémas exprimant des règles représentant des connaissances expertes.

3.2.2 Approches basées sur les connaissances pour l'extraction d'événements

Les méthodes d'extraction des événements, basées sur les connaissances, utilisent souvent des modèles prédéfinis (ou appris) exprimant des règles de connaissances expertes. Leurs procédures de TM (Text Mining) reposent donc intrinsèquement sur les connaissances linguistiques et lexico-graphiques ainsi que sur les connaissances humaines existantes concernant le contenu des textes à traiter. Nous pouvons faire une distinction approximative entre deux types de modèles qui peuvent être appliqués aux corpus en langage naturel pour l'extraction des événements, à savoir les modèles lexico-syntaxiques (Hung et al., 2010, ung et al., 2010) et les modèles lexico-sémantiques (Li et al., 2002, i et al., 2002). Les anciens modèles combinent les représentations lexicales, sémantiques et syntaxiques d'informations. Ils sont plus expressifs que les modèles purement syntaxiques ou sémantiques. Avant d'utiliser des modèles d'extraction sur un ensemble de données des textes linguistiques, dans la plupart des approches fondées sur la connaissance, le corpus est prétraité à l'aide d'analyseurs fondés sur les données et sur les connaissances. La plupart des modèles fonctionnent sur des jetons, c'est-à-dire de petits segments de texte qui sont généralement des mots, des groupes de mots, des nombres, des espaces ou des signes de ponctuation. Ces jetons sont caractérisés par diverses propriétés, en fonction de processus de traitement du langage naturel analysant le corpus. Les propriétés communes sont le concept sémantique associé au jeton, la catégorie lexicale, la catégorie orthographique, le lemme avec suffixe et / ou affixe, référence pronominale, etc. Finalement, les motifs, construits selon une grammaire prédéfinie, sont appariés sur des grandes collections de

jetons. Généralement, en cas de correspondance, des données supplémentaires (propriétés) sont collectées et stockées dans des structures de données (ex. des sujets et des objets) pour une utilisation ultérieure. Lors de la mise en oeuvre des approches basées sur les connaissances pour l'extraction des événements, il s'avère souvent difficile de rester dans les limites de ces approches. La plupart des méthodes de détection d'événements comportent souvent une (petite) composante basée sur les données. Par exemple, un regroupement initial pour la classification peut être nécessaire afin de déterminer les éléments pouvant être utilisés pour construire des motifs comme les noms propres, les verbes, les noms des sociétés et des personnes. Les modèles d'analyse lexico-syntaxique apparaissent souvent dans des travaux antérieurs sur l'extraction des événements basée sur la connaissance (Yakushiji et al., 2000, akushiji et al., 2000). Mais, ils restent toujours populaires dans les approches plus récentes en raison de leur indépendance de domaine. Les modèles reposent principalement sur des propriétés syntaxiques (grammatical significations) comme les verbes, les noms, les prépositions et les pronoms. Idéalement, ils doivent être définis de manière qu'ils se produisent fréquemment et couvrent ainsi de nombreuses instances d'événements.
En pratique, tels modèles couvrent un nombre limité de déclarations différentes traitant le même événement. La généralisation du motif provoque la perte de précision lors de détection de l'événement. Comme alternative, on peut énumérer tous les verbes et les conjugaisons possibles. Cela impacte considérablement les temps de développement et la flexibilité de règle générale.
Pour faire face à ces problèmes et à d'autres problèmes connexes comme la synonymie, l'homonymie et la polysémie, des schémas lexico-syntaxiques très complexes doivent être construits. Ceci souligne la nécessité des modèles de plus haut niveau, tels que les modèles lexico-sémantiques qui permettent d'extraire plus précisément des informations à partir des textes en enrichissant les modèles lexico-syntaxiques avec la sémantique, c'est-à-dire la signification linguistique et le contexte du domaine. Les modèles lexico -sémantiques permettent d'obtenir des expressions plus puissantes car ils exploitent les modèles lexico-syntaxiques existants pour une abstraction de niveau plus élevé. Contrairement aux modèles lexico-syntaxiques, dans certains domaines, la connaissance est nécessaire pour créer des modèles de haute précision qui récupèrent de nombreux événements dans un corpus arbitraire. Cela rend la création des modèles moins triviale.
Mais, comme ces modèles peuvent être moins généraux que les modèles lexico-syntaxiques, ils permettent une description plus précise des besoins d'extraction des événements et renvoient des résultats à un niveau de précision supérieur à leurs équivalents lexico-syntaxiques, sans beaucoup perdre au niveau du taux du rappel.

En littérature, il existe deux notions de modèles lexico-sémantiques. En effet, certaines recherches sont axées sur l'extraction des événements, au moyen des bases sémantiques, ajoutée à travers les listes de mots qui sont itérativement cherché en analysant les corpus. Mais comme le passage des approches lexico-syntaxiques à celles lexico-sémantiques est une étape incrémentielle mineure, cette approche a été souvent utilisée (Hogenboom et al., 2016, ogenboom et al., 2016).
D'autres recherches ont employé des modèles basés sur des concepts et relations ontologiques qui capturent la sémantique du domaine. Ces modèles impliquent un typage plus complexe car leurs éléments capturent la sémantique du domaine et sont plus avancés que les éléments sémantiques, syntaxiques et à typage simple.

De plus, les restrictions et les relations, s'appliquant aux concepts spécifiés dans l'ontologie sous-jacente, peuvent être utilisées lors de l'application du raisonnement avec un moteur d'inférence. Comparées à d'autres approches basées sur des modèles, les modèles complexes nécessitent plus de compétences en raison de la complexité accrue. Cependant, ils génèrent souvent des meilleurs résultats à cause de leur plus grande expressivité.
Toutefois, la plupart des langages lexico-sémantiques à typage complexe réduisent leur complexité en supprimant les fonctionnalités supplémentaires fréquemment utilisées dans les langages lexico-syntaxiques, telles que les opérateurs de répétition ou les caractères génériques, car elles sont davantage axées sur l'utilisation des concepts (Aone and Ramos-Santacruz, 2000, one and Ramos-Santacruz, 2000). Certains langages offrent une expressivité totale non seulement en tenant compte des classes et des relations ontologiques, mais aussi en supportant les opérateurs d'étiquetage, de négation, de caractère générique et de répétition (IJntema et al., 2012, Jntema et al., 2012).

En outre, le problème de vérification du contenu était considéré comme une rumeur se propageant dans les médias sociaux (Shi et al., 2017, hi et al., 2017) (Boididou et al., 2018, oididou et al., 2018). Dans (Wei et al., 2017, ei et al., 2017), une approche a été introduite pour filtrer les utilisateurs non pertinents en fonction de l'estimation de l'emplacement des utilisateurs de Twitter.

Contrairement aux méthodes basées sur les données, les techniques reposant sur la connaissance requièrent peu de données. Mais, elles requièrent inversement une quantité considérable de connaissances linguistiques, lexicographiques et, pour les modèles lexico-sémantiques, une connaissance du domaine, augmentant de manière inhérente la quantité d'expertise requise.
Par rapport aux méthodes d'extraction basées sur les données, l'accent est mis moins sur le temps de développement, mais d'avantage sur les temps d'exécution nécessaires pour détecter tous les concepts requis. Les méthodes basées sur la connaissance se différencient l'une de l'autre. A titre d'exemple, les modèles lexico-syntaxiques nécessitent moins de données pour les phases de formation initiale (regroupement) car ils sont plus proches de la structure grammaticale du texte. Cependant, les modèles lexico-sémantiques requièrent souvent plus de temps de développement en raison du besoin de sémantique du domaine. Les résultats des modèles lexico-sémantiques excellent, en outre, en interopérabilité et en qualité du fait de leur traçabilité, mais les coûts de leurs maintenance sont considérablement plus élevés que ceux des modèles lexico-syntaxiques.

3.2.3 Approches hybrides pour l'extraction des événements

C'est clair que chaque type d'approches a certains inconvénients, d'où l'utilisation d'une seule catégorie d'approches d'extraction d'événement peut ne pas donner les meilleurs résultats pour détecter les événements. Ainsi, combiner des approches basées sur les données avec celles reposant sur les connaissances permettra d'atténuer les inconvénients de chaque type, ce qui crée un nouveau type d'approches appelées les approches hybrides.

Les algorithmes basés sur des modèles nécessitent un clustering initial fait au moyen des données statistiques.

De plus, les résultats des approches basées sur la connaissance sont souvent utilisés dans les étapes de traitement ultérieures, telles que le filtrage des événements découverts à l'aide des statistiques d'occurrence des termes (Chun et al., 2004, hun et al., 2004). En outre, les méthodes traditionnellement basées sur des statistiques, telles que le balisage partiel de la parole, ont été améliorées en ajoutant des domaines de connaissances, ce qui est particulièrement utile, par exemple, dans le domaine biomédical où les mots communs sont souvent utilisés différemment, mais aussi dans de nombreux autres cas spécifiques à un domaine ou à une langue (Lee et al., 2003, ee et al., 2003). Ainsi, les modèles d'extraction des événements, basés sur les connaissances, peuvent être appris en appliquant des techniques d'apprentissage automatique telles que les champs aléatoires conditionnels et les machines à vecteurs de support (Hogenboom et al., 2016, ogenboom et al., 2016)(Shi et al., 2017, hi et al., 2017).

De plus, le contenu généré sur les médias sociaux joue un rôle si important dans la procédure de capture des événements d'actualité afin de classifier et de vérifier les histoires. Dans ce contexte, nous pouvons mentionner le travail dans (Boididou et al., 2018, oididou et al., 2018) où les auteurs ont comparé trois méthodes de vérification. La première technique consistait à utiliser des modèles textuels pour extraire des affirmations selon lesquelles un tweet était une fausse ou une vraie information. La deuxième méthode est basée sur l'emploi des informations dans lesquelles la crédibilité prouve la similarité du sujet d'actualité de l'événement extrait des tweets. La troisième technique appliquait un schéma d'apprentissage semi-supervisé qui affectait les décisions de deux classificateurs distincts de la crédibilité de l'actualité des événements. Les expériences effectuées sur différents types de données ont montré que la troisième approche était plus performante que les deux autres techniques en termes de précision et de rappel.

Dans (Xie et al., 2016, ie et al., 2016), TopicSketch a été utilisé pour détecter les événements en temps réel à partir des données à grande échelle. Les chercheurs ont développé, dans (Khan et al., 2013, han et al., 2013), une approche alternative basée sur les graphes pour extraire les nouvelles d'événements. Ils ont d'abord appliqué la LDA pour l'identification des sujets de discussion au sein d'un flux. Par la suite, un graphe avec les tweets dans chaque sujet a été construit en s'appuyant sur la co-occurrence de mots. Dans les premières études, le PageRank a été utilisé pour identifier, dans chaque sujet, les tweets saillants.
Dans des travaux plus récents (Meladianos et al., 2018, eladianos et al., 2018), une autre approche, basée sur les graphes, a été introduite pour analyser l'occurrence des changements rapides dans les graphes afin d'identifier les événements importants à incorporer dans le résumé.

Dans les méthodes d'extraction des événements hybrides, en raison de l'application des techniques pilotées par les données, la quantité des données requises augmente par rapport aux systèmes basés sur la connaissance, mais elle reste généralement inférieure à celle des techniques purement axées sur les données. La combinaison de l'approche base sur les connaissances et celles basées sur les données génèrent une expertise et une complexité

plus élevés que l'utilisation d'une une seule approche. Cela conduit éventuellement à des temps d'exécution plus longs.
D'autre part, la quantité des connaissances d'expert requises pour une découverte d'événements efficace est inférieure à celle des méthodes basées sur des modèles car le manque de connaissance du domaine peut être compensé par l'utilisation des méthodes statistiques. En ce qui concerne l'interprétation des résultats, il est plus difficile d'attribuer les résultats à des parties spécifiques de l'extraction des événements en raison de l'ajout des méthodes pilotées par les données. Cependant, l'interprétabilité bénéficie encore, dans une certaine mesure, de l'utilisation de la sémantique comme il est le cas dans les approches basées sur la connaissance.

3.2.4 Synthèse de l'état de l'art

Le tableau suivant synthétise l'état de l'art en comparant les méthodes proposées selon cinq critères : (i) la description du corpus d'évènements, (ii) la détermination du durée des évènements, (iii) l'analyse des groupes des utilisateurs des médias sociaux diffusant les évènements, (iv) la prise en compte de l'aspect social du flux de messages et (vi) la visualisation des évènements détectés.

Auteur/Date	**Description du corpus d'événements**	**Aspect temporel des événements**	**L'analyse des groupes diffusant les évènements**	**Prise en compte de l'aspect social**	**Vizuel**
Shamma et al.(Shamma et al., 2011, hamma et al., 2011)	Fréquence de chaque mot	Une tranche temporelle	non	non	non
Benhardus et Kalita (Benhardus and Kalita, 2013, enhardus and Kalita, 2013)	N-gramme	Une tranche temporelle	non	non	non
Lau et al.(Lau et al., 2012, au et al., 2012)	On-line LDA	Une tranche temporelle	non	non	non
Yuheng et al.(Yakushiji et al., 2000, akushiji et al., 2000)	ET-LDA	Indéfinie	non	non	non
Li et al.(Li et al., 2012, i et al., 2012)	Un ensemble de n-grammes	Un nombre fixe de tranches temporelles	non	non	non
Lei et al.(Shi et al., 2017, hi et al., 2017)	Mesure en cosines des mots	Une tranche temporelle	non	non	non
Khan et al.(Khan et al., 2013, han et al., 2013)	Regroupement de N-grammes	Indéfinie	Un graphe des sujets de tweet	non	non
Xie et al.(Xie et al., 2016, ie et al., 2016)	N-gramme détection de sujet	Tranche temporelle	non	non	non
Meladianos et al.(Meladianos et al., 2018, eladianos et al., 2018)	Un ensemble de N-grammes	Un nombre fixe de tranches temporelles	Construction de graphe	non	non

Description textuelle des évènements : La plupart des recherches existantes se limitent à prédéfinir un mot, un N-gramme et un ensemble de N-grammes ou un ensemble de mots-clés pour décrire les évènements. Ainsi, les approches existantes basées sur les données ou les connaissances ignorent l'intégrité des termes. Les informations sociales fournies par des utilisateurs non-experts sont généralement floues, peu claires, inexactes et incomplètes.

Aspect temporel des évènements : Il apparaît que les méthodes existantes se reposent sur une hypothèse commune, à savoir que tous les évènements détectés ont une même durée. Certaines méthodes considèrent que la durée de chaque évènement est égale à celle d'une tranche temporelle. Dans le premier cas, les auteurs partitionnent généralement les messages par tranches de 24 heures, tandis que, dans le second cas, la longueur des séquences est souvent fixée à 24 heures, mais la durée des tranches temporelles peut être ajustée afin de définir le niveau du détail voulu pour la mesure de similarité temporelle. Néanmoins, tous les évènements ne suscitent pas le même intérêt auprès des utilisateurs des medias sociaux et, par conséquent, la durée pendant laquelle ceux-ci continuent d'en discuter varie de quelques heures à plusieurs jours. En outre, en fixant la durée de tous les évènements, cette information est perdue et il paraît crucial d'estimer dynamiquement la durée de chaque évènement.

Aspect social : Les travaux issus de la littérature se concentrent sur le contenu textuel des messages échangés et négligent l'aspect social. Toutes les méthodes décrites dans cet état de l'art se focalisent sur la fréquence des mots ou des N-grammes des mots. En plus, les médias sociaux ont, pour vocation, de favoriser les interactions sociales entre leurs utilisateurs. Ces interactions sont notamment symbolisées au sein des messages publiés via des marqueurs spécifiques tels que les mentions. Par conséquent, il semble important de ne pas se concentrer uniquement sur l'aspect textuel des messages, mais également sur l'aspect social des flux de messages produits par les médias sociaux. L'analyse des groupes diffusant les évènements De nombreuses approches de détection d'événements ont ignoré la tâche d'analyse de communauté, ont utilisé des méthodes de graphes et ont appliqué une classification en ligne pour signaler des informations potentielles. En fait, la détection des communautés permet d'identifier des profils parfaits, effectuer des actions ciblées, mieux ajuster les recommandations, ainsi que de réorganisation et d'identifier les acteurs centraux / influents, etc.

Visualisation des événements : Les travaux issus de la littérature ont négligé le processus de visualisation des événements détectés. L'analyse visuelle facilite l'interprétation des événements détectés.

Le temps et la prédiction des liens sont un aspect très important pour l'analyse des communautés de détection d'événements. Cela nécessite clairement l'identification et le suivi des structures communautaires dans des réseaux sociaux en constante évolution, qui est le sujet de la section suivante.

3.3 Suivi de l'évolution des événements détectés dans le temps

Suivre et comprendre l'évolution des événements est une tâche difficile basée sur l'identification du principal événement pouvant survenir aux communautés et aux individus. Dans le monde réel, les membres des communautés ont tendance à se changer progressivement. Cet aspect dynamique est à l'origine de l'apparition de la prédiction de liens puisque cette tâche sert à évaluer l'évolution du réseau. Il est donc important non seulement de détecter les communautés, mais également de suivre les changements dans la composition des membres au fil du temps et de prédire l'apparition d'un lien entre deux entités.

La détection des événements est un problème très important pour les réseaux de télécommunication où un événement est, sans doute, une panne action nécessitant une intervention. Faire cette détection automatiquement est une exigence car, avec l'accroissement de la taille des réseaux, une surveillance humaine de tous les éléments est impossible. Cette notion des événements peut aussi s'appliquer à d'autres types de réseaux. Un événement, sur un réseau social, peut être la diffusion d'une nouvelle marquante. Cependant, sur un réseau de bibliométrie, il peut correspondre à l'apparition d'une nouvelle idée innovante. La détection des événements consiste souvent à identifier les points de rupture. Puisque les communautés servent à décrire la structure du réseau, il est alors naturel de vouloir étudier les moments où cette structure change particulièrement.

Plusieurs travaux portent spécifiquement sur le problème du suivi des communautés au fil du temps. Berger-Wolf et Saia (Berger-Wolf and Saia, 2006, erger-Wolf and Saia, 2006) ont proposé une méthode décrivant les communautés en tant que modèles locaux indépendants. Ils ont défini une communauté (ou un méta-groupe) comme une séquence de groupes ayant une similarité suffisamment élevée. La similarité entre deux groupes est le nombre des membres communs normalisé par la taille des deux groupes. Les caractéristiques des communautés ont été étudiées via des mesures statistiques basées sur la communauté, telles que le nombre de toutes les communautés possibles, leur taille et leur durée de vie. Berger-Wolf et Saia ont également introduit une approche pour examiner la survie des communautés en trouvant un ensemble critique de groupes dont le retrait ne laisse que des communautés de courte durée.

Spiliopoulou et al.(Spiliopoulou et al., 2006, piliopoulou et al., 2006) ont développé une méthode, appelé MONIC, permettant de suivre les communautés au fil du temps et utilise une fonction de similarité des groupes à des pas de temps différents. La fonction prend en compte le nombre des membres communs, la taille des groupes et la décroissance temporelle entre les groupes. Ensuite, deux groupes, appartenant à la même communauté, sont liés si leur similarité dépasse un certain seuil. La méthode définie non seulement enchaîne les groupes dans les communautés, mais détecte également la division et la fusion des communautés par un ensemble distinct de paramètres de seuil.

Dans (Van Laarhoven and Marchiori, 2014, an Laarhoven and Marchiori, 2014)(Viard et al., 2016, iard et al., 2016), les auteurs ont formulé le problème du suivi des communautés en tant que problème d'optimisation.

Dans le domaine de l'exploration de bases de données, Aggarwal et Yu (Aggarwal, 2011, ggarwal, 2011) ont proposé une méthode de détection des modifications en ligne et de synthèse des données pour les requêtes exploratoires hors ligne. ils sont concentré sur les applications de traitement analytique en ligne (OLAP(online analytical processing)) plutôt que sur la modélisation et la détection des communautés. Ainsi, Falkowski et al.(Falkowski et al., 2008, alkowski et al., 2008) ont utilisé une approche basée sur la théorie des graphes et composée en deux étapes pour détecter les communautés et suivre leurs changements au fil du temps. La première étape consiste à appliquer un algorithme de clustering des graphes. Ensuite, une mesure de correspondance a été employée.

Sun et al.(Sun et al., 2010, un et al., 2010) ont proposé GraphScope, une méthode de regroupement sans paramètre pour les réseaux dynamiques basée sur le principe de longueur de description minimale. Cette technique est un algorithme heuristique en ligne qui détecte les changements radicaux dans le flux d'entrée. L'output de cette dernière technique est une séquence codée de segments de graphe. Les instantanés de graphe, dans chaque segment, sont codés en utilisant le même groupement (cluster) de sommets. Pour chaque nouvel instantané de graphe, la méthode tente d'incorporer le nouvel instantané de graphe au segment en cours en regroupant, fractionnant et fusionnant des groupes de sommets du segment en cours de manière à réduire les coûts d'encodage. Ensuite, la methode de GraphScope compare, ce coût d'encodage au coût de démarrage d'un nouveau segment pour choisir le meilleur des deux. Les instantanés de graphe, dans chaque segment, peuvent être considérés comme un fichier (échantillon) et agrégés dans un seul graphe statique. En fait, la méthode ne détecte pas les communautés dynamiques en soi car elle ne permet pas, aux changements graduels, d'aboutir à une modification radicale rétrospective. En outre, elle ne vérifie pas l'existence et la survie des groupes changements radicaux entre deux segments adjacents. Par conséquent, il parait essentiel de rechercher des communautés dans chaque instantané du graphe Gi.

Palla et al.(Palla et al., 2007, alla et al., 2007) ont étudié la percolation et l'évolution des cliques sous un modèle restrictif qui considère l'évolution uniquement entre des pas du temps adjacents.

Suivant l'approche de Newman pour formuler la modularité du réseau (Newman and Girvan, 2004, ewman and Girvan, 2004) pour les réseaux statiques, Mucha et al. (Mucha et al., 2010, ucha et al., 2010) ont développé une fonction de qualité pour les communautés dynamiques basée sur un modèle nul en termes de stabilité sous-dynamique laplacienne. Ultérieurement, ils ont optimisé la fonction à l'aide de la technique de Louvain (Blondel et al., 2008, londel et al., 2008) et post-traité les résultats avec la permutation de noeud de Kernighan-Lin (Kernighan and Lin, 1970, ernighan and Lin, 1970).
Comme la détection de communautés statiques semble arrivée à maturité, la quasi-totalité des chercheurs ont eu l'idée de réutiliser les concepts des solutions statiques dans le cas dynamique. Ce qui a fait l'objet de plusieurs tentatives d'adaptation des algorithmes statiques aux réseaux dynamiques (Aynaud et al., 2013, ynaud et al., 2013). L'idée générale est de considérer un réseau dynamique comme une succession de réseaux statiques, appelés « instantanés », dont chacun représente l'image du réseau dynamique à un instant donné. Le principe de cette approche s'articule donc autour de trois étapes. Tout d'abord, dé-

composer l'évolution du réseau en plusieurs instantanés. Ensuite, appliquer un algorithme statique à chaque instantané. Enfin, faire correspondre les communautés trouvées dans un instantané avec celles de l'instantané précédent. La Figure suivante montre trois instantanés d'un réseau dynamique avec une association entre les communautés des différentes étapes. L'instantané t contient uniquement deux communautés colorées, respectivement, en bleu et en vert. La communauté colorée en bleu à l'instantané t se divise à t + 1en deux sous-communautés, colorées en bleu et en rouge, tandis que la communauté de couleur verte reste intacte. A l'instantané t + 2, la communauté de couleur bleu reste la même, celle en rouge et celle en vert se rétrécissent et une nouvelle communauté apparait, de couleur bleu clair.

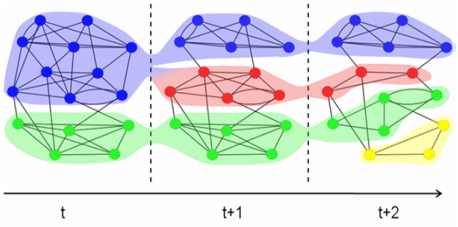


FIGURE 3.1: Illustration des approches par instantanés uniformes successifs (Aynaud et al., 2013, ynaud et al., 2013)

En effet, nous pouvons dire qu'en termes de discrétisation/continuité du temps, il n'existe que deux grandes approches, celle d'instantanés et celle dynamique. Les autres méthodes reprennent l'idée de l'approche par instantanés en ajoutant ou modifiant quelques notions pour réduire l'instabilité des algorithmes statiques.

3.4 Prédiction des liens

Le dynamisme et l'instabilité des interactions humaines dans les réseaux sociaux rend la détection des événements plus difficile. En fait, des nouvelles personnes et des nouvelles relations s'ajoutent à la communauté, discutant un évènement au fil du temps, ce qui empêche parfois un chercheur d'obtenir la vraie information concernant un événement. Ainsi, il est très important de prédire les nouveaux liens qui vont apparaître au futur.

Modélisation du problème de prédiction du lien Considérons un réseau social G (V ; E) à un moment particulier t où V et E sont respectivement des ensembles de noeuds et de liens. La prédiction des liens vise à prédire des nouveaux liens, des liens supprimés entre les noeuds pour un temps futur $t^{'}(t^{'} > t)$, des liens manquants ou des liens non-observés dans le réseau actuel. Etant donné des instantanés d'un réseau social de temps t_i au temps t_k, le probleme de prédiction des liens revient a déterminer l'endroit où

les liens seront ajoutées au réseau pendant l'intervalle de temps. La figure suivante résume la visualisation du problème de prédiction de liens(Tang, 2017, ang, 2017).

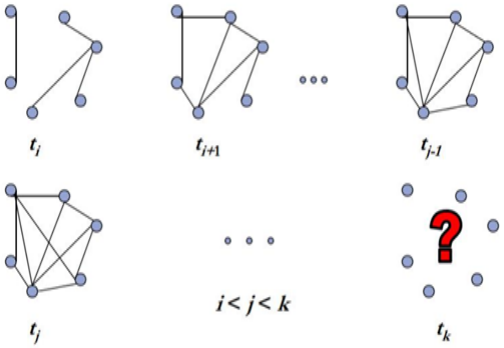


FIGURE 3.2: Processus visualisés du problème de prédiction de liens

Pour un réseau social initial, il existe deux manières de prédire l'évolution du lien : les approches basées sur la similarité et celles reposant sur l'apprentissage(Choudhury and Uddin, 2016, houdhury and Uddin, 2016). D'une part, une approche basée sur la similarité consiste à calculer les similarités sur des paires de noeuds non-connectés dans un réseau social, à savoir qu'elle repose sur des mesures permettant d'analyser la proximité des noeuds. Un score est attribué a chaque paire de noeuds potentiels (x ; y) ; un score élevé montre une probabilité plus élevée que x et y soient liés à l'avenir, et inversement. Subséquemment, une liste, classée par ordre décroissant de score, est obtenue et les liens en haut de la liste sont les plus susceptibles d'apparaître. D'autre part, une approche basée sur l'apprentissage traite le problème de prédiction de lien comme une tâche de classification binaire. Par conséquent, certains modèles d'apprentissage automatique typiques, tels que le classifieur et le modèle probabiliste, peuvent être utilisés pour résoudre ce problème. Chaque paire de noeuds non-connectés correspond à une instance avec des caractéristiques décrivant les noeuds et l'étiquette de classe. S'il existe un lien potentiel reliant une paire de noeuds, cette paire serait étiquetée comme positive, sinon elle est négative.

Techniques de prédiction du lien

La littérature est riche des travaux de prédiction du lien. Il existe des approches qui emploient les informations des noeuds, la topologie et la théorie sociale pour calculer les similitudes des paires de noeuds. De plus, malgré que les méthodes de prédiction des liens basées sur l'apprentissage soient plus complexes, elles sont établies à partir des fonctionnalités fournies par les métriques de base et les informations externes. Dans cette section, nous présenterons une revue systématique de ces métriques et méthodes de prévision des liens.

Métriques basées sur les noeuds Bhattacharyya et al.(Bhattacharyya et al., 2011, hattacharyya et al., 2011) ont défini un modèle arborescent de catégorisation multiple pour étudier les mots-clés des profils d'utilisateurs. Puis, ils ont spécifié la distance entre les mots-clés afin de déterminer la similarité entre deux utilisateurs. Leur observation la plus importante est que, à l'exception des amis directs, la similarité entre les utilisateurs est approximativement égale, quelles que soient les métriques topologiques. Ils ont montré aussi également que l'augmentation du nombre d'amis et de mots-clés réduit la similarité moyenne entre l'utilisateur et ses amis. Akcora et al.(Akcora et al., 2013, kcora et al., 2013) ont constaté que la plupart des profils d'utilisateurs des réseaux sociaux actuels sont manquants. Pour surmonter cette limitation en termes de similarité, ils ont proposé une méthode permettant d'inférer une partie des valeurs manquantes d'un profil d'étranger avant de calculer la similarité. Cette technique d'inférence est basée sur les informations de profil des amis communs et de schéma de vote à la majorité. Anderson et al.(Anderson et al., 2012, nderson et al., 2012) ont appliqué le chevauchement des intérêts des utilisateurs pour mesurer la similarité.

Les mesures basées sur les noeuds présentent l'inconvénient qu'elles utilisent principalement les attributs et les actions qui peuvent refléter les intérêts personnels et les comportements sociaux pour calculer les similitudes entre les paires de noeuds. Toutefois, les valeurs d'attributs sont des formes textuelles et les actions des utilisateurs dans les réseaux sociaux doivent être généralement utiles dans la prédiction du lien.

Métriques basées sur la topologie Liben-Nowell et Kleinberg ont discuté plusieurs métriques reposant sur les caractéristiques structurelles des graphes(Liben-Nowell and Kleinberg, 2007, iben-Nowell and Kleinberg, 2007). Après leur travail, de nombreuses métriques basées sur la topologie ont été proposées. Les caractéristiques structurelles de ces métriques leur permettent d'être divisées en métriques basées sur le voisinage, métriques basées sur le chemin d'accès et métriques basées sur le parcours aléatoire. Pour clarifier les descriptions suivantes, nous donnons d'abord des notations standard. Soit la matrice A la matrice d'adjacence d'un réseau social donné. Soit $\Gamma(x)$ l'ensemble des voisins du noeud x et $|\Gamma(x)|$ le nombre des voisins du noeud x.

— Mesures à base de voisinage

Voisins Communs (Common Neighbors (CN)) : Pour deux noeuds, x et y, le CN est défini comme le nombre des noeuds avec lesquels x et y ont une interaction directe. Un plus grand nombre de voisins communs facilite la création d'un lien entre x et y. Cette mesure est définie par la formule suivante :

$$CN(x, y) = |\Gamma(x) \cap \Gamma(y)| \tag{3.4}$$

Coefficient de Jaccard (Jaccard Coefficient (JC)) : Le coefficient de Jaccard normalise la taille des voisins communs. Il suppose des valeurs plus élevées pour les paires des noeuds partageant une proportion plus élevée de voisins communs par rapport au

nombre total de voisins qu'ils ont. Cette mesure est définie comme suit :

$$JS(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|} \tag{3.5}$$

Indice de Sorensen (Sorensen Index (SI)) : En outre, le fait de prendre en compte la taille des voisins communs indique également que les probabilités de liaison seraient plus faibles.

$$SI(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)| + |\Gamma(y)|} \tag{3.6}$$

Similarité de cosinus de Salton (SC) : SC est une métrique de cosinus commune permettant de mesurer la similarité entre deux noeuds x et y. Elle est définie comme suit :

$$SC(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{|\Gamma(x)|.|\Gamma(y)|}} \tag{3.7}$$

Hub promu (HP) : HP définit le chevauchement topologique des noeuds (x et y). La valeur HP est déterminée par le plus faible degré des noeuds.

$$HP(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\min |\Gamma(x)|.|\Gamma(y)|} \tag{3.8}$$

Hub Déprimé (HD) : Hub Déprimé (HD) : Zhou et al. ont proposé une métrique similaire à HP (Zhou et al., 2009, hou et al., 2009) dont la valeur est déterminée par le plus haut degré des noeuds.

$$HD(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{\max |\Gamma(x)|.|\Gamma(y)|} \tag{3.9}$$

Leicht-Holme-Nerman (LHN) : Cette métrique attribue une grande similarité aux paires de noeuds qui ont de nombreux voisins communs par rapport non pas au maximum de voisins possible, mais au nombre attendu de tels voisins.

$$LHN(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x)|.|\Gamma(y)|} \tag{3.10}$$

Dépendance des paramètres (PD) : Pour améliorer la précision de la prévision des liens populaires et impopulaires, la métrique PD est comme suit :

$$PD(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{(|\Gamma(x)| + |\Gamma(y)|)^{\lambda}} \tag{3.11}$$

λ est un paramètre libre. Lorsque $\lambda = 0$, PD dégénère en CN. Si $\lambda = 0.5$ et $\lambda = 1$, il dégénère en métrique de Salton et LHN, respectivement.

Coefficient Adamic-Adar (AA) : La mesure AA est formulée en fonction du coefficient de Jaccard. Cependant, dans ce contexte, les voisins communs qui ont moins de voisins ont plus de poids. Cette mesure est définie comme suit :

$$AA(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma(z)|} \tag{3.12}$$

Attachement préférentiel (PA) : La métrique PA indique que les nouveaux liens seront plus susceptibles de connecter des noeuds de degré plus élevé que ceux de degré inférieur. Elle est définie par :

$$PA(x, y) = |\Gamma(x)|.|\Gamma(y)| \tag{3.13}$$

Allocation des ressources (RA) : Cette métrique a été développée par Zhou et al.(Zhou et al., 2009, hou et al., 2009). Elle a une forme similaire à celle de AA. Les deux mesures (AA et RA) se différencient des autres métriques par le fait qu'elles non seulement utilisent des voisins directs, mais elles considèrent également les voisins des voisins. RA est définie comme suit :

$$RA(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{|\Gamma(z)|} \tag{3.14}$$

Supposons que n est le nombre moyen des voisins dans un réseau. Pour deux noeuds x et y, la complexité temporelle de la recherche de tous les voisins d'un noeud est $O(n)$ et la complexité temporelle du calcul de l'intersection ou de l'union de deux ensembles est $O(n^2)$. CN, SI, SC, HP, HD, LHN, PD ont une complexité temporelle $O(n^2)$) car ils doivent calculer une intersection de deux ensembles. La complexité temporelle de JC est $O(2n^2)$) puisqu'elle calcule une intersection et une union de deux ensembles. AA et RA doivent calculer une intersection de deux ensembles et trouver des voisins de voisins communs. Par conséquent, leur complexité temporelle est $O(2n^2)$). PA, ayant une complexité temporelle $O(n^2)$), a seulement besoin de trouver les voisins de x et y. Il convient de choisir les bonnes métriques en fonction des caractéristiques des réseaux sociaux car de nombreux résultats d'évaluation expérimentale ont montré qu'il n'existait pas de métrique absolument dominante pour différents jeux de données [102].

— Mesures basées sur le chemin

Outre que les informations sur les noeuds et les voisins, les chemins entre deux noeuds peuvent également être utilisés pour calculer les similarités des paires de noeuds.

Chemin local (Local Path (LP)) : La métrique LP étend l'utilisation des informations aux voisins situés à moins de trois distances du noeud actuel.
Evidemment, plus le chemin est court, plus le chemin sera pertinent. Il y a donc un facteur d'ajustement α qui a été appliqué dans cette méthode afin de fournir un résultat plus raisonnable. Il convient de mentionner que α devrait être un nombre $-1 \leq \alpha \leq 1$. La formule de cette métrique est représentée ci-dessous où A^2 et A^3 désignent les matrices d'adjacence ayant respectivement 2 et 3 longueurs de distance aux noeuds.

$$LP = A^2 + \alpha A^3 \tag{3.15}$$

Katz (KA) : Par rapport à LP qui considère les chemins jusqu'à 3 distances, la métrique Katz [88], qui compte tous les chemins entre deux noeuds, est basée sur l'ensemble des chemins. Comme pour les métriques CN présentées ci-dessus, en KA, plus le chemin est long, moins il a de poids pour les similarités finales. Ainsi, comme indiqué ci-dessous,$paths^l_{x,y}$ représente l'ensemble de tous les chemins de x à y de longueur l,

$0 \leq \beta \leq 1$ est le facteur de décroissance utilisé pour limiter la croissance en longueur et le plus petit β entraîne une réduction de l et un large degré de similarité.

$$katz(x,y) = \sum_{l=1}^{\infty} \beta^l . paths^l_{x,y} = \beta A + \beta^2 A^2 + \beta^2 A^2 + ... \tag{3.16}$$

Comparées aux métriques basées sur les noeuds et les voisins qui utilisent uniquement des informations topologiques locales, les métriques basées sur les chemins prennent en compte davantage d'informations topologiques. En d'autres termes, elles considèrent non seulement les voisins locaux, mais aussi une sorte d'informations globales importantes, à savoir les chemins entre les paires des noeuds. En fait, la complexité temporelle des métriques basées sur le chemin est supérieure à celles basées sur le voisin. Cependant, les chemins les plus longs ne sont pas toujours plus utiles que les chemins plus courts. Dans la justification théorique des heuristiques populaires de prédiction du lien de Sarkar et al. (Sarkar et al., 2011, arkar et al., 2011), l'auteur montre que les métriques prenant en compte des chemins plus longs ne sont utiles que si les chemins plus courts ne sont pas assez nombreux. Par conséquent, les métriques basées sur les chemins peuvent optimiser leurs performances en supprimant les chemins trop longs si les réseaux ont suffisamment de chemins plus courts.

— Mesures basées sur les marches aléatoires

Les interactions sociales entre les noeuds des réseaux sociaux peuvent également être modélisées par marche aléatoire qui utilise les probabilités de transition d'un noeud à ses voisins pour désigner la destination d'un promeneur aléatoire à partir du noeud actuel. Il existe des métriques de prédiction du lien permettant de calculer les similitudes entre les noeuds en fonction de la marche aléatoire.

Temps de frappe (Hitting Time (HT)) : Cette idée revient à la théorie des graphes. Pour deux sommets, x et y, dans un graphe, HT (x, y) définit le nombre attendu d'étapes requises pour une marche aléatoire de x à y. HT le plus court signifie que la paire de noeuds a un score de similarité plus élevé(da Silva Soares and Prudêncio, 2012, a Silva Soares and Prudêncio, 2012).

Marche aléatoire avec redémarrage : Random walk with Restart(RWR) Cet index est inspiré de l'algorithme PageRank. L'idée de base est de démarrer la marche à partir du noeud u pour atteindre itérativement un voisin aléatoire avec une probabilité p et revenir, ensuite, au noeud u avec une probabilité égale à 1.

SimRank (Jeh and Widom, 2002, eh and Widom, 2002) La métrique SimRank est définie de manière auto-cohérente, en supposant que deux noeuds sont similaires s'ils sont connectés à des noeuds similaires. SimRank ont mesuré la rapidité avec laquelle on s'attend à ce que deux surfeurs aléatoires se rencontrent au même sommet s'ils débutent individuellement aux sommets x et y et parcourent, de manière aléatoire, les arêtes du graphe inverse. La complexité de calcul de SimRank est $O(n^4)$, au pire de cas, où n est le nombre des sommets. Un tel coût limite son utilisation étendue pour les réseaux à grande échelle.

Index de chemin local (local path index(LP)) : Cet index assure une précision acceptable proportionnellement à la complexité de calcul puisqu'il considère le chemin local.

Marche aléatoire superposée (superposed random walk(SRW)) : Cet indice a été proposé dans (Liu and Lü, 2010, iu and Lü, 2010) où le marcheur aléatoire est libéré en continu au niveau du point de départ, ce qui entraîne une similarité élevée entre le noeud cible et le noeud voisin.

Métrique basée sur la théorie sociale L'analyse des réseaux sociaux est basée sur des théories sociales classiques telles que la communauté, la fermeture triadique, les liens forts et faibles, l'homophilie et l'équilibre structurel. Contrairement aux métriques précédentes, qui utilisent uniquement des noeuds et des topologies, les métriques de prédiction de lien basées sur la théorie sociale peuvent améliorer les performances en capturant des informations utiles supplémentaires sur les interactions sociaux. Valverde-Rebaza et Lopes ont associé la topologie aux informations sur la communauté en considérant l'intérêt et les comportements des utilisateurs et puis en prédisant les liens futurs sur Twitter (Valverde-Rebaza and de Andrade Lopes, 2013, alverde-Rebaza and de Andrade Lopes, 2013).

Liu et al. ont proposé un modèle de prédiction de lien basé sur des liens faibles et des centralisations à trois noeuds de voisins communs : degré, proximité et centralité entre les bases (Liu et al., 2013, iu et al., 2013). Chaque voisin commun joue un rôle différent pour la probabilité de connexion des noeuds en fonction de leur centralité. Le lien faible a été pris en compte pour améliorer la précision de la prédiction.

Méthodes basées sur l'apprentissage

Soit $u, v \in V$ les noeuds d'un graphe $G(V, E)$ et $l^{u,v}$ l'étiquette de l'instance de paires des noeuds (u, v). Nous pouvons avoir :

$$\begin{cases} +1 & si \quad u, v \in E \\ -1 & sinon \end{cases}$$

où une paire de noeuds peut être étiquetée positive s'il existe un lien reliant les noeuds ; sinon la paire est étiquetée négative. Il s'agit donc d'un problème de classification binaire typique qui peut être résolu par des nombreux modèles d'apprentissage par classification supervisée, à savoir les arbres de décision, les machines à vecteurs de support, Bayes, etc(Lu et al., 2010, u et al., 2010).

Pujari et Kanawati ont suggéré une nouvelle approche de prédiction de lien topologique dyadique appliquant un algorithme de choix social supervisé(Pujari and Kanawati, 2012, ujari and Kanawati, 2012). Ils ont employé ces données pour apprendre les poids associés à chaque entité calculée en fonction de la capacité de chaque attribut à prédire les liens observés. Ces poids ont ensuite été appliqués dans des algorithmes de choix sociaux supervisés pour prédire de nouveaux liens. Pour utiliser les réseaux sociaux auxiliaires ou

les réseaux de proximité disponibles, Lu et al. ont proposé une méthode d'apprentissage supervisé capable d'appréhender, de manière efficace, la dynamique des réseaux sociaux en présence des réseaux auxiliaires, puis de construire une grande variété de caractéristiques basées sur des chemins en utilisant plusieurs sources pour la prédiction des liens (Lu et al., 2010, u et al., 2010).

Dans un réseau social, une valeur de probabilité, par exemple une similarité topologique ou une probabilité de transition en marche aléatoire, peut être attribuée à un lien entre chaque paire de noeuds. Il s'agit d'un graphe probabiliste. Plusieurs techniques de prédiction de liens basées sur l'apprentissage ont été proposées en exploitant le modèle de graphe probabiliste. Des études ont suggéré que de nombreux réseaux présentent une structure hiérarchique dans laquelle les noeuds se divisent en groupes pouvant être subdivisés en sous-groupes des groupes, et ainsi de suite sur diverses échelles. Clauset et al. (Clauset et al., 2008, lauset et al., 2008) ont proposé un modèle pour déduire la structure hiérarchique du réseau et l'appliquer pour résoudre le problème de la prédiction des liens. En fait, un réseau hiérarchique est représenté par un dendrogramme appelé graphe aléatoire hiérarchique où N feuilles correspondent aux noeuds du réseau et chaque N-1 noeud interne correspond à une probabilité p_r où r est l'ancêtre commun le plus bas des deux noeuds. Soit G un réseau, D un dendrogramme, Er le nombre des arêtes, où r est l'ancêtre commun le plus bas des deux noeuds et Lr et Rr les nombres de feuilles des sous-arbres de gauche et de droite prenant racine en r. Alors, la probabilité du réseau est :

$$L(D, Pr) = \Pi_r Pr^{Er}(1 - Pr)^{LrEr-Rr} \tag{3.17}$$

La probabilité des noeuds internes est facile à déterminer en maximisant $L(D, Pr_{x,y})$. Pour prédire si une paire de noeuds non-connectés (x et y) sont connectés, un ensemble de dendrogrammes est, d'abord, échantillonné avec une probabilité proportionnelle à leur vrai semblance. Puis, la probabilité moyenne $p_{x,y}$ sur les dendrogrammes d'échantillon est calculée en faisant la moyenne de la probabilité correspondante $p_{x,y}$. Le modèle de graphe aléatoire hiérarchique est capable d'exprimer une structure d'association et de désassortiment et d'obtenir des prévisions précises pour un large éventail de réseaux. Toutefois, cela prend beaucoup de temps de calcul et s'applique généralement aux réseaux contenant des milliers de noeuds.

Wang et al. ont développée une méthode qui utilisait trois types de caractéristiques, à savoir les caractéristiques de probabilité de cooccurrence, les caractéristiques de topologie et les caractéristiques sémantiques, pour résoudre le problème de la prédiction de liaison (Wang et al., 2007, ang et al., 2007). Pour dériver la probabilité de co-occurrence (la probabilité de liaison entre deux noeuds), un modèle de graphe probabiliste local utilisant des champs aléatoires de Markov (MRF) a été introduit. Pour prédire si deux noeuds (x et y) seront liés, trois étapes sont nécessaires : (1) utiliser les informations topologiques pour identifier l'ensemble de voisinage central de x et y ; (2) sélectionner des ensembles d'éléments qui appartiennent entièrement à cet ensemble et les employer comme données d'apprentissage pour former un modèle probabiliste local (ici, le processus de formation est traduit en un problème d'optimisation d'entropie maximum) ; et (3) estimer les caractéristiques de probabilité de co-occurrence par inférence sur le modèle local. Ensuite, la régression logistique a été utilisée en tant que classificateur pour former les données

combinant ses trois types des entités.

Factorisation Matricielle Menon et al.(Menon and Elkan, 2011, enon and Elkan, 2011)ont traité la prédiction de lien en tant que problème d'achèvement de matrice et ont étendu la méthode de factorisation matricielle pour résoudre ce problème. Ils ont factorisé le graphe G en L ($U \wedge U^T$) pour $U \in R^{n,k}$, et $\wedge \in R^{k,k}$ où n est le nombre des noeuds et k le nombre des caractéristiques latentes. Chaque noeud x a un vecteur latent correspondant $u_x \in R^n$. Ensuite, le score prédit par le modèle pour la paire (x ; y) est L ($u_T^x \wedge u_y$).

Un grand nombre de méthodes de prédiction des liens sont limitées aux réseaux homogènes avec des arêtes et des noeuds de type unique, par contre les réseaux sociaux pratiques ont généralement plusieurs types de relations et de noeuds. Par exemple, un réseau bibliographique hétérogène contient des noeuds, tels que les publications, les auteurs et les lieux, et des contours comme le co-auteur, cite et workin. En outre, il est important de résoudre le problème de prédiction des liens dans de réseaux sociaux hétérogènes.

3.5 Conclusion

Dans ce chapitre, nous avons mené une étude du suivi de changement de communauté dynamique et du problème de prédiction des liens dans laquelle nous avons décrit les méthodes proposées dans ce travail de recherche. Nous remarquons, suite à cette étude de l'état de l'art, que les travaux de suivi de l'évolution et de prédiction des liens ont négligé la définition, la sélection et la collection des fonctionnalités appropriées des réseaux sociaux. Diverses techniques de prédiction des liens dans les réseaux sociaux n'ont pris en compte que les caractéristiques et les attributs topologiques. Peu de travaux ont considéré la structure communautaire dans les réseaux sociaux. En fait, l'intérêt du lien social sont différentes les unes des autres même si elles appartiennent au même groupe. Un modèle d'affinité pour quantifier, extraire et analyser les forces des liens sociaux au sein des groups est nécessaire. Ce problème est un point d'intérêt dans nos contributions présentées dans la partie suivante.

Contributions

Chapitre 4

Classification Floue des sentiments dans les réseaux sociaux

4.1 Introduction

Dans ce chapitre, nous présentons notre approche de classification des sentiments et des opinions exprimés dans les réseaux sociaux (Toujani and Akaichi, 2016, oujani and Akaichi, 2016) (?, ?). Cependant, les méthodes d'analyse des sentiments existantes se concentrent uniquement sur l'extraction des opinions positives et négatives. En fait, notre travail vise à extraire des opinions floues.

La signification du mot sentiment varie selon les domaines d'application. Comme dans les phrases interrogatives et conditionnelles, les mots de sentiment ne peuvent exprimer aucune émotion et les phrase présentant objectivement des faits n'expriment aucun sentiment.

Les textes en langage naturel ne peuvent pas être directement interprétés par un classificateur ou par les algorithmes de classification. Dans ce chapitre, nous détaillons dans la section 4.2 la représentation des corpus. La collecte de données brutes fait l'objet du setion 4.3. Le prétraitement linguistique des mots du texte est presente dans la section 4.4 et nous définissons dans la seconde 4.5 l'annotation des termes de sentiments et dans la section 4.6l'extraction des caractéristiques. Le classificateur Fuzzy Support Vector Machine est détaillé dans la section 4.6.

4.2 Représentation des corpus

Dans cette section, nous nous intéressons en premier lieu à la représentation des corpus.

Les algorithmes de classification sont incapables d'interprétés directement les textes en langage naturel. Les unités linguistiques spécifiques peuvent exprimer le sens d'un document aux caractéristiques plus ou moins élaborées obtenues par l'étude du corpus documentaire. On appelle « lemmes des mots » les premières unités linguistiques qui s'évoquent du sens. Pour reconnaître ces unités, représentées par des mots ou des phrases, il est nécessaire de réaliser un prétraitement linguistique des mots du texte. Dans la représentation par mots, l'unité linguistique réfère au mot tel qu'il se manifeste dans le document. Dans ce cas, l'extraction de chaque mot du texte est effectuée en prenant compte des séparateurs comme l'espace, la tabulation, la ponctuation, etc. Le nombre des mots dans un corpus peut être énorme, d'où l'importance de conserver un sous- ensemble de ces mots.

Pour se faire, nous commençons par la collecte des données brutes. Cette étape consiste à extraire la mise à jour des statuts des utilisateurs via une application Web appelée « I Told You» et les stocker dans une base de données personnalisée. Ensuite, nous mettons l'accent sur le développement du Lexicon en se concentrant sur le langage informel des réseaux sociaux en ligne. Pour cette raison, trois types de lexiques ont été créés : un lexique pour les acronymes sociaux, un lexique pour les émoticônes et un lexique pour les interjections. Puis, nous passons au prétraitement des données qui implique le nettoyage des données brutes et la suppression des mots inutiles qui n'ont aucune signification pour la tâche de classification.

4.3 Collecte de données brutes

L'une des fonctions des réseaux sociaux est d'assurer la protection des données de ses utilisateurs pour des raisons de confidentialité. Pour cela, la collecte et l'extraction, automatique des données à partir des réseaux sociaux sont des tâches difficiles et parfois impossibles. L'équipe des développeurs Facebook a fourni, au cours des derniers mois de 2010, des services Web tels que Graph API Explorer. Cet API permet d'extraire l'information, modifier l'information de l'entité, telle que le téléchargement des photos, l'écriture des commentaires (comment), prendre la liste des amis, etc. Selon ce service, en vous inscrivant pour un jeton d'accès, vous pouvez accéder aux informations sur les utilisateurs et leurs activités. Toutefois, l'utilisation de l'API Graph présente certaines limites. Parmi eux, on peut citer l'indépendance de la quantité d'informations accessibles de notre relation avec l'utilisateur ciblé. Une autre contrainte à prendre en compte, dans cette étape, est qu'avec Graph API nous ne pouvons pas retourner dans le temps pour une période spécifique. En d'autres termes, le nombre des commentaires extraits est limité. StatusHistory.com est un autre outil utilisé pour extraire les statuts exprimés sur Facebook. Le rôle de cette application est de récupérer les anciens statuts, de les trier par date, par nombre de commentaires ou par nombre de « likes » sur chaque statut. Cette application permet également d'avoir une vue générale et des statistiques sur le profil de l'utilisateur. Cependant, elle ne permet

pas de récupérer les statuts écrits en langue arabe. Pour faire face à toutes ces contraintes, nous avons utilisé l'application « I Told You » dont le rôle est de récupérer les anciens statuts de Facebook et de les exporter au format PDF, TXT, etc. Les statuts extraits sont souvent écrits en utilisant une combinaison de plusieurs langues. Cela produit une multitude de structures, de grammaires et de dictionnaires. De plus, les outils de fouille de texte et d'analyse des sentiments emploient généralement une seule langue à la fois. Par conséquent, il est nécessaire de se concentrer uniquement sur une langue précise pour pouvoir prétraiter efficacement les données, extraire les caractéristiques de publication et créer le modèle de classification. Ainsi, une traduction en anglais a été appliquée à tous les statuts extraits. Une fois les données brutes ont été traduites, nous effectuons un prétraitement des données afin de les nettoyer, supprimer les mots vides, extraire les fonctions pertinentes nécessaires à notre analyse de sentiments. Ces étapes sont expliquées dans la sous-section suivante.

4.4 Prétraitement du texte

Le processus de prétraitement des sentiments est similaire à celui du prétraitement de texte traditionnel effectué pour la classification du texte. Il a pour objectif de nettoyer les données considérées en supprimant les mots et les ponctuations n'ayant aucune influence sur la classification des sentiments, d'où l'amélioration des performances de la tâche de classification. Par conséquent, le prétraitement est une tâche primordiale qui implique deux tâches principales : génération de la liste des mots vides et racinisation.

— Suppression des mots vides Avant de passer à la stemmatisation, il est usuel d'utiliser une "stop-list" pour éliminer tout les mots qui ne participent pas activement à extraire le sens du document. Cette liste contient les pronoms, les articles et les mots trop fréquents pour être discriminants. Nous éliminons donc toutes les unités linguistiques non-discriminantes. Le risque de la stop-list est qu'on peut éliminer les mots qui pourraient être utiles pour la classification. En fait, les mots vides, tels que les pronoms, les prépositions, les conjonctions, sont ceux qui n'ajoutent pas de contenu significatif à l'ensemble de données. Par conséquent, leur suppression réduit considérablement l'espace des éléments dans les données de test et diminue la diversité des mots utilisés dans les statuts publiés. Des exemples de tels mots incluent "le", "un", "de", "et", "à", etc. Ainsi, la première étape du prétraitement consiste à supprimer ces mots vides.

Input	**Output**
The Tunisian women accept the polygamy!!! Tunisia after 14/01 from bad to worse : ((	~~The~~ Tunisian women accept the polygamy!!! Tunisia ~~after~~ 14/01 ~~from~~ bad to worse : ((

Table 4.1: Exemple de suppression des mots vides

— Stemming Au cas où les mots sont employés comme des unités linguistiques, plusieurs d'entre eux vont avoir des sens communs ou vont former simplement une autre configuration de conjugaison. Donner un sens différent à ces mots résulte de la redondance sans pertinence sémantique. Par conséquent, une stemmatisation doit être réalisée pour trouver la racine de chaque mot. Par exemple, les mots « love », « loved », « loving » and « loves » seront réduits à la racine du mot « love ». Cela affaiblit la diversité des conjugaisons dans lesquelles un mot peut apparaître et, par conséquent, diminue la quantité des données. Cette tâche peut être effectuée par plusieurs algorithmes comme le stemmer de Porter appliqué pour analyser morphologiquement le texte . Ce traitement repose sur un dictionnaire de suffixes utilisé pour l'extraction du radical de mot. On appelle lemmatisation le traitement basé sur un lexique qui exige une analyse plus complexe que la stemmatisation. Un lexique est généralement considéré comme un groupe de lemmes avec lequel nous référons au dictionnaire. La lemmatisation permet d'attribuer une entrée dans le lexique à chaque mot. Puisque de nombreux mots d'une graphie identique peuvent produire des différents lemmes, l'analyse morphologique n'est pas suffisante pour détecter le sens d'un message. Par conséquent, l'analyse syntaxique par la lemmatisation est nécessaire pour résoudre les ambigüités au niveau des opinions. Elle assure donc une analyse morpho-syntaxique.

L'analyse morpho-syntaxique consiste à :
- Attribuer à chaque jeton un ordre de catégorie grammatical (verbe, nom, adjectif, etc.) ;
- Segmenter du le texte en phrases à l'aide des marques de ponctuation : ".", " ?" et " !" ;
- Appliquer un "tokenization" : (Dans ce qui suit nous parlons des jetons plutôt que des mots). Cette étape a pour but de segmenter les phrases en une séquence de jetons utilisant des caractères non-alphabétiques, "espace", "tabulation", ".", "]", etc.

4.5 L'extraction des caractéristiques

L'une des principales étapes de l'extraction des caractéristiques est l'application du modèle n-gramme (Carpenter, 2005, arpenter, 2005) qui est une séquence contiguë de n-mots . Le modèle n-gramme consiste à extraire un sac de mots qui représentent le champ du texte. Chaque caractéristique correspond à un mot n, avec la valeur « true » si cette fonctionnalité est présente et « false » sinon. Dans ce travail, nous utilisons des unigrammes, des bigrammes, des trigrammes, et des parties du discours comme caractéristiques. L'unigramme est un mot unique, tandis que le bigramme est une paire de mots qui apparaissent l'un à côté de l'autre, etc. Comme les unigrammes sont le type de texte le plus typique, nous les employons dans notre travail pour composer le message en mots.

4.6 L'analyse des sentiments

Nous avons classifié les mots en catégories (positives, négatives) faisant référence à SentiStrength qui calcule approximativement la robustesse des opinions ou des émotions positives et négatives dans différentes langues. La Figure suivante présente les principales étapes d'extraction des sentiments.

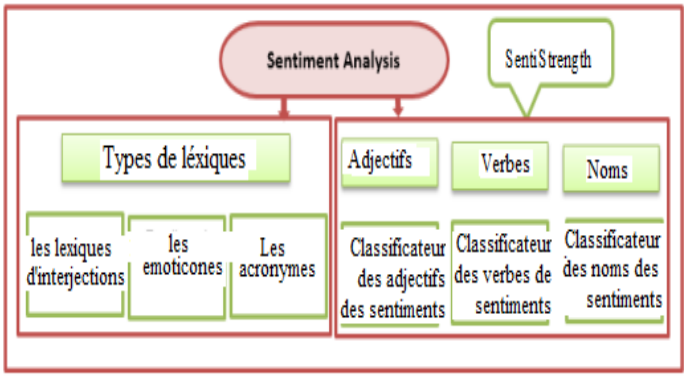


FIGURE 4.1: Analyse des sentiments

Nous avons développé trois types de lexiques : les émoticônes, les acronymes et les interjections. Les lexiques ne sont pas exhaustifs et nous n'avons pas abordé tous les émoticônes, acronymes et interjections possibles. Nous ne couvrons que les plus utilisés sur Facebook et sur notre ensemble de données extraites.

moticone	**Sentiment**
:) or :-)	Positif
:p or :-p	Positif
:(or :-(	Négatif
:/	Négatif
:'(	Négatif
;)	Positif
:>	Positif
:* or :-*	Positif
:<	Négatif
:-x	Positif

TABLE 4.2: Lexique des émoticônes

— Lexique des acronymes Ce lexique désigne les acronymes les plus utilisés sur les réseaux sociaux (Tableau 2).

Acronyme	**Sentiment**
LOL	Positif
GR8	Positif

TABLE 4.3: Lexique des acronymes

— Lexique des interjections
Le lexique d'interjection contient les principales interjections utilisées dans les réseaux sociaux (Tableau 3).

Interjection	**Sentiment**
Wow, waw	Positif
Haha, hihi, hehe	Positif
Oh dear	Negatif
Thank you	Positif
Help	Positif
No way	Negatif
Oy	Positif

TABLE 4.4: Lexique d'Interjections

L'extraction des caractéristiques, dans notre contexte, désigne le processus d'extraction des principales sentiments du texte considéré. Pour qu'un algorithme d'apprentissage automatique fonctionne correctement, il est essentiel de disposer des fonctionnalités descriptives du texte. Il est important donc d'identifier les mots qui expriment un sentiment. En outre, nous visons à étiqueter les statuts qui nous sont injectés dans le SVM flou proposé. Pour atteindre cet objectif, l'outil OpinionFinder semble très efficace pour extraire automatiquement des phrases subjectives et des expressions de sentiment. En fait, nous classons les mots en catégories (positifs et négatifs) faisant référence à SentiStrength qui calcule approximativement la robustesse des opinions ou des émotions positives et négatives dans différentes langues. Cependant, SentiStrength est incapable de déterminer une opinion floue. Pour cela nous proposons un algorithme de détection d'opinion floue.

— Identification d'opinion floue
Par conséquent, l'opinion floue peut être incluse dans le processus de classification pour les raisons suivantes :
- La signification du mot sentiment varie selon les domaines d'application.
- Comme dans les phrases interrogatives et conditionnelles, les mots de sentiment

ne peuvent exprimer aucun sentiment.
- Les phrases sarcastiques reposent sur des opinions dures tel qu'il est le cas dans les discussions politiques.
- L'objectivité s'exprime en phrases sans mot de sentiment.
Nous introduisons un algorithme pour la détection des opinions floues. Il prend comme entrée une liste des termes exprimant un sentiment vague. Par exemple, on peut mentionner incertain, indécis, vague, confus, hésitant, équivoque, ambigu, douteux et suspect. Cette liste sera stockée dans un thésaurus (THS). Ensuite, nous récupérons les différentes opinions floues dans un nouveau dictionnaire de termes flous nommé (DFT). En fait, une opinion floue est un terme défini en fonction d'une valeur particulière d'une variable linguistique incluant des informations numériques et linguistiques. Les termes linguistiques sont considérés comme des ensembles flous de l'univers du discours qui utilise des valeurs numériques. Si la liste (THS) est déjà extraite, nous générons une base de la variable linguistique pour analyser les sentiments. Ainsi, la base de la variable est utilisée plus tard pour identifier les concepts flous. L'algorithme proposé pour la détermination des opinions floues prend en entrée tous les termes extra-dits de sentiment. Subséquemment, il vérifie l'existence des termes dans la base des termes flous. En fin, la liste des sentiments flous est générée en sortie. L'algorithme de détection des termes flous est présenté dans le pseudo-code suivant :

Algorithm 4.1 Algorithme : Détection des opinions floues

Require: thesaurus d'opinions (THS) et le dictionnaire des termes flous nommés (DFT)
Ensure: liste des opinions floues (FOL)
for $c_i \in$ THS do
$FOL = \emptyset$
if $c_i \in$ DFT Then ;
FOL $\cup c_i$;
Else
FOL=FOL$\setminus \cup c_i$;
END IF

La classification floue (Fuzzy Support Vector Machine(SVM))

Pour introduire l'aspect flou et classifier les sentiments flous, nous choisissons d'appliquer SVM qui est un classificateur fiable et plus performant que les autres techniques de classification. En fait, SVM est utilisé pour classer les données en deux catégories et repérer une création hyper plane créant la plus grande marge entre les deux classes. En

SVM classique, chaque point d'entrée est étiqueté en tant qu'opinion positive ou négative. Cependant, dans nos recherches, nous introduisons l'aspect flou. En d'autres termes, certains points d'entrée peuvent ne pas être affectés exactement à l'une de ces deux classes. La SVM flou proposée est explorée pour les scénarios multi-class (positives, négatives et floues). En fait, nous proposons un raffinement qui vise à considérer les variables des ressorts qui font référence à la fonction d'appartenance à des classes floues. Cette dernière est définie comme suit :

$$\epsilon_i = \gamma_{w,b} - (Y_i f_{w,b}(X_i))+ \tag{4.1}$$

où $(X)+$ représente la partie positive de X ∈ R. Cette partie est liée au concept de marge. Elle mesure l'emplacement d'un couple (X_i,Y_i) par rapport à la marge du classificateur $f_{w,b}$. Par conséquent, trois cas sont introduits :

- si $\epsilon_i > \gamma_{w,b}$, le couple (X_i,Y_i) est mal classé par $\gamma_{w,b}$;
- si 0< $\epsilon_i < \gamma_{w,b}$, le couple est bien placé mais proche de l'hyperplan(la distance de l'hyperplan est inférieure à la marge)
- ϵ_i= 0 , alors le couple est dans sa classe majoritaire (la distance l'hyperplan est supérieur à la marge).

La figure ci-dessous montre ces variables de ressorts. En effet, l'introduction de ces variables

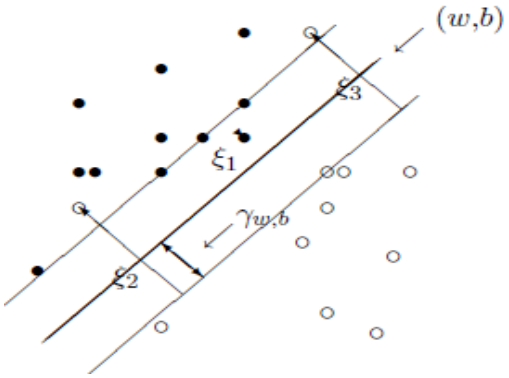


FIGURE 4.2: Introduction des variables de ressorts

de ressorts permet de relâcher la formulation mathématique de notre SVM floue.

Formulation mathématique de SVM floue Soit S l'ensemble des échantillons d'apprentissage :

$$S = \{(X_i, y_i, s_i), i = 1, ...N \in R^N\} \tag{4.2}$$

où chaque $X_i \in R^N$ est un échantillon d'apprentissage et $y_i \in -1, 1$ représente son étiquette de classe. s_i={1,2,...,N} correspond à notre classe floue qui satisfait $\sigma < s_i < 1$ avec σ une constante suffisamment petite $\sigma \geq 0$.
On note un ensemble comme suit :

$$Q = \{X_i | (X_i, y_i, s_i) \in S\} \tag{4.3}$$

Clairement, il contient deux classes ; chacune comporte un exemple de point X_i avec $y_i = 1$ désignant cette classe par C^+ :

$$C^+ = \{(X_i|(X_i \in S)\}ety_i = 1 \tag{4.4}$$

L'autre classe englobe un exemple de point X_i avec y_i = -1 désignant cette classe par C^- :

$$C^- = \{(X_i|(X_i \in S)\}andy_i = -1 \tag{4.5}$$

En effet,

$$Q = C^+ \cup C^- \tag{4.6}$$

Ainsi, la formulation mathématique de notre problème de classification des sentiments et du classificateur linéaire de la machine à vecteur de support flou peut être obtenue en résolvant le problème d'optimisation suivant :

$$\min 1/2\|W\|^2 + C\sum_{i=1}^{n} s_i\epsilon_i \tag{4.7}$$

$$y_i(w^T\Phi(X_i + b)) \geq 1 - \epsilon_i \tag{4.8}$$

$$\epsilon_i \geq 0, i = \{1, ...N\} \tag{4.9}$$

où C désigne une constante, la classe floue s_i est le terme d'opinion s_i libellé flou envers une classe, le paramètre ε_i représente le pourcentage de classe des opinions et le terme s_i ϵ_i est consideré comme une mesure du degré de flou. Il est à noter qu'un s_i plus petit peut réduire l'effet du paramètre s_i ϵ_i dans le problème de classification. Ainsi, la formulation mathématique du SVM flou est résolue en intégrant le datamining dans les méta-heuristiques.
A part la combinaison des méta-heuristiques l'hybridation entre les méthodes d'optimisation exactes et métaheuristiques, la fusion des méta-heuristiques et datamining a été aussi proposée dans plusieurs travaux. Par conséquent, la découverte du savoir et l'apprentissage automatique améliorent la métaheuristique. La figure suivante illustre l'intégration des méta-heuristiques dans notre SVM flou(Toujani Radhia and Jalel, 2015, oujani Radhia and Jalel, 2015).

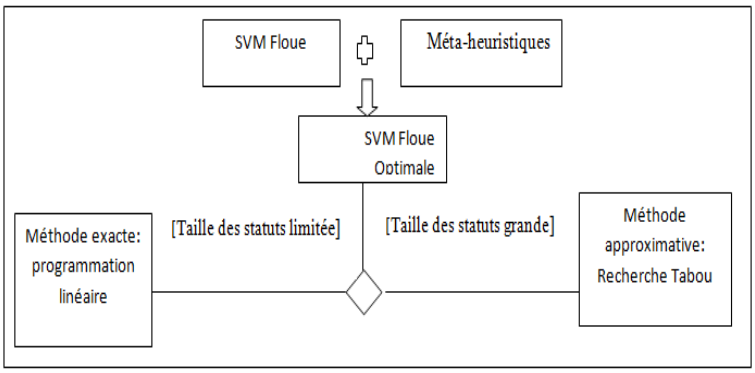


FIGURE 4.3: Classification des sentiments optimale

Pour une longueur d'un statut ≤20 mots, la méthode exacte de programmation linéaire est appliquée via le solveur Cplex. Cependant, si la taille de notre corpus devient importante (nombre de bigramm>20), la solution exacte n'est plus générée. Pour cette raison, nous appliquons la méta-heuristique Recherche Tabou.

Modélisation du SVM Floue par la Recherche Tabou La recherche taboue est une méthode métaheuristique capable de guider les méthodes de recherche locales traditionnelles pour échapper aux optima locaux à l'aide de la mémoire adaptative. En fait, nous étions motivés à utiliser la recherche tabou en raison de sa capacité de mémoire adaptative. La mémoire adaptative de la recherche tabou permet, aux procédures de recherche, d'explorer un ensemble beaucoup plus vaste de solutions candidates dans l'espace des solutions d'une manière économique et efficace. En outre, la fonctionnalité adaptative de mémoire de recherche tabou semble particulièrement adaptée à l'apprentissage automatique et au SVM pour lesquels il existe de nombreuses structures graphiques d'optima locales.

Comme indiqué dans la figure 4.4, dans sa forme la plus simple, notre algorithme de recherche de tabou adaptable commence par une solution réalisable. Il choisit la meilleure partition selon une fonction d'évaluation précise, tout en veillant à ce que la méthode ne revienne pas dans les partitions générées précédemment. Ceci est accompli en introduisant des restrictions taboues sur les mouvements possibles afin de décourager le retournement et, dans certains cas, la répétition des mouvements sélectionnés. En fait, la liste taboue contenant ces mouvements interdits est appelée fonction de mémoire à court terme. Elle fonctionne en modifiant la trajectoire de recherche pour exclure les déplacements conduisant à de nouvelles solutions contenant des attributs appartenant aux solutions précédemment visitées dans un horizon temporel régi par la mémoire à court terme. Des fonctions de mémoire intermédiaires et à long terme peuvent également être incorporées pour intensifier et diversifier la recherche.

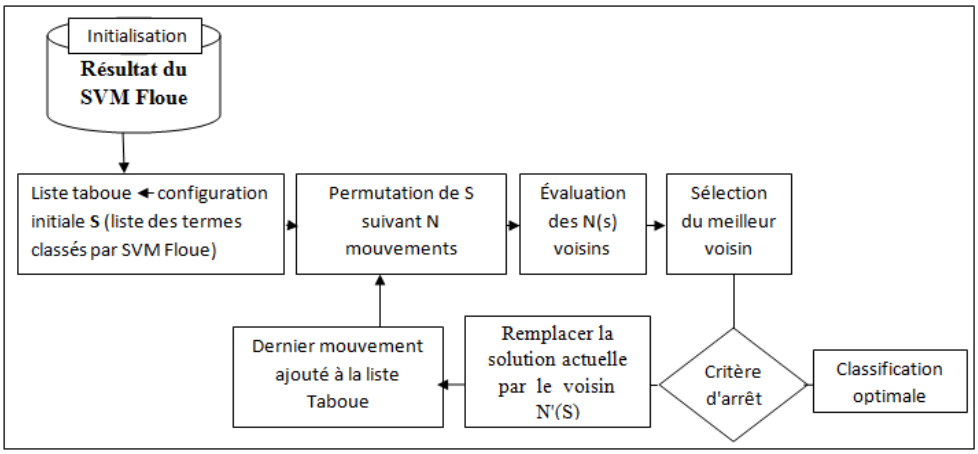


FIGURE 4.4: Organigramme de l'algorithme de classification des sentiments basé sur la recherche taboue

Initialisation Le résultat du classificateur SVM flou constitue la configuration initiale de notre classification optimale. Cette dernière solution initiale est stockée dans la mémoire taboue adaptative.

Voisinage et mouvement La fonction de voisinage N(s) est définie sur l'espace de recherche. En fait, les stratégies de recherche taboue sont basées sur des considérations à court terme où N'(s) est typiquement un sous-ensemble de N(s). La classification taboue sert à identifier les éléments de N(s) exclus de N'(s). Toutefois, dans les stratégies de recherche taboue qui incluent des considérations à plus long terme, N'(s) peut également être élargi pour inclure des solutions non-trouvées normalement dans N(s), telles que les solutions trouvées et évaluées lors des recherches précédentes, ou celles identifiées comme des voisins de haute qualité des solutions passées. Ainsi, la recherche taboue est considérée comme une méthode de voisinage dynamique. L'opération de retournement d'une variable pour obtenir une configuration voisine est appelée un mouvement. Alors, notre classification taboue conduit la recherche à la limite de critères d'agrégation trouve les solutions situées à l'intérieur de l'ensemble réalisable. La nouvelle solution actuelle est choisie parmi les solutions efficaces locales de l'ensemble généré qui n'appartient pas à la liste taboue. L'ensemble des solutions potentiellement efficaces est mis à jour. Si le nombre d'itérations est fixe, l'ensemble des solutions potentiellement efficaces ne serait pas amélioré et l'algorithme s'arrête.

Liste taboue Le rôle de la liste taboue (liste des derniers mouvements) est d'empêcher la recherche du cycle à court terme. La mémoire basée sur la récence à court terme interdit le contournement d'un voisinage local dans l'espace de solution en définissant les derniers mouvements T comme tabous. Les mouvements effectués récemment sont stockés dans la liste taboue des mouvements. En fait, la taille de la liste taboue dépend du nombre de mouvements dans la liste. Cette liste est générée d'une façon circulaire décrite comme suit : On élimine le plus vieil Tabou et on insère la nouvelle solution à la fin du cycle selon le principe du premier entré premier sorti. Les informations de récence stockées dans la liste taboue présentent la configuration de la solution. En outre, nous définissons les solutions Taboues en fonction des "transformations" ou "mouvements" qui les ont engendrées. Ainsi, nous gardons une liste des |T| derniers mouvements effectués et nous nous interdisons également d'effectuer leurs mouvements inverses.

Définition : le voisinage N(i,k) d'une solution k a l'itération i est défini en supprimant, de N(i), certaines solutions récemment visitées. N(i,k) sera alors égal à N(i)-T :

$N(i,k) =$ { j / $\exists$ m $\in$ M(i) *et* $i \oplus m$ }

Plus la valeur de T est grande, plus les mouvements et les solutions restent tabous. Ainsi, la liste taboue est un ensemble de solutions créées récemment par un opérateur de transformation. La mémoire basée sur la fréquence à long terme permet de mener des recherches dans les voisinages les plus prometteurs. Cependant, étant donné un problème d'optimisation, il est souvent difficile, voire impossible, de trouver une valeur qui empêche le cycle de patitionnement et ne limite pas excessivement la recherche de toutes les occurrences du problème d'une taille donnée. Un moyen efficace de résoudre cette difficulté consiste à utiliser une liste

taboue de taille variable. Chaque élément de la liste lui appartient pour un nombre d'itérations limité par des valeurs maximales et minimales données.De plus, il n'y a aucune garantie sur le non-bouclage du processus de recherche. Au cas d'attribution du caractère "Tabou" à des solutions non encore visitées, une solution optimale risque de ne pas être éventuellement visitée. Afin de palier à ce dernier problème, une relaxation du critère Tabou est recommandée pour des solutions jamais visitées. Cette relaxation est connue sous le nom du « critère d'aspiration ».

Critères d'aspiration Les restrictions de tabou font l'objet d'une exception importante. Quand un mouvement tabou a une évaluation suffisamment attrayante et qu'il en résulterait une solution meilleure que celle qui a été visitée jusqu'à présent, sa classification taboue peut être remplacée. Une condition permettant un tel remplacement est appelée « critère d'aspiration ». L'autorisation des exceptions dans la liste des tabous serait possible si de tels changements mènent à des solutions prometteuses. Un mouvement tabou m appliqué à une solution i serait accepté s'il engendre une solution qui n'a jamais été visitée auparavant. Plus généralement, un mouvement m sera choisi si son niveau d'aspiration a(i,m) est meilleur qu'une certaine limite A(i,m) fixée au préalable.
Définition : soit A(i,m) l'ensemble des classes d'opinions préférées, si a(i,m) $\in$ a(i,m), l'état i sera accepté malgré son statut Tabou.

Intensification et diversification *Intensification* : Afin d'intensifier la recherche dans les régions prometteuses, nous devons d'abord revenir à l'une des meilleures solutions de classification. Ensuite, la taille de la liste des tabous peut être simplement réduite pour un certain nombre d'itérations. Notre classification taboue effectue une intensification basée sur une mémoire à long terme. Chaque solution ou mouvement est caractérisé par un ensemble de composants mémorisés. Pendant la phase d'intensification, les solutions sont évaluées en tenant compte de la quantité des bons composants. Cette mémoire à long terme est considérée comme une sorte de processus d'apprentissage. Ainsi, nous choisissons, comme forme d'intensification, de sauvegarder, dans une liste, les meilleures solutions rencontrées lors du processus de recherche. Ces solutions sont qualifiées comme solutions élites.
Diversification : L'un des principaux problèmes de toutes les méthodes basées sur la recherche locale est que ces techniques ont tendance à passer la majeure partie de leur temps, sinon tout, dans une partie restreinte de l'espace de recherche. La conséquence négative, de ce fait, est qu'on peut ne pas explorer les parties les plus intéressantes de l'espace de recherche et aboutir ainsi à des solutions qui sont encore loin d'être optimales bien que de bonnes solutions puissent être trouvées. La diversification est un mécanisme algorithmique qui tente de résoudre ce problème en forçant la recherche dans des zones antérieures de l'espace de recherche. Le principe de diversification introduit repose sur la mémoire à long termes ou mémoire de fréquence permettant d'enregistrer le nombre d'itérations indiquant les partitions obtenues à chaque niveau hiérarchique.

Critère d'arrêt La recherche est terminée une fois qu'un nombre maximum prédéfini d'itérations k de l'algorithme de recherche taboue principale a été atteint. Un critère d'arrêt alternatif pourrait être un nombre prédéterminé de tentatives visant

à définir la même solution dans la liste des solutions taboues en tant que nouvelle solution actuelle. Cela indique que la recherche a été capturée dans un optimum local, mettant ainsi fin à la recherche.

La recherche des tabous fait partie des méthodes de recherche locales. Notre algorithme repose donc sur la détermination d'un bon voisinage. Le pseudo-code ci-dessus souligne notre approche de l'application de la méthode Taboue pour le SVM flou.

Algorithm 4.2 Algorithme : SVM Flou basé sur la Recherche Taboue

Require: les opinions détectées par le classificateur SVM Flou
Ensure: des classes d'opinions optimales

```
Initialisation
s_0 = l_k /*génération de la solution initiale*/
T=φ
Créer le nombre d'itérations L
Trouver la solution initiale s_0
Recherche
Tant que <critère de résiliation non satisfait> faire
Sélectionnez s en N (s)
s = échange (s) et
si (f(s)<c) alors
f= f(s)
Choisir le meilleur coup autorisé
Mettre à jour la liste des tabous avec le mouvement choisi
Effectuer le mouvement choisi en s
Mettre à jour le jeu de configuration non dominante avec s
Fin Tant que
```

Dans sa forme la plus simple, notre algorithme de recherche de tabou adaptable commence par une solution réalisable et choisit le meilleur mouvement, selon une fonction d'évaluation, tout en prenant des mesures pour faire en sorte que la méthode ne revisite pas une solution précédemment générée. Ceci est accompli en introduisant des restrictions taboues sur les mouvements possibles afin d'éviter la répétition des mouvements sélectionnés. La liste des tabous contenant ces mouvements interdits est appelée « fonction de mémoire à court terme ». Elle opère en modifiant la trajectoire de recherche pour exclure les mouvements conduisant à des nouvelles solutions contenant des attributs appartenant aux solutions précédemment visitées dans un horizon temporel régi par la mémoire à court terme. En outre, nous incorporons des fonctions de mémoire intermédiaires et à long terme pour intensifier et diversifier la recherche. À partir d'une solution initiale, notre algorithme consiste à passer de solution en solution en évitant de revenir à une solution déjà rencontrée. Plus précisément, nous définissions un voisinage V(S) pour chaque classification S.

Nous supposons, en outre, que nous ayons, à chaque itération, une liste T de toutes les classifications rencontrées depuis le début de l'exécution de notre algorithme.

Ainsi, à partir de la configuration actuelle S, on choisit dans V(S) une solution S_0 qui maximise notre fonction objectif, à savoir une marge maximale entre les classes d'opinions.
Exemple numérique :
$\max 9X1 + 6X2 + 5X3 + 3X4 + 5X5 + 2X6 + 5X7 + 10X8 + X9 + 6X10$
$\max 15X1 + 12X2 + 8X3 + 11X4 + 14X5 + 10X6 + 14X7 + 9X8 + 11X9 + 14X10$
Sous contrainte
$\begin{cases} 200X1 + 150X2 + 160X3 + 500X4 + 260X5 + 140X6 + 200X7 + 400X8 + 250X9 + 120X10 <= 1250 \\ 14X1 + 83X2 + 94X3 + 3X4 + 26X5 + 14X6 + 65X7 + 8X8 + 11X9 + 87X10 <= 250 \end{cases}$

$X_i \in 0, 1$
$i = 1, ..., 10$ **Étape0 : Initialisation :**
P1 :$max 9X1 + 6X2 + 5X3 + 3X4 + 5X5 + 2X6 + 5X7 + 10X8 + X9 + 6X10$ s.t
$200X1+150X2+160X3+500X4+260X5+140X6+200X7+400X8+250X9+120X10 <=$ 1250
$14X1 + 83X2 + 94X3 + 3X4 + 26X5 + 14X6 + 65X7 + 8X8 + 11X9 + 87X10 <= 250$
P2 :$Max 15X1 + 12X2 + 8X3 + 11X4 + 14X5 + 10X6 + 14X7 + 9X8 + 11X9 + 14X10$ s.t
$200X1+150X2+160X3+500X4+260X5+140X6+200X7+400X8+250X9+120X10 <=$ 1250
$14X1 + 83X2 + 94X3 + 3X4 + 26X5 + 14X6 + 65X7 + 8X8 + 11X9 + 87X10 <= 250$
$X_i \in 0, 1$
$i = 1, ..., 10$
$X = 1; 1; 0; 0; 1; 0; 0; 1; 0; 1 et X = 1; 0; 0; 0; 1; 1; 1; 0; 1; 1$ sont des solutions optimales du S0.
Étape1
$S0 = (1; 1; 0; 0; 1; 0; 0; 1; 0; 1); (1; 0; 0; 0; 1; 1; 1; 0; 1; 1)$
$k = 1$
$Max 24X1 + 18X2 + 13X3 + 14X4 + 19X5 + 12X6 + 19X7 + 19X8 + 12X9 + 20X10$ s.t
$9X1 + 6X2 + 5X3 + 3X4 + 5X5 + 2X6 + 5X7 + 10X8 + X9 + 6X10 > 28$
$15X1 + 12X2 + 8X3 + 11X4 + 14X5 + 10X6 + 14X7 + 9X8 + 11X9 + 14X10 > 64$
$200X1+150X2+160X3+500X4+260X5+140X6+200X7+400X8+250X9+120X10 <=$ 1250
$14X1 + 83X2 + 94X3 + 3X4 + 26X5 + 14X6 + 65X7 + 8X8 + 11X9 + 87X10 <= 250$
$X_i \in 0, 1$
$i = 1, ..., 10$
En résolvant le problème ci-dessus, nous obtenons $X = (1; 1; 0; 0; 1; 1; 0; 0; 1; 1)$.
Par conséquent, $S = (1; 1; 0; 0; 1; 1; 0; 0; 1; 1); (1; 1; 0; 0; 1; 0; 0; 1; 0; 1); (1; 0; 0; 0; 1; 1; 1; 0; 1; 1$
$k = 2$
$Max 24X1 + 18X2 + 13X3 + 14X4 + 19X5 + 12X6 + 19X7 + 19X8 + 12X9 + 20X10$ s.t
$9X1 + 6X2 + 5X3 + 3X4 + 5X5 + 2X6 + 5X7 + 10X8 + X9 + 6X10 > 28$
$15X1 + 12X2 + 8X3 + 11X4 + 14X5 + 10X6 + 14X7 + 9X8 + 11X9 + 14X10 > 64$
$Y1 + 9X1 + 6X2 + 5X3 + 3X4 + 5X5 + 2X6 + 5X7 + 10X8 + X9 + 6X10 > 29$
$Y2 + 15X1 + 12X2 + 8X3 + 11X4 + 14X5 + 10X6 + 14X7 + 9X8 + 11X9 + 14X10 > 76$

$Y1 + Y2 = 1200X1 + 150X2 + 160X3 + 500X4 + 260X5 + 140X6 + 200X7 + 400X8 +$
$250X9 + 120X10 <= 1250$
$14X1 + 83X2 + 94X3 + 3X4 + 26X5 + 14X6 + 65X7 + 8X8 + 11X9 + 87X10 <= 250$
$X_i \in 0, 1$
$i = 1, ..., 10$
Nous obtenons $X = 1; 0; 0; 0; 1; 0; 1; 1; 0; 1$ et $S = (1; 0; 0; 0; 1; 0; 1; 1; 0; 1); (1; 1; 0; 0; 1; 1; 0; 0; 1; 1); (1; 1; 0; 0; 1; 0; 0; 1; 0; 1);$
$k = 3$
$Max 24X1 + 18X2 + 13X3 + 14X4 + 19X5 + 12X6 + 19X7 + 19X8 + 12X9 + 20X10$
s.t
$9X1 + 6X2 + 5X3 + 3X4 + 5X5 + 2X6 + 5X7 + 10X8 + X9 + 6X10 > 28$
$15X1 + 12X2 + 8X3 + 11X4 + 14X5 + 10X6 + 14X7 + 9X8 + 11X9 + 14X10 > 64$
$Y1 + 9X1 + 6X2 + 5X3 + 3X4 + 5X5 + 2X6 + 5X7 + 10X8 + X9 + 6X10 > 35$
$Y2 + 15X1 + 12X2 + 8X3 + 11X4 + 14X5 + 10X6 + 14X7 + 9X8 + 11X9 + 14X10 > 66$
$Y1 + Y2 = 1200X1 + 150X2 + 160X3 + 500X4 + 260X5 + 140X6 + 200X7 + 400X8 +$
$250X9 + 120X10 <= 1250$
$14X1 + 83X2 + 94X3 + 3X4 + 26X5 + 14X6 + 65X7 + 8X8 + 11X9 + 87X10 <= 250$
$X_i \in 0, 1$
$i = 1, ..., 10$
$y_1 y_2 \in 0, 1$

En résolvant le problème pour $M = 10$, on obtient $X = 1; 1; 0; 0; 1; 1; 0; 0; 1; 1$. La solution générée est un sous-ensemble de S et l'algorithme est terminé. Solutions finaux sont $S = (1; 0; 0; 0; 1; 0; 1; 1; 0; 1); (1; 1; 0; 0; 1; 1; 0; 0; 1; 1); (1; 1; 0; 0; 1; 0; 0; 1; 0; 1); (1; 0; 0; 0; 1; 1; 1; 0; 1; 1)$

4.7 Conclusion

Dans ce chapitre, nous avons proposé une démarche pour la classification des opinions dans les réseaux sociaux. La coordination entre l'apprentissage automatique et le traitement du langage naturel a été étudiée afin d'extraire l'opinion des statuts. A part les classes positives et négatives, la classification introduite ajoute la classe des opinions Floues. Par conséquent, nous avons décrit un SVM floue.

Pour assurer l'optimalité du classificateur définie, nous nous avons combiné la métaheuristique et la méthode d'apprentissage automatique pour résoudre notre problème d'analyse des sentiments. Pour une petite taille de statut, nous avons développé une méthode exacte et avons suivi, plus précisément, le principe de la programmation linéaire en utilisant le solveur Cplex. Cependant, si la taille du statut est grande (plus que 20 mots), nous avons choisi d'appliquer une métaheuristique et de suivre le principal de recherche taboue.

Dans ce chapitre, nous avons détaillé le processus de classification des sentiments pour chaque utilisateur des réseaux sociaux. Les résultats obtenus à l'issue de ce processus sont exploites pour analyser l'opinion des groupes dans les réseaux sociaux et détecter les

communautés partageant les mêmes opinions qui seront étudiées dans le chapitre suivant.

Chapitre 5

Classification hiérarchique mixte des communautés dans les réseaux sociaux

5.1 Introduction

Les réseaux ou les graphes sont généralement utilisés pour modéliser divers systèmes complexes tels que les systèmes physiques, biologiques, sociaux et politiques. Le noeud représente, dans un réseau complexe, un membre. Cependant, une arête montre les relations existant entre deux membres. En fait, les réseaux complexes sont caractérisés par une structure de communauté ayant une valeur théorique et pratique importante. La détection de ces communautés est largement appliquée dans divers domaines du monde réel, comme la détection des gènes dotés de la même fonctionnalité dans un réseau biologique (Girvan and Newman, 2002, irvan and Newman, 2002), la détection des acteurs d'influence dans un réseau politique (Ratkiewicz et al., 2011, atkiewicz et al., 2011), etc. En outre, les réseaux montrent la structure et les relations hiérarchiques des communautés. Cette hiérarchie simplifie la division du réseau en quelques grandes communautés qui peut être subdivisées en unités plus petites. En fait, la définition de la structure hiérarchique et modulaire appropriée des réseaux complexes joue un rôle primordial pour la compréhension des systèmes complexes où la détection de communauté est considérée comme un problème de partitionnement de graphe NP-complet.

Dans ce chapitre, nous présentons les étapes principales du processus de détection des communautés proposé, à savoir la modélisation des communautés dans les réseaux sociaux sous forme de graphe dans la section 5.2 et la classification hiérarchique mixte (Toujani and Akaichi, 2019a, oujani and Akaichi, 2019a) dans la section 5.3 .

5.2 Modélisation et hiérarchisation des communautés

Dans cette section, nous présentons la formalisation de la structure du réseau, décrivons le treillis d'un patrtionnement d'une communauté et définissons les fonctions utilisées pour modéliser le réseau et pondérer le graphe.

5.2.1 Formalisation

Nous nous intéressons, dans ce travail, à la détection des communautés en fonction de la structure hiérarchique du réseau. Soit $SN = (V, E, \mu)$ où SN correspond au graphe modélisant le réseau social, V dénote les noeuds représentant les membres des réseaux sociaux et μ indique les poids des arêtes. Le problème de détection de communauté hiérarchique défini consiste à trouver le partitionnement de SN $P(SN) = \{P_1, P_2, ..., P_n\}$ fourni avec les opérateurs hiérarchiques à savoir l'agrégation et la décomposition. Dans notre travail, le terme communauté ou groupe réfère au nombre des noeuds regroupés dépendants et partageant fortement les mêmes opinions.

5.2.2 Treillis et partitionnement

Soit $W(SN)$ l'ensemble des partitions de SN. $P_1, P_2, ..., P_k$ est une partition en k sous-ensembles de SN si et seulement si :

- $\forall\ p_i \in W(SN)$ et $p_i \neq \emptyset$ pour $i = 1, ..., k$
- $\forall\ i \neq j\ ,\ p_i \cap p_j \neq \emptyset$
- $\bigcup_{i=1,...,k}\ p_i = SN$

Soit $P(SN) = \{p_1, p_2, ..., p_s\}$ and $G(SN) = \{g_1, g_2, ..., g_r\}$, Notre méthode de détection de communauté consiste à trouver un partitionnement hiérarchique fourni par la relation $\leq$: "être plus fine que" définie de la manière suivante.

$$P \leq G \Leftrightarrow \forall p_i \in P, \exists g_j \in G : p_i \subseteq g_j \tag{5.1}$$

Par ailleurs, si $P < G$, chaque classe de p est l'union des classes g qu'elle contient :$\forall g_j \in G : g_j = \cup_{i:p_i \subseteq g_j} p_i$

Ainsi, P détermine une partition des classes de S. La relation $\leq$ est un ordre partiel, c'est à dire qu'elle vérifie les trois conditions suivantes :

- Elle est réflexive :

$$\forall Z \in W(SN) ona Z \leq Z \tag{5.2}$$

- Elle est antisymétrique :

$$P \leq G, G \leq P \Rightarrow P = G \tag{5.3}$$

— Elle est transitive :

$$P \leq Z, Z \leq G \Rightarrow P \leq G \tag{5.4}$$

$P = \underbrace{p_1, p_2, ..., p_a}_{g_1} + \underbrace{p_{a+1}, ..., p_b}_{g_2} + \underbrace{p_{b+1}, ..., }_{g_3} + \underbrace{p_u, ..., p_s}_{g_r}.$

$G = g_1, g_2, g_3, ..., g_r.$

P et G ont un infimum **(noté** $P \cap G$**)** et un supremum **(noté** $P \cup G$**)** définis comme :

$$(P \cap G) \subseteq P\text{et}(P \cap G) \subseteq G, \text{et} \forall D \subseteq P\text{et} D \subseteq G \Rightarrow D \subseteq (P \cap G) \tag{5.5}$$

$$(P \cup G) \supseteq P\text{et}(P \cup G) \supseteq G, \text{et} \forall D \supseteq P\text{et} D \supseteq G \Rightarrow D \supseteq (P \cup G) \tag{5.6}$$

D désigne tout élément (sous-partition) de SN. Par conséquent, la relation entre $\cap$, $\cup$ et $\subseteq$ est donnée par :

$$P \subseteq G \Leftrightarrow P \cup G = P \Leftrightarrow P \cap G = G \tag{5.7}$$

Ainsi, les partitions de $W(SN)$ sont fournies avec la relation $\subseteq$ et les opérateurs $\cap$, $\cup$ constitue un treillis. La figure 5.1 décrit la construction du pseudo-treillis de notre partitionnement hiérarchique pour $P = \{m_1, m_2, m_3, m_4\}$.

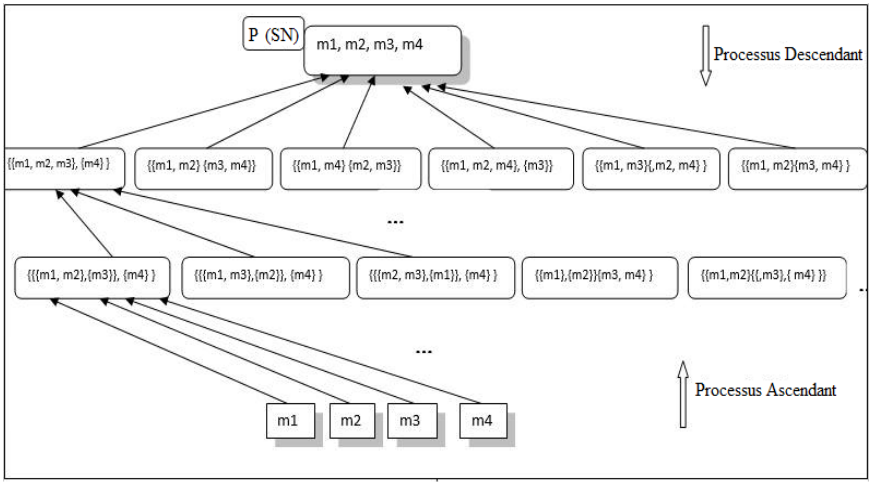


FIGURE 5.1: Exemple des treillis obtenus avec $P(SN) = \{m_1, m_2, m_3, m_4\}$

5.2.3 Fonctions Utiles

Pour faciliter la modélisation des communautés, nous utilisons quelques fonctions qui servent à définir le poids des arrêts du graphe décrivant les groupes des utilisateurs des réseaux sociaux.

— Covariance(Talbi, 2013, albi, 2013) :
Pour mettre l'accent sur l'indépendance et la similarité de la hiérarchie de la structure communautaire dans le réseau social, nous mesurons la covariance qui

qualifie le degré d'indépendance entre deux variables. En fait, nous appliquons cette mesure de covariance sur les liens reliant les noeuds du réseau. En effet, un lien existe entre deux noeuds ayant un grand nombre de voisins communs. Dans notre contexte, deux noeuds ont la même tendance d'opinions. Si un lien connecte deux noeuds avec un petit nombre de voisins communs, il présente alors une faible covariance. Pour calculer la covariance, il faut calculer la matrice d'adjacence A du réseau. Si deux noeuds, Vi et Vj, de ce réseau sont directement connectés, $A[i,j]$ serait égale à 1, sinon $A[i,j] = 0$. La covariance d'un lien est représentée par $E_{i,j}$ est définie par l'équation suivante :

$$cov(E_{i,j}) = \frac{1}{n}\sum_{k=1}^{n}(A[i,k] - \overline{A}[i])(A[j,k] - \overline{A}[j]) \tag{5.8}$$

Où $\overline{A}$ représente la moyenne du vecteur $A[i]$, n est le nombre de noeuds du réseau et k désigne un compteur indiquant le $i^{ème}$ noeud du réseau

— Similarité de Jaccard (Liben-Nowell and Kleinberg, 2007, iben-Nowell and Kleinberg, 2007) :
Le coefficient de Jaccard est le rapport entre le cardinal (taille) de l'intersection des ensembles considérés et le cardinal de l'union des ensembles. Nous utilisons ce coefficient pour mesurer le chevauchement des opinions contenant deux noeuds voisins connectés à un lien donné. En fait, la fonction de Jaccard introduite est axée sur la classification des sentiments (opinions positives, négatives ou floues) échangées entre les utilisateurs des réseaux sociaux.
Nous commençons par l'élaboration du graphe pondéré modélisant le réseau. Ainsi, nous associons, à chaque noeud, un ensemble d'opinions déterminé à l'aide de la classification floue introduite dans (Toujani Radhia and Jalel, 2015, oujani Radhia and Jalel, 2015). Ensuite, nous pondérons chaque arête par la valeur de Similarité de Jaccard. Ainsi, étant donné deux opinions extraites des noeuds voisins Op (V_i) et Op(V_j), nous utilisons le coefficient de jaccard pour mesurer la similarité sémantique et la similarité structurelle lorsque nous remplaçons Op (V_i) par N_i et Op (V_j) par N_j. Ce coefficient est déterminé comme suit :

$$JS(OpV_i, OpV_j) = \frac{Op(V_i) \cap Op(V_j)}{Op(V_i) \cup Op(V_j)} \tag{5.9}$$

En fait, si deux noeuds ont plus de voisins communs, le chevauchement des voisinages serait important et les liens entre eux seraient plus écrasants.
$Score_{importantOp}$: Le score d'importance des noeuds : L'importance d'un noeud dans un réseau quantifie la capacité de ce noeud dans la diffusion d'informations au reste du graphe. En fait, l'identification des utilisateurs influents dans les réseaux sociaux ou les communautés est utile pour suivre, par exemple, la diffusion des informations et pour tirer profit des discussions entre les membres des groupes. De plus, découvrir des utilisateurs influents garantit la diffusion des informations précises et des conseils positifs.
Dans le cadre de notre travail, pour mesurer le degré d'importance des noeuds, nous nous concentrons à la fois sur la dépendance et la similarité des utilisateurs

des réseaux sociaux. Nous combinons, d'une part, la moyenne de la covariance des liens reliant deux noeuds, ce qui montre l'indépendance des noeuds. D'autre part, nous nous focalisons sur la similarité basée sur les opinions obtenues en appliquant le coefficient de Jaccard. Par conséquent, pour obtenir la moyenne arithmétique entre la covariance et la similarité de jaccard, il est nécessaire de normaliser les intervalles de ses deux dernières fonctions. Pour ce faire, nous introduisons une conversion des valeurs de covariance. Soit $[-1,1[=[-1,0[\cup[0,1[$. Après avoir calculé la valeur de covariance, nous définissons la transformation suivante : $\mu(cov(E_{i,j})) = \begin{cases} -(cov(E_{i,j}))if cov(E_{i,j}) \in [-1,0[\\ cov(E_{i,j})if cov(E_{i,j}) \in [0,1[\end{cases}$
Ensuite, l'équation suivante montre les formules utilisées pour calculer le score d'importance des noeuds :

$$Score_{importantOp} = \frac{\mu(cov(E_{i,j})) + JS((V_i),(V_j))}{2} \tag{5.10}$$

Une fois nous attribuons les valeurs de $Score_{importantOp}$: aux different arêtes du graphe, nous procédons à son partitionnement. Dans la section suivante, nous détaillerons les étapes principales de notre processus de détection hiérarchique des communautés.

5.3 Approche de classification hiérarchique des communautés

Afin de trouver une structure de communauté hiérarchique significative, nous formulons le problème de détection des communautés comme un problème de classification hiérarchique. Malgré le grand nombre des algorithmes de détection hiérarchique de communauté proposés, certains problèmes restent non résolus. Par exemple, concernant les réseaux à grande échelle, la majorité des algorithmes existants se caractérisent par leur faible efficacité et leur complexité temporelle élevée. Ainsi, les algorithmes hiérarchiques utilisés pour la détection de communauté avec une qualité élevée produisent des réseaux de petite ou moyenne taille qui ne peuvent pas être appliqués efficacement dans les grands réseaux en raison de leur complexité considérable. En outre, en appliquant une méthode de classification unique, il est possible d'obtenir des hiérarchies communautaires déséquilibrées pouvant être dominées par une seule grande communauté absorbant des sommets uniques un par un. Ceci résulte en un déséquilibre de la hiérarchie à tous les niveaux hiérarchiques.

Nous pouvons conclure que, malgré l'importance des algorithmes hiérarchiques de division et d'agglomération, la combinaison des deux n'a pas été examinée de manière approfondie dans les travaux existants pour obtenir une structure de communauté hiérarchique complète. Pour cette raison, notre principal objectif, dans cette étude, est d'obtenir une hiérarchie des communautés qui révèle plus clairement et plus complètement la structure

de la communauté en réseau. La figure 5.2 présente le processus proposé de la classification hiérarchique mixte.

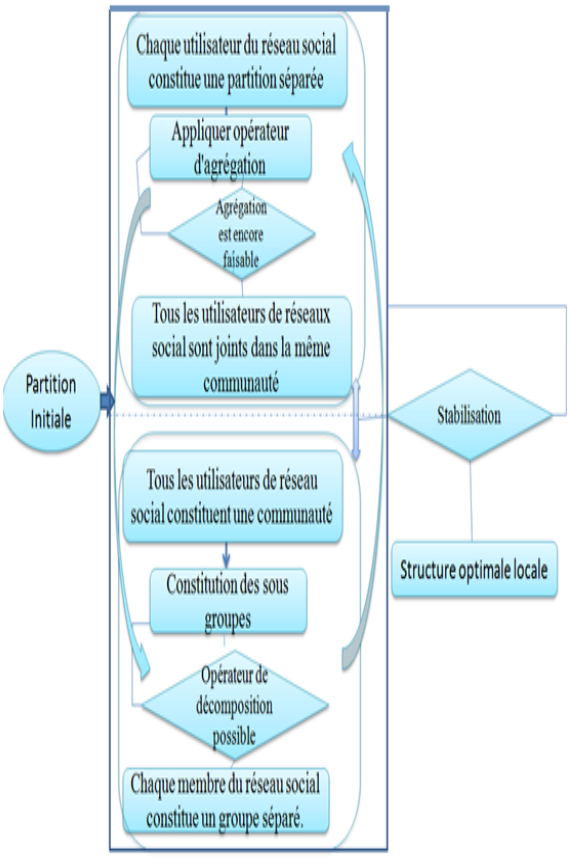


FIGURE 5.2: Architecture de l'approche de classification mixte.

Comme la méthode introduite mixte suppose l'existence d'une partition initiale, nous commençons par la génération de cette partition. Ensuite, nous appliquons successivement l'opérateur d'agrégation et l'opérateur de décomposition jusqu'à l'obtention d'une partition stable.

5.3.1 Partitionnement initial pour la détection hybride de communauté hiérarchique

L'algorithme hiérarchique hybride introduit suppose l'existence d'une partition initiale, tandis que l'entrée de l'algorithme affecte significativement sa sortie. Par conséquent, nous visons à éviter la génération aléatoire de cette partition initiale en assimilant le problème de détection de communauté à un problème d'optimisation combinatoire, plus

particulièrement à un problème de sac à dos multi-objectifs (en anglais, Multi Objective Knapsack Problem (MOKP). Nous proposons également la métaheuristique de Recherche Taboue pour résoudre ce problème(Toujani and Akaichi, 2017, oujani and Akaichi, 2017).

Modélisation du partition initiale

Nous considérons un graphe pondéré $G = (V, E, \omega)$ représentant les intérractions entre les membres du réseau social initial où V est l'ensemble des noeuds, E dénote l'ensemble des arêtes ou des liens et ω correspond au Coût de communication (Cc) entre les noeuds.

En fait, ce coût souligne la popularité et la réputation des statuts. En fait, plusieurs paramètres prouvent la popularité et la réputation des pages sociales. Citons à tires d'exemple, le nombre de commentaires sur les messages. Evidemment, si un utilisateur commente un message, il est influencé par le diffuseur de ces messages. En outre,le nombre des vues, des j'aime, de partage des messages jouent un rôle primordial dans la détermination de la popularité et la réputation d'un message. Le nombre de personnes suivant ce qu'un utilisateur partage indique sa popularité et son influence.
Par conséquent, l'importance des messages est définie par la somme des nombres de signaux sociaux (aimer, partager et commenter) qui lui sont attribués. Ainsi, le coût de la communication d'un poste s'obtient comme suit : $Cc = \sum like_{i,j} + \sum like_{j,i} + \sum comment_{i,j} + \sum comment_{j,i} + \sum share_{i,j} + \sum share_{j,i}$.
Nous modélisons mathématiquement le réseau social sous forme d'une matrice symétrique A de coût de communication précis :

- $A_{i,j} = \alpha$ s'il existe un coût de communication (Cc) α entre V_i et Vj.
- sinon $A_{i,j} = 0$

MOKP pour la modélisation de la partition initiale

Dans le tableau 5.1, nous présentons l'analogie entre le principe du partitionnement introduit en se basant sur MOKP et le MOKP de base.

Etant donné que chaque membre du réseau social peut appartenir a plusieurs sous-communautés (pour assurer le chevauchement du structure communautaire), la contrainte de capacité, dans notre cas, vise que le coût de communication entre les membres places dans un sous-groupe ne doit pas dépasser le coût de la communication totale du groupe. Soit $C = \underbrace{\{\{m_1, m_2\}\}}_{c_1} + \underbrace{\{\{m_3, m_4\}\}}_{c_2} + \underbrace{\{\{m_i, m_j\}\}}_{c_k}$ la communauté globale constituée par k sous-communautés détectées. Le coût de communication globale de la communauté est égale à : $\sum_{i=1}^{k} Cc(c_K)$ chaque sous-groupe a :

- Un poids $Cc_{(m_i,m_j)}$exprimant le coût de communication entre les membres m_i et m_j de chaque sous-partition c_k,
- Un profit Q(c_k) mesurant la modularité de chaque sous-partition c_k. Ainsi, Q est une mesure qui présente la différence entre la fraction des arêtes intra-communauté et la même valeur attendue pour une fraction aléatoire des arêtes [1]. Ainsi, la mo-

MOKP de base	**MOKP pour la structure de partition initiale**
C_i : Capacité du sac à dos_i	Cc (C_i) est le Coût de communication général de la communauté C_i
$p_{i,j}$ profit d'élement j selon Knapsack i	Modularité de la communauté sous-détectée c_k selon toute la communauté C
$q_{i,j}$ coût d'élement j selon Knapsack i	φ (ck) Mesure de la conductance de la communauté sous-détectée, en fonction de toute la communauté C
$\omega_{i,j}$ Poids d'élement j selon Knapsack i	$Cc_{i,k}$ coûts de communication du membre des réseaux sociaux m_i par rapport au groupe sous-détecté c_k
$(x_1, x_2, ..., x_n) \in [0, 1]$ est le vecteur de décision qui montre si l'élément est sélectionné ou non	$(x_1, x_2, ..., x_n) \in [0, 1]$ est le vecteur de décision qui indique si les membres sont sélectionnés ou non)

TABLE 5.1: La modélisation de la partition initiale par MOKP assimilée à celle de MOKP de base

dularité d'un graphe par rapport à certaines mesures de division montre à quel point la division est bonne ou comment sont séparés les différents types de vertex les uns des autres. Elle est définie comme :

$$Q = \frac{1}{2m} \sum_{i,j} (A_{i,j} - \frac{k_i k_j}{2m}) \delta(c_i, cj) \tag{5.11}$$

où m est le nombre d'arêtes, A_{ij} correspond à l'élément de la matrice d'adjascence A de la ligne i et de la colonne j, k_i représente le degré de i, k_j désigne le degré de j.

— Un coût $\phi(c_k)$ présenté par la valeur de conductance de chaque sous-partition c_k.Cette dernière mesure (pour un cluster) désigne le rapport entre la connectivité interne et externe [20]. En effet, la conductance a des valeurs dans la plage (0, 1), les valeurs les plus basses indiquant une meilleure qualité de clustering. Soit un graphe G et une partition (S, S), la conductance est définie comme :

$$\phi(S) = \frac{\sum i \in S, j \notin SA_{i,j}}{\min(a(S), a(\overline{S}))} \tag{5.12}$$

où A_{ij} sont les entrées de la matrice d'adjacence A de G et $a(S) = \sum_{i \in S} \sum j \in GA_{ij}$.

Notre problème de détection de partition initiale vise à répondre à la question suivante :

Quels sont les membres qui doivent être intégrés dans le meme groupe afin de maximiser la modularité et minimiser le coût de communication ente les sous-partitions tout en ne dépassant pas le coût de communication globale ?
Par conséquent, nous présentons la formulation mathématique du partitionnement basé sur MOKP comme suit :

La fonction objective s'écrit alors :
$Z(C) = f(C) + g(C)$ avec

$$f(C) \doteq \max \sum_{i=1}^{k} x_i * Q(c_k) \tag{5.13}$$

$$g(C) \doteq \min \sum_{j=1}^{k} x_i * \phi(c_k) \tag{5.14}$$

sous contraintes :

$$\vee m_i \in c_k \sum_{i=1}^{n} Cc(m_i)x_i \leq Cc(C), i \doteq 1, 2, ..., n. \tag{5.15}$$

$x_i = \{x_1, x_2, ..., x_n \in \{0, 1\}\}$ x_i représente notre variable de décision avec :

- $x_i = 1$ si m_i est choisit pour l'inclure dans c_k
- 0 si non

Recherche Taboue pour le partitionnement basé sur MOKP

Nous appliquons la méthode Taboue pour résoudre le partitionnement initial modélisé sous forme d'un problème du sac à dos. Dans le tableau suivant, nous présentons les notations utilisées dans l'algorithme tabou introduit.

Symboles	**Description**
s_0	La solution initiale
L	Le nombre d'itérations
N(s)	Le voisinage d'une solution s
T	La liste taboue
s	La solution actuelle
c	Capacité
f(s)	Fonction d'évaluation de s

TABLE 5.2: Notations utilisées dans l'algorithme de Recherche Taboue

Ainsi, Le pseudo-code de ce dernier algorithme est défini comme suit :

Algorithm 5.1 Algorithme : Recherche Taboue pour la structure de communauté basée sur MOKP

Require: l_k k sous communautés modélisées par MOKP
Ensure: partition initiale s
1: Initialisation
2: $s_0 = l_k$ /*génération de la solution initiale*/
3: T=ϕ
4: Créer le nombre d'itérations L
5: Trouver la solution initiale s_0
6: Recherche
7: Tant que <critère de résiliation non satisfait> faire
8: Sélectionnez s en N (s)
9: s = échange (s) et s pas dans T
10: si (f(s)<c) alors
11: f= f(s)
12: Choisissez le meilleur coup autorisé
13: Mettre à jour la liste des tabous avec le mouvement choisi
14: Effectuer le mouvement choisi en s
15: Mettre à jour le jeu de configuration non dominante avec s

La ligne1 et la ligne5 présentent la partition initiale s0 obtenue via le processus d'optimisation introduit à savoir MOKP.

Ligne 9 : montre que la méthode d'échange est un mouvement où les positions de deux sommets sont interchangées pour créer un nouveau tour de l'existant et de remplacer une solution actuelle par la meilleure solution de voisinage réalisable (générée par le voisinage de la fonction N (s)).

Line7-8 : Le voisinage d'une solution est l'ensemble de tous les mouvements possibles qui peuvent être faits, la taille du voisinage et le nombre des variables. Ainsi, le voisinage de la solution actuelle est limité aux solutions n'appartenant pas à la liste des tabous. À chaque itération, la meilleure solution parmi l'ensemble autorisé est choisie comme nouvelle solution actuelle. Pour éviter le risque de revenir sur une solution déjà visitée, nous mémorisons les solutions sélectionnées auparavant. De plus, cette solution est ajoutée à la liste taboue et l'une des solutions qui figuraient déjà dans la liste taboue est supprimée (Lignes 13,14). L'algorithme s'arrête lorsqu'une condition de terminaison est remplie. Il peut également s'arrêter si l'ensemble autorisé est vide, c'est-à-dire si toutes les solutions de N (s) sont interdites par la liste taboue. Bien que le stockage des attributs au lieu des solutions complètes soit beaucoup plus efficace, il en résulte une perte d'informations car l'interdiction d'un attribut signifie l'attribution du statut tabou à probablement plus d'une solution. Cela revient à prendre des objets dans le sac à dos et à les retirer du sac. Enfin, si les critères de terminaison sont remplis, renvoyez la solution avec la valeur de fonction de fitness la plus robuste, sinon répétez le cycle d'optimisation en déterminant le prochain déplacement.

Une fois la partition initiale est déterminée, notre approche de classification mixte procèderait par une combinaison successive des opérateurs ascendants et descendants et aboutirait si une solution stable est obtenue en appliquant soit l'agrégation soit l'opérateur de décomposition. Avant la description de la méthode mixte, nous allons mettez l'accent, dans les sections suivantes, sur l'analyse hiérarchique ascendante introduite et celle descendante.

5.3.2 Analyse hiérarchique ascendante

Le principe de l'algorithme ascendant introduit est le même que celui appliqué par les méthodes de classification hiérarchique : A partir d'une partition du niveau r, on recherche, parmi toutes celles de niveau r-1 moins fine, celles qui minimisent le critère d'optimalité. Toutefois, notre approche de classification est basée sur des techniques de la théorie de graphe et des techniques heuristiques. Ainsi, nous définissons notre propre principe et critères d'optimalité.

Principes et optimalité

Nous modélisons le réseau social par un graphe $SN = (V; E;\text{ScoreimportantOp})$. En fait, l'algorithme ascendant est un algorithme itératif qui démarre avec une partition de k groupes. Au début du processus ascendant, chaque membre m du réseau social (sommet) représente une communauté distincte : $C = \{\{m_1\}, \{m_2\}, \{m_3\}, ..., \{m_k\}\}$. Dans

chaque itération, l'algorithme ascendant se déplace d'un niveau de treillis vers le suivant en réalisant une procédure d'agrégation qui consiste à regrouper, dans un même groupe, les membres ayant le plus haut degré de dépendance et similarité. L'algorithme s'arrête quand la procédure d'agrégation n'est plus réalisable et tous les utilisateurs sont regroupés dans la même communauté.

Pour un niveau de treillis donné, l'algorithme Bottom-up examine toutes les paires des groupes constituant la communauté C_k pour trouver les deux groupes c_α et c_β ayant la valeur la plus élevée de $Score_{importantOp}$.

Par conséquent, le pseudo-code de l'algorithme ascendant est donné dans l'algorithme suivant :

Algorithm 5.2 Agglomerative Algorithm *AgA*

Require: graphe pondéré $SN(V, E, Score_{importantOp})$
Initpart[n,n]
Ensure: tous les utilisateurs sont regroupés dans la même communauté
1: $K := n - 1$
2: **repeat**
3: Find $max_{ScoreimportantOp}$(Initpart)
4: $MergingPart(cordMax)$
$k := k - 1$
5: **until** K=2

Par conséquent, la fonction utilisée pour rechercher les sous-partitions avec le plus grand $Score_{importantOp}$ prend, en entrée, une partition initiale et renvoie les sous-partitions les plus similaires et dépendantes.
Cette fonction est décrite comme suit :
$cordMax(Initpart[n : n])$ {
$pe = 0$
tant que $(n \geqslant 2)$
si $(cordMax(m_p) > cordMax(m_q))$ {
$pe \leftarrow cordMax(m_q)$
$cordMax(m_p) \leftarrow cordMax(m_q)$
$cordMax(m_q) \leftarrow pe$
}
retour (cordMax) }
Le processus d'agrégation, à chaque niveau hiérarchique ascendant, est démontré dans l'algorithme suivant :

Algorithm 5.3 Algorithme : MergingP art

Require: Initpart[n,n], cordMax(mp,mq)
Ensure: nouvelle structure agglomérative de communauté

```
1: h=0
2: matx<- Initpart
3: pour(k in 1 :(n-h)){
4: pour (l in k :(n-h)) {
5: si (k == l) {
6: matx[k, l] <- 0
7: }
8: sinon {
9: si((k !=i) || (k !=j)){
10: si (l⩽j) {
11: si ((l==i) || (l==j)){
12: matx [k, l] <- mat [k, i]+mat[k, j]
13: }
14: si ((l !=i) & (l !=j)){
15: matx [k, l] <- mat[k, l]
16: } }
17: sinon {
18: si (h==0){
19: matx [k, l] <- mat[k, l]
20: }
21: si (h>0){
22: matx [k, l] <- mat[k, l + 1]
23: }
24: } }
25: si ((k==i) || (k==j)) {
26: si (l⩽i){
27: matx [k, l] <-mat [l, i] + mat[l, j]
28: }
29: si ((l⩾i)&(l⩽j)){
30: matx[k, l] <-mat[k, l]+mat[l, j]
31: }
32: si (l⩾j){
33: matx[k, l] <-mat[i, l]+mat[j, l]
34: } }
35: } } }
```

Complexité

L'algorithme ascendant n'est évidement pas optimal : nous pouvons seulement être sûr que l'algorithme trouvera les optimums locaux (C_1^*, C_n^*), appelés partitions triviales, ainsi que C_{n-1}^* puisque, dans ces niveaux hiérarchiques (1,n et $n-1$, il aura examiné toutes les partitions. Pour les autres niveaux,$\ell-2$, $\ell-3$,....., 2 nous ne pouvons pas prouver l'optimalité puisque l'algorithme n'aura examiné que les partitions qui lui sont accessibles, c'est-à-dire celle obtenues par le regroupement de deux groupes de la partitions précédente. Pour un niveau k donné, l'algorithme ascendant examine tous les couples de sous-communauté de la partition C_k pour trouver les deux sous partitions, c_a et c_b, les plus dépendantes et similaires ce qui est équivalent à l'examen de toute les partitions du niveau moins fin que C_k pour trouver celle qui minimise le $Score_{importantOp}$. Le passage du niveau ℓ au niveau $\ell-1$ fait l'examen d'un nombre de partitions égal au nombre du couple de sous partitions du niveau précédant, c'est-à-dire :

$$\sum_{\ell=n}^{2} \frac{\ell(\ell-1)}{2} \tag{5.16}$$

Exemple

La Figure 7.3 illustre un échantillon pondéré du graphe social non orienté composé de 6 noeuds et de 15 arêtes. En fait, ce graphe est modélisé sous forme de matrice symétrique car la valeur du $Score_{importantOp}$ (poids de graphe) dépend des deux sommets V_i et V_j.

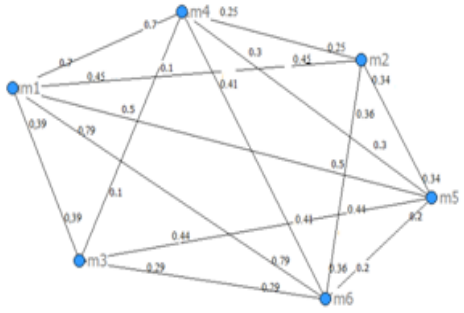


FIGURE 5.3: Echantillon pondéré pour le graphe non orienté

Nous illustrons, dans l'exemple ci-dessous, les différentes étapes de l'algorithme ascendant introduit sur l'échantillon mentionné précédemment (voir figure 7.3). Comme le montre la figure 7.2, la structure de la communauté a été affectée à chaque niveau hiérarchique.

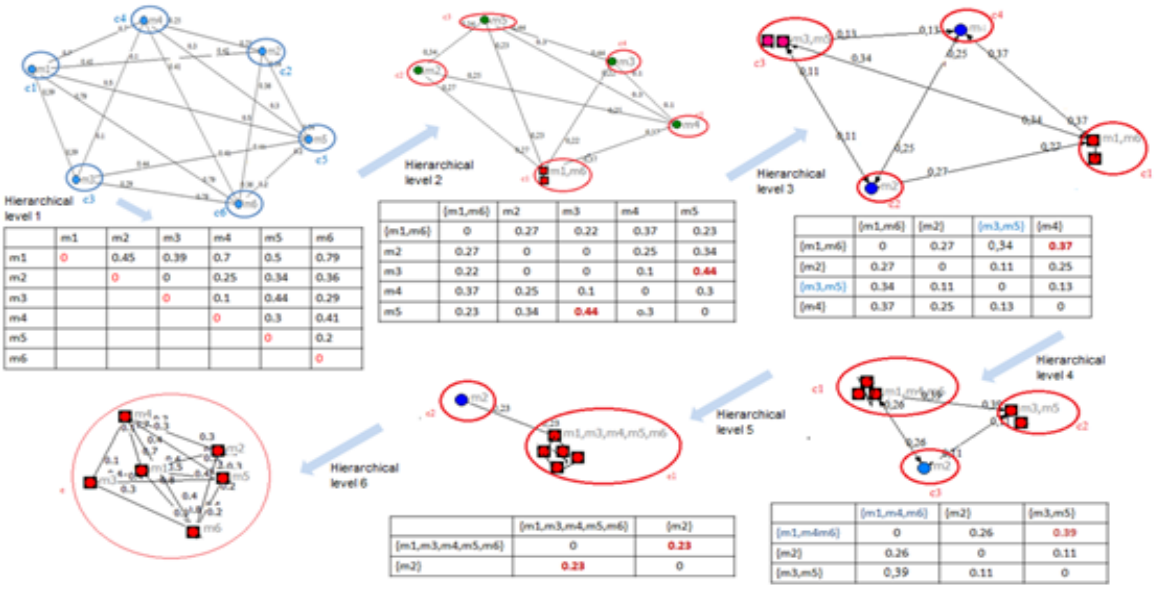


FIGURE 5.4: Visualisation des étapes principales d'algorithme ascendant

Initialement, chaque membre m forme un groupe g séparé :
$C = \underbrace{\{\{m_1\}\}}_{g_1} + \underbrace{\{\{m_2\}\}}_{g_2} + \underbrace{\{\{m_3\}\}}_{g_3} + \underbrace{\{\{m_4\}\}}_{g_4} + \underbrace{\{\{m_5\}\}}_{g_5} + \underbrace{\{\{m_6\}\}}_{g_6}.$
A chaque niveau hiérarchique, nous définissons le Tableau de similarité des dépendances (TSD) associé en fonction de la valeur $Score_{importantOp}$ entre les sous-partitions générées. En se référant au TSD relatif au premier niveau d'agglomération, nous remarquons que les deux groupes sous-détectés, $\{\{m_1\}\}$ and $\{\{m_6\}\}$, ont la valeur la plus élevée de $Score_{importantOp}$. Par conséquent, dans le deuxième niveau hiérarchique ascendant, ces deux groupes ont été fusionnés et la nouvelle structure de la communauté est devenue :
$C = \underbrace{\{\{m_1, m_6\}\}}_{g_1} + \underbrace{\{\{m_2\}\}}_{g_2} + \underbrace{\{\{m_3\}\}}_{g_3} + \underbrace{\{\{m_4\}\}}_{g_4} + \underbrace{\{\{m_5\}\}}_{g_5}.$
Toutefois, à partir du deuxième niveau ascendant, le passage d'un niveau à l'autre est basé sur la moyenne de $Score_{importantOp}$ décrite dans TDS. En fait, obtenons $MoyScore_{importantOp}$ pour chaque groupe sous-détecté comme suit :

$$MoyScore_{importantOp}(g_k) = \frac{\sum_{i=1}^{n} Score_{importantOp}(m_i) \forall m_i \in g_k}{\sum_{i=1}^{n} m_n} \tag{5.17}$$

Par exemple, dans le deuxième niveau hiérarchique, $MoyScore_{importantOp}(g_1, g_2)$ a été obtenu comme suit : $MoyScore_{importantOp}(g_1, g_2) = \frac{Score_{importantOp}(m_1,m_2)+Score_{importantOp}(m_6,m_2)}{3}$
$\frac{0.45+0.36}{3} = 0.27$
Nous appliquons la même stratégie pour tous les prochains niveaux d'agglomération, à savoir niveaux 3, 4 et 5, à l'exception du niveau 6 où l'opérateur d'agrégation n'était pas réalisable et tous les membres étaient réunis dans le même groupe.

5.3.3 Analyse hiérarchique descendante

Malgré que l'algorithme de décomposition introduit soit comparable à celui d'agglomération, il procède par une construction hiérarchique opposée. Dans ce qui suit, nous allons détailler le principe et l'optimalité du processus de division définie en mettant l'accent sur sa complexité et présentant un exemple illustratif.

Principe et optimalité

L'algorithme de division introduit est basé sur le principe de décomposition. Il procède par des séparations successives en deux classes engendrant le moindre degré de dépendance et de similarité. Ces séparations sont comparées aux autres décompositions possibles du même niveau. Il démarre avec une communauté contenant une seule partition. Nous passons d'une partition $g_r = \{\{m_1\}, \{m_2\}, \{m_3\}, ..., \{m_r\}\}$ du niveau r à une partition du niveau r+1 par la division d'une de ses classes en deux . En partant de la partition triviale $C1 = \underbrace{\{\{m_1, m_2, ..., m_n\}\}}_{g_1}$, l'idée est alors de traverser les différents niveaux de treillis en essayant, à chaque étape, de créer les deux sous partitions ayant la moindre valeur de $Score_{importantOp}$. La condition d'arrêt sera atteinte lorsque la procédure d'éclatement n'est plus possible et que chaque membre constitue un groupe séparé. Par conséquent, dans l'algorithme suivant, nous décrivons le pseudo-code de l'algorithme de décomposition proposé.

Algorithm 5.4 Algorithme de division $DivA$

Require: graphe pondéré $SN(V, E, Score_{importantOp})$
Initpart[n,n]
Ensure: chaque membre du réseau social constitue une communauté séparée
1: $K := 1$
2: **repeat**
3: Find $Min_{Scoreimportant Op}$(Initpart)
4: $DecomposePart(cordMin)$
$k := k + 1$
5: **until** K=n-1

La fonction appliquée pour rechercher les sous-partitions avec le plus petit $Score_{importantOp}$ est décrit comme suit :

Algorithm 5.5 Fonction Find min_ $Score_{importantOp}$

Require: partition initiale

Ensure: les membres du réseau social ayant le plus faible $Score_{importantOp}$.

```
1: cord_min <- c(0,0)
2: cord_min(Initpart){
3: f=max(Initpart)
4: i1=j1=0
5: for (k in 1 :n){
6: for (l in k :n){
7: if (k !=l){
8: d<- Initpart [k, l]
9: if (d<f){
10: f<-d
11: i1<-k
12: j1<-l
13: }
14: }
15: }}
16: cord_min <- c(i1,j1)
17: return(cord_min)}
```

L'algorithme suivant montre Le processus de division appliqué à chaque niveau hiérarchique :

Algorithm 5.6 *DecomposePart*

Require: input :Initpart[n,n], $cord_{min}$(mi,mj)
Ensure: output : new divisive community structure

```
1: h = 0
2: matx ← Initpart
3: for k in 1 :(n-h) do
4:    for l in K :(n-h) do
5:       if k = l then
6:          matx[k, l] ← 0
7:       else
8:          if ((k! = i)|(k! = j)|(l <= j)|((l == i)|(l == j)) then
9:             matx[k, l] ← mat[k, i] + mat[k, j]
10:         end if
11:      end if
12:      if ((l! = i)&(l! = j))‖(h = 0) then
13:         matx[k, l] ← mat[k, l]
14:      else
15:         if (h > 0) then
16:            matx[k, l] ← mat[k, l + 1]
17:         end if
18:      end if
19:      if ((k == i)|(k == j))|(l <= i) then
20:         matx[k, l] ← mat[l, i] + mat[l, j]
21:      end if
22:      if ((l >= i)&(l <= j)) then
23:         matx[k, l] ← mat[k, l] + mat[l, j]
24:      end if
25:      if (l >= j) then
26:         matx[k, l] ← mat[i, l] + mat[j, l]
27:      end if
28:   end for
29: end for
```

Le nombre des partitions explorées par l'algorithme de décomposition est plus important que celui des partitions explorées par l'algorithme d'agrégation, puisque, pour trouver le meilleur découpage d'une communauté à p partitions en deux sous communautés, nous devons explorer $2^{p-1} - 2$ possibilités.

Complexité

En fait, l'algorithme de décomposition n'est pas optimal car pour évaluer la complexité de l'algorithme de décomposition introduit, nous calculons premièrement la complexité de chaque niveau hiérarchique. Les seuls optimums locaux trouvés sont C_1^*, C_2^* and C_n^* puisque dans ces niveaux il aura examine toutes les partitions. Pour les autres niveaux (3,4,...,n-1), l'optimalité n'est pas prouvée puisque l'algorithme n'examine que les partitions qui lui sont accessibles, c'est-à-dire celles obtenues par le partitionnement d'un des sous-partitions de la partition précédente en deux. En outre, l'algorithme de décomposition a des bonnes chances de trouver des partitions dont le degré de dépendance et similarité est petit.
Imaginons qu'il existe une communauté optimale C_k^* au niveau k pour laquelle $Score_{importantOp}(C_k^*) = \epsilon$ est petit. Alors, toute partition moins fine que C_k^* aura un degré de dépendance et similarité égal ou supérieur à ϵ. Ces partitions forment des chaines entre C_1^* et C_k^* qui traversent toutes les niveaux entre C_1^* et C_k^*. A partir du niveau 2, il suffit d'examiner uniquement les partitions en deux des sous-communautés créées dans l'étape antérieure. Soit C_1 et C_2 des sous-communautés. Le nombre des alternatives examinées est alors :

$$f(|c_1|, |c_2|) = \frac{2^{|c_1|-2}}{2} + \frac{2^{|c_2|-2}}{2} \tag{5.18}$$

Pour C_1, toutes les partitions de la forme $\{c_1 \backslash m, m\}$ doivent être testées avec , $m \subseteq c_1$ et $m \neq \emptyset$. Au niveau k, il est facile de prouver que$|c_1| + |c_2| < n - (k-2)$ puisqu'il faut que chacune des $k-1$ sous-partitions aient au moins un membre du réseau social. Ainsi, le nombre des partitions examinées pour passer du niveau $k > 2$ au niveau $k+1$ est au maximum $2^{n-k} - 1$. Par ailleurs, pour passer du niveau 1 au niveau 2, il faut tester $\frac{2^{n-2}}{2}$ $= (2^{n-1} - 1)$ partitions. Dans ce cas, la complexité de l'algorithme de décomposition va être inférieure à

$$\sum_{k=1}^{n-1} 2^{n-k} - 1 \tag{5.19}$$

En outre, ce résultat ne constitue, au pire des cas, qu'une borne supérieure de la complexité de l'algorithme de décomposition.

Exemple

Nous utilisons le même échantillon que celui décrit à la figure 7.3 pour illustrer les différentes étapes de notre algorithme de décomposition présentées par la figure 5.5.

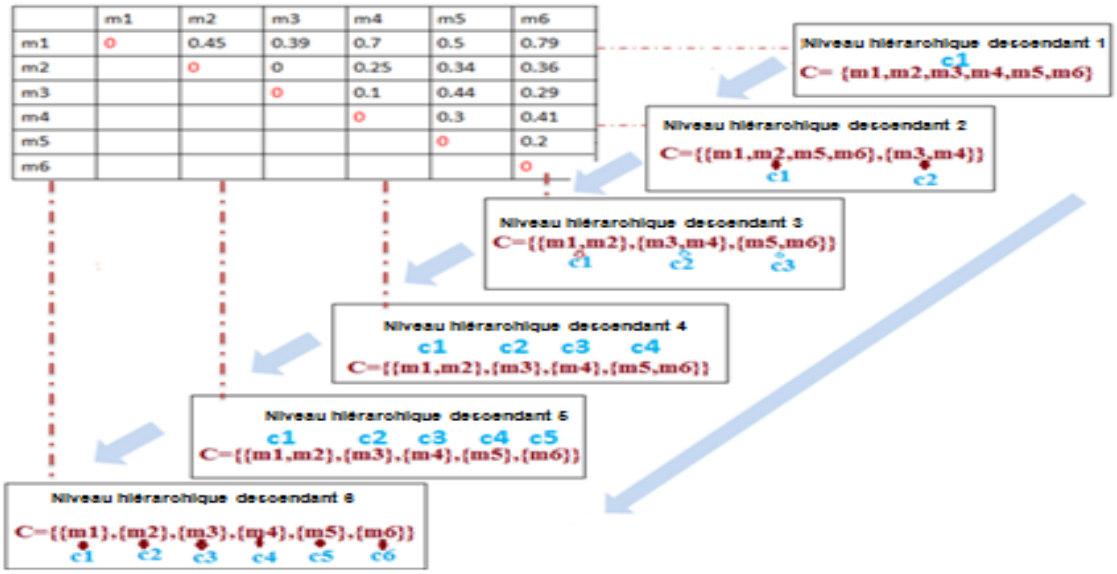


FIGURE 5.5: Visualisation des étapes principales de l'algorithme de décomposition

Comme indiqué dans la figure 5.5, TDS reposant sur la valeur $Score_{importantOp}$ entre les sous-partitions générées est toujours notre point de référence.
Initialement, tous les membres du réseau social constituent une seule communauté. Puis, en se référant à TDS, nous isolons, au niveau hiérarchique descendant 1, les sous-groupes $\{m_3\}$ et$\{m_4\}$ de la communauté parcequ'il ont la moindre valeur $Score_{importantOp}$. De plus, pour passer du niveau hiérarchique descendant 2 à celui 3, nous séparons, de la communauté $C = \{\{m_1, m_2, m_5, m_6\}, ..., \{m_3, m_4\}\}$, les membres $\{m_5\}$ et $\{m_6\}$ ayant la plus petite valeur $Score_{importantOp}$. Ainsi, la communauté obtenue au niveau hiérarchique descendant 3 est composée de trois communautés sous-détectées c_1,c_2 et c_3 ($C = \underbrace{\{m_1, m_2\}}_{c_1}, \underbrace{\{m_3, m_4\}}_{c_2}, \underbrace{\{m_5, m_6\}}_{c_3}$). Nous appliquons la même stratégie pour le reste des niveaux hiérarchiques descendants. A partir du niveau hiérarchique descendant 6, chaque membre constitue une communauté sous-détectée distincte. De ce fait, l'opérateur de décomposition n'était pas réalisable et les critères d'arrêt ont été atteints. Par conséquent, nous pouvons conclure que le nombre des communautés sous-détectées et celui des niveaux hiérarchiques descendants apparentés sont conformes (par exemple : au niveau 3, nous avons 3 groupes sous-détectés ; au niveau 5, nous avons 5 groupes sous-détectés,..., au niveau k, nous avons k groupes sous-détectés).

5.3.4 Analyse hiérarchique mixte

L'algorithme mixte introduit suppose l'existence d'une partition initiale composée de k groupes. En fait, l'algorithme ne change pas le nombre des partitions, mais il varie la répartition fournie par la partition initiale.

Principe et optimalité Notre algorithme procède par une combinaison successive de l'opérateur d'agrégation et celui de décomposition. L'opérateur d'agrégation fournit une partition de $(k-1)$ groupes sur laquelle il suffit d'appliquer l'opérateur de décomposition pour obtenir le nombre initial des groupes k et vice versa.

L'algorithme mixte proposé se termine si les communautés sous-détectées obtenues restent constantes. Ainsi, le processus de stabilisation a été atteint et nous obtenons la même communauté détectée en appliquant soit une méthode ascendante, soit une technique descendante. Le processus hiérarchique mixte recherche, pour chacun des niveaux de treillis, une condition vérifiant la condition d'optimalité (partition stable). La convergence est assurée par le caractère contractant des opérateurs d'agrégation et de décomposition appliqués. Plus concrètement, les boucles principales des algorithmes AgA et $DivA$ deviennent des sous-programmes appelés par l'algorithme mixte. Ainsi, le pseudo-code de la méthode hiérarchique mixte introduite est décrit dans l'algorithme suivant :

Algorithm 5.7 $MHA(Initpart \leftarrow C_k)$

Require: structure communautaire initiale (C_k)
Ensure: communauté hiérarchique mixte (C'_k)
  repeat
    repeat
      C_k =AgA∘DivA(C_k).
    until $(AgA \circ DivA(C_k))$=C_k
    repeat
      C_k =DivA∘AgA(C_k).
    until $(DivA \circ AgA\ (C_k))$= C_k
  until $(AgA \circ DivA(C_k))$=$(DivA \circ AgAC_k))$= C_k

Complexité

L'algorithme mixte fournit, pour chaque niveau hiérarchique soit de décomposition ou d'agglomération, une ou plusieurs partitions stables. Parmi celles-ci, pour les niveaux 1 et n (trivialement) ainsi que pour les niveaux 2 et $n-1$, les optimums globaux seront trouvés étant donne que l'algorithme mixte combine les propriétés des algorithmes de décomposition et celles des algorithmes d'agrégation. Pour les autres niveaux compris strictement entre 3 et $n-2$, les partitions trouvées possèdent une propriété d'optimalité nécessaire.

Ainsi, lors du processus de stabilisation, nous appliquons $AgA \circ DivA$ ou $DivA \circ AgA$ dans un ordre assez irrégulier : : nous appliquons $AgA \circ DivA$ jusqu'à l'obtention d'une partition fixe similaire à celle en appliquant $AgA \circ DivA$ stabilisation des opérateurs.
Enfin, supposons qu'on a j partitions initiales à chaque niveau k et qu'à chacune d'elles on applique j fois les opérateurs de contraction reviennent à $\frac{j}{2}$ applications de $AgA \circ DivA$ et $\frac{j}{2}$ applications de $DivA \circ AgA$. Pour les niveaux $k = 2$ et $k = n - 1$, une seule application de AgA ou $DivA$ suffit pour trouver les partitions optimales du niveau 2 et $n - 1$.
Soit $\Gamma(n, k, \frac{j}{2})$ (respectivement $\Lambda(n, k, \frac{j}{2})$)la plus petite borne supérieure du nombre des partitions examinées par $\frac{j}{2}$ applications de $AgA \circ DivA$ (respectivement $DivA \circ AgA$)au niveau k du treillis des partitions de n sous-communautés. En combinant les complexités du processus d'agrégation et celui de décomposition :
$\Gamma(n, K, \frac{j}{2}) = \frac{j}{2}\left|(2^{n-K} - 1) + \frac{K(K-1)}{2}\right|$.
$\Lambda(n, K, \frac{j}{2}) = \frac{j}{2}\left|(2^{n-K} - 1) + \frac{K(K-1)}{2}\right| = \Gamma(n, K, \frac{j}{2})$.
Enfin, supposons que 'on a j partitions initiales à chaque niveau k et qu'à chacune d'elles on applique j fois les opérateurs de contraction, le nombre total des partitions examinées, autrement la complexité de l'algorithme mixte, est borné supérieurement par :
$\frac{2^n-2}{2} + \frac{n(n-1)}{2} \sum_{k=3}^{n-2} \left|\Gamma(n, k, \frac{j}{2}) + \Lambda(n, k, \frac{j}{2})\right|$.

Exemple Le scénario de modélisation hiérarchique mixte est décrit ci-dessous :

— A chaque niveau hiérarchique, nous élaborons le TDS reposant sur le calcul du $AverageScore_{importantOp}$ entre les communautés sous-détectées.
Soit $C = \underbrace{\{\{m_1, m_6\}\}}_{g_1} + \underbrace{\{\{m_2\}\}}_{g_2} + \underbrace{\{\{m_3\}\}}_{g_3} + \underbrace{\{\{m_4\}\}}_{g_4} + ... \underbrace{\{\{m_5\}\}}_{g_i}$,

$$MoyScore_{importantOp}(g_i, g_j) = \frac{\sum_{i=1}^{n} Score_{importantOp}(m_i, m_j) \forall m_i \in g_i, m_j \in g_j}{\sum_{i=1}^{n} V_i} \tag{5.20}$$

— A chaque niveau hiérarchique ascendant(en anglais Bottom-up hierarchical level (B-U-H-L)), nous fusionnons, les deux groupes ayant la valeur moyenne la plus élevée de $Score_{importantOp}$. Nous nous référons toujours au TDS initial décrivant le réseau afin de déterminer les utilisateurs ayant la valeur de $Score_{importantOp}$ la plus élevée.
— De la même manière, à chaque niveau hiérarchique descendant(en anglais Top-down hierarchical level (T-U-H-L)), nous décomposons, les deux groupes ayant la valeur moyenne de $Score_{importantOp}$ la moins faible.

Nous considérons, au cours de cet exemple, l'échantillon du graphe décrit précédemment dans la figure 7.3. Ainsi, $Score_{importantOp}$ entre les membres du réseau social est décrite dans le tableau initial suivant :

	m1	**m2**	**m3**	**m4**	**m5**	**m6**
m1	0	0.45	0.39	0.7	0.5	0.79
m2	0.45	0	0	0.25	0.34	0.36
m3	0.39	0	0	0.1	0.44	0.29
m4	0.7	0.25	0.1	0	0.3	0.41
m5	0.5	0.34	0.44	0.3	0	0.2
m6	0.79	0.36	0.29	0.41	0.2	0

TABLE 5.3: Tableau initial des valeurs de $Score_{importantOp}$

La valeur $Score_{importantOp}$ dépend des deux sommets V_i et V_j. Par conséquent, l'importance d'un lien permet de qualifier l'indépendance des deux noeuds en fonction de la structure du graphe et des opinions partagées. En outre, à chaque niveau hiérarchique, nous mentionnons la fonction $GScore_{importantOp}$ liéé à la structure de la communauté générée. Cette fonction désigne le $Score_{importantOp}$ entre tous les membres du groupe détecté. L'équation suivante exprime la formule $GScore_{importantOp}$:

$$GScore_{importantOp} = \sum_{i=1}^{n} Score_{importantOp}(m_i) \tag{5.21}$$

Structure communautaire initiale La partition initiale, qui serait ultérieurement injectée comme entrée dans cet exemple, a été obtenue à l'aide de la méthode introduite dans (Toujani.R et Akaichi.J., 2017). Cette dernière partition est composée de trois groupes sous-détectés, à savoir $C = \underbrace{\{\{m_1, m_2\}\}}_{c_1} + \underbrace{\{\{m_3, m_4\}\}}_{c_2} + \underbrace{\{\{m_5, m_6\}\}}_{c_3}$.
Le tableau suivant illustre la valeur de $Score_{importantOp}$ entre ces trois groupes initiaux sous-détectés $c_1 = \{m_1, m_2\}$, $c_2 = \{m_3, m_4\}$ and $c_3 = \{m_5, m_6\}$:

	$\{m_1, m_2\}$	$\{m_3, m_4\}$	$\{m_5, m_6\}$
$\{m_1, m_2\}$	0	0.32	0.49
$\{m_3, m_4\}$	-	0	0.39
$\{m_5, m_6\}$	-	-	0

TABLE 5.4: Valeur de $Score_{importantOp}$ de la structure de communauté initiale

$$GScore_{importantOp}(\{m_1, m_2\}, \{m_3, m_4\}, \{m_4, m_6\}) = 1.2 \tag{5.22}$$

Premier cas : L'algorithme mixte commence par le processus descendant

La méthode mixte commençant par le processus descendant est illustrée par la figure suivante :

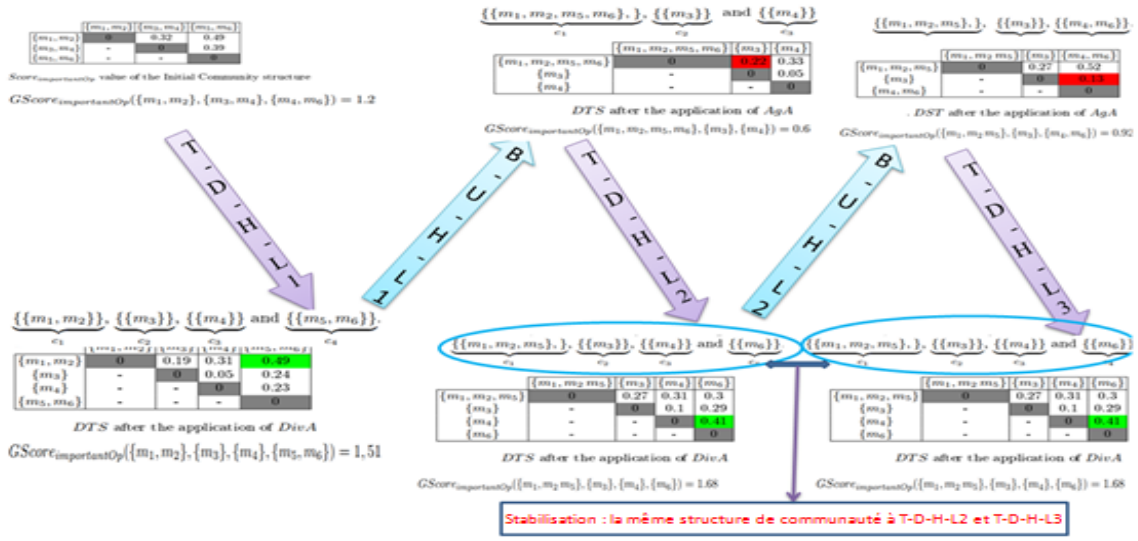


FIGURE 5.6: Visualisation des étapes principales du processus mixte commençant par l'opérateur de décomposition

Dans ce cas, nous commençons par appliquer la méthode *DivA* introduite. Vu que $\underbrace{\{\{m_1, m_2\}\}}_{c_1}$ and $\underbrace{\{\{m_3, m_4\}\}}_{c_4}$ ont la plus petite valeur $Score_{importantOp}$, nous séparons, de ces sous-groupes, m_3 and m_4 ayant le moindre degré d'opinions communs. Ensuite, les partitions générées dans T-D-H-L1 sont injectées en tant qu'entrée dans B-U-H-L1.
A ce dernier niveau, nous appliquons l'opérateur d'agrégation défini dans *AgA* et nous fusionnons $\underbrace{\{\{m_1, m_2\}\}}_{c_1}$ and $\underbrace{\{\{m_5, m_6\}\}}_{c_4}$ parce qu'ils ont la plus haute valeur de $MoyScore_{importantOp}$.
En outre, à T-D-H-L2, m_3 et m_4 sont séparés. Ainsi, à ce niveau hiérarchique, nous isolons du groupe $\{m_1, m_2, m_5, m_6\}$ les membres ayant la valeur $Score_{importantOp}$ minimale avec m_3. Par conséquent, m_3 et m_6 sont décomposés car ils ont la plus petite valeur $Score_{importantOp}$.
Ainsi, la structure de la communauté générée à T-D-H-L2 est $\underbrace{\{\{m_1, m_2, m_5\},\}}_{c_1}$, $\underbrace{\{\{m_3\}\}}_{c_2}$, $\underbrace{\{\{m_4\}\}}_{c_3}$ and $\underbrace{\{\{m_6\}\}}_{c_4}$. De plus, dans B-U-H-L3, m_4 et m_6 sont fusionnés car ils ont la valeur la plus élevée pour $MoyScore_{importantOp}$.
A ce niveau, nous obtenons, en tant que structure de communauté hiérarchique, $\underbrace{\{\{m_1, m_2, m_5\},\}}_{c_1}$, $\underbrace{\{\{m_3\}\}}_{c_2}$, $\underbrace{\{\{m_4, m_6\}\}}_{c_3}$.
Cette dernière partition est injectée comme entrée dans le T-D-H-L3. Dans ce cas, m_4 est

séparé des deux communautés sous-détectées $\underbrace{\{\{m_3\}\}}_{c_2}$ and $\underbrace{\{\{m_4, m_6\}\}}_{c_3}$ ayant la valeur la plus basse du $Score_{importantOp}$. A ce niveau, la structure du communauté détectée reste constante. C'est celle de T-D-H-L2. Ainsi, Le critère d'arrêt de notre méthode mixte est atteint à cette structure de communauté stable$\underbrace{\{\{m_1, m_2, m_5\}, \}}_{c_1}$, $\underbrace{\{\{m_3\}\}}_{c_2}$, $\underbrace{\{\{m_4\}\}}_{c_3}$ and $\underbrace{\{\{m_6\}\}}_{c_4}$.

. **Deuxième cas : Algorithme mixte commençant par le processus ascendant**

La figure suivante illustre les étapes principales de l'algorithme mixte commençant par le processus ascendant.

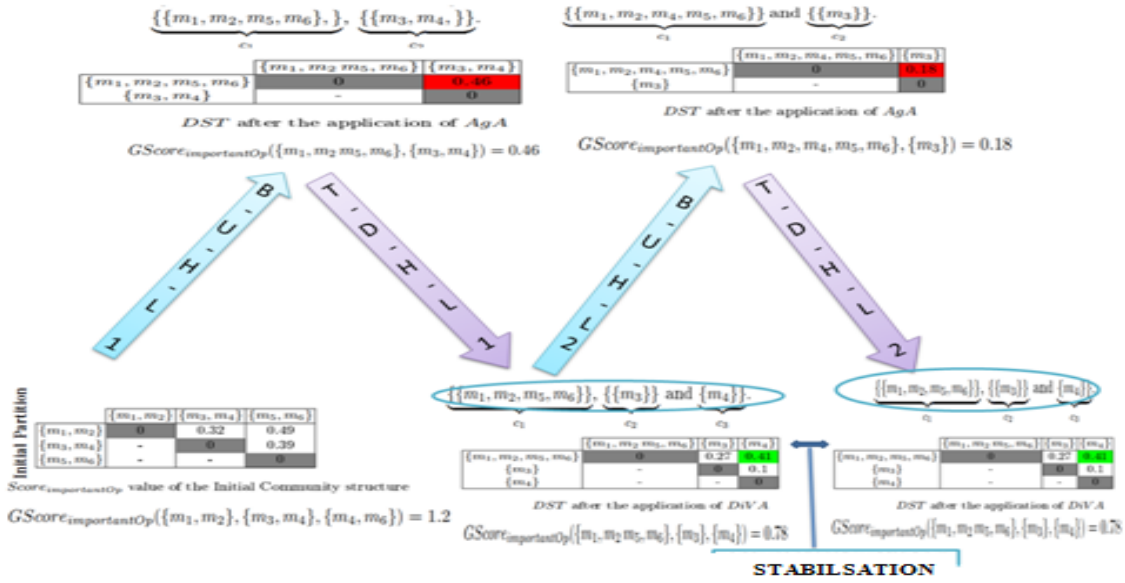


FIGURE 5.7: Description des étapes principales de la méthode mixte commençant par le processus ascendant

Le tableau initial décrivant le degré de similarité et de dépendance TSD entre les membres est toujours le point de départ pour décider quelles partitions seront jointes. Ainsi, les deux groupes sous-détectés $\{m_1, m_2\}$ et $\{m_5, m_6\}$ sont fusionnés car ils avaient la valeur la plus élevée de $Score_{importantOp}$. C=$\{\{m_1, m_2, m_5, m_6\}, \{m_3, m_4\}\}$ est la structure de communauté obtenue à B-U-H-L1. En T-D-H-L 1, nous décomposons, à partir de cette dernière communauté, le membre ayant la plus petite valeur $Score_{importantOp}$ avec les autres membres. Par conséquent, m_3 et m_4 sont séparés et la structure de communauté, à ce niveau hiérarchique, est C=$\{\{m_1, m_2, m_5, m_6\}, \{m_3\}, \{m_4\}\}$. De plus, les deux communautés sous-détectées ($\{\{m_1, m_2, m_5, m_6\}$ et $\{m_4\}$) sont associés à B-U-H-L2 car ils ont le plus haut $MoyScore_{importantOp}$. En conséquence, la structure de la communauté,

à ce niveau hiérarchique, devient C=$\{\{m_1, m_2, m_5, m_6, m_4\}, \{m_3\}\}$. A T-D-H-L2, nous appliquons l'opérateur de décomposition aux communautés détectées à B-U-H-L2. Evidemment, m_3 et m_4, ayant le moins important ScoreimportantOp, forment deux groupes séparés. Ainsi, nous obtenons les mêmes communautés sous-détectées que celle de T-D-H-L1. Le processus de stabilisation est réalisé en obtenant la même partition fixe. Généralement, en commençant par DivA ou AgA, la méthode mixte introduite fournit la même structure communautaire fixe. En fait, comme indiqué dans la figure 5.8, l'évolution de$GScore_{importantOp}$ montre que la structure de la communauté, obtenue par notre méthode combinée, représente un optimum local de la fonction $GScore_{importantOp}$.

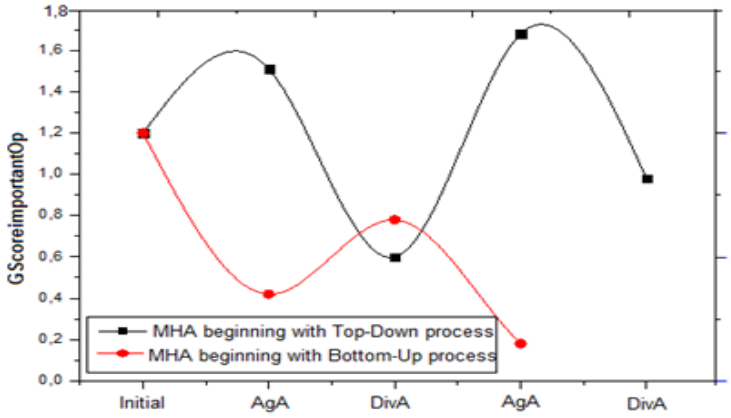


FIGURE 5.8: Recherche de la structure de la communauté locale

En fait, la figure 5.8 illustre la recherche de l'optimum local et l'évolution du $GScore_{importantOp}$ pour les applications successives de $DivA$ et de AgA.

5.4 Conclusion

Dans ce chapitre, nous avons présenté notre méthode de classification hiérarchique hybride pour la détection de communauté dans les réseaux sociaux.
Nous avons présenté trois algorithmes pour la classification hiérarchique des communautés : l'algorithme d'agrégation, l'algorithme de décomposition et l'algorithme mixte. Ce dernier, basé sur la combinaison des deux autres algorithmes, suppose l'existence d'une partition initiale de k classes. Il a pour but de détecter de nouvelle structure hiérarchiquue de la communauté et non pas la modification du nombre de ses partitions.
L'algorithme hybride procède par des combinaisons successives de l'opérateur d'agrégation et celui de décomposition. L'opérateur d'agrégation a fourni une partition de $(k-1)$ classes sur lesquelles il est suffisant d'appliquer l'opérateur de décomposition pour obtenir

de nouveau le nombre initial de k partitions. L'algorithme peut commencer par l'opérateur d'agrégation ou celui de décomposition. Dans ce cas, la structure communautaire obtenue ne sera pas obligatoirement la même. En fait, l'algorithme s'arrête à l'obtention d'une structure fixe. Ceci représente l'optimum local de la fonction $GScore_{importantOp}$. Par conséquent, pour éviter la convergence vers une structure communautaire locale, nous avons intégré les métaheuristiques dans notre méthode de classification. La démarche proposée pour l'utilisation des métaheuristiques dans le processus de classification sera étudiée dans le chapitre suivant.

Chapitre 6

Intégration des métaheuristiques dans la classification hiérarchique mixte

6.1 Introduction

L'organisation hiérarchique des communautés, réunies pour former l'ensemble du réseau social, montre certaines des propriétés basiques de sa structure hiérarchique. Ainsi, la classification hiérarchique pour la découverte des communautés est une tâche cruciale dans la détection des communautés dans les réseaux sociaux. En fait, la plupart des algorithmes existants proposés pour la détection de communauté hiérarchique sont basés sur un processus d'agglomération ou celui de division. Pour le meilleur de notre connaissance, il n'y a pas d'approche qui combine ces deux processus. Dans le chapitre précédent, nous avons développé une nouvelle méthode hybride de classification hiérarchique qui a posé le problème de la convergence vers une communauté détectée localement optimale.
Dans la section 6.2, nous montrons l'intégration des métaheuristiques dans la classification hiérarchique mixte introduite pour éviter le piège d'optimum local. Nous introduisons, dans la section 6.3, une modélisation génétique pour la classification hiérarchique mixte des communautés(Toujani and Akaichi, 2018, oujani and Akaichi, 2018). Dans le but d'approfondir notre analyse hiérarchique, la section 6.4 met l'accent sur l'intelligence en essaim pour générer une structure communautaire hiérarchique globalement optimale(Toujani and Akaichi, 2019a, oujani and Akaichi, 2019a).

6.2 Modélisation de l'optimisation des communautés hiérarchiques

Formalisation du problème Comme nous avons déjà présenté dans le chapitre précédent qui décrit la classification hiérarchique mixte, nous modélisons les réseaux sociaux sous la forme d'un graphe pondéré $G = (V; E; \alpha)$ où V représente les utilisateurs dans les réseaux de médias, E dénote les différentes interactions entre eux et α est la valeur du $Score_{importantOp}$ entre les membres. Etant donné un partitionnement initial, $P = \{p_1, p_2, p_k\}$ de $G = (V; E; \alpha)$ et Q une fonction objectif indiquant la qualité de ce partitionnement, l'intégration des métaheuristiques dans la détection hiérarchique de communauté consiste à trouver la structure communautaire globale optimale. Ainsi, il faut définir la fonction à optimiser pour assurer un bon partitionnement.

Fonction objectif :

La nouvelle fonction de modularité basée sur le $Score_{importantOp}$: Nous améliorons la modularité de Newman[] en ajoutant le degré de dépendance et celui de similarité entre les membres des réseaux sociaux mesuré par $Score_{importantOp}$ défini dans le chapitre précédent.aa La nouvelle fonction de modularité exige qu'une bonne partition soit une partition pour laquelle l'importance des sommets dans un cluster i spécifique est plus élevée que le degré d'importance entre tous les groupes de sommets. Par conséquent, nous définissons la fonction de modularité reposant sur la dépendance et la similarité (Q_{DS}) comme indiqué ci-dessous :

$$Q_{DS} = \sum_{i=1}^{NP} \left(\frac{IOpLD_i}{TOpLD} - \left(\frac{DOpLD_i}{TOpLD} \right)^2 \right) \tag{6.1}$$

où NP est le nombre de partitions, $IScore_{importantOp}$ représente le total de $Score_{importantOp}$ des sommets au sein de partition i,$DScore_{importantOp}$ désigne le total de $Score_{importantOp}$ entre les sommets de la partition et tout sommet du graphe, $TScore_{importantOp}$ correspond au total de $Score_{importantOp}$ entre n'importe quels deux sommets du graphe, $Score_{importantOp}$ représente le degré d'importance entre un sommet dans un $cluster_i$ spécifique, V désigne l'ensemble des sommets du graphe et V_i est l'ensemble des sommets du $cluster_i$. Par conséquent, nous attribuons une fonction objective Q_{DS} pour chaque opérateur hiérarchique :

— ***Fonction objective du processus ascendant***
L'équation qui décrit la fonction de qualité au niveau hiérarchique agglomératif se présente comme suit :

$$AgQ_{DS} = \max \sum_{i=1}^{NP} Q_{DS} \tag{6.2}$$

— ***Fonction objective du processus descendant***
L'équation suivante définit la fonction de fitness au niveau hiérarchique de division :

$$DivQ_{DS} = \min \sum_{i=1}^{NP} Q_{DS} \tag{6.3}$$

— ***Fonction objective du processus hiérarchique mixte***
En conséquence, la fonction objective du processus mixte combine AgQ_{DS} et $DivQ_{DS}$. Cette fonction de qualité est illustrée dans l'équation suivante :

$$MixQ_{DS} = AgQ_{DS} \circ DivQ_{DS} \tag{6.4}$$

La nouvelle version de classification hiérarchique mixte applique, comme métaheuristiques, les algorithmes génétiques et l'intelligence en essaim pour optimiser la fonction de qualité Q_{DS} et éviter la convergence vers une structure de communauté hiérarchique localement optimale. Dans les sections suivantes, nous mettons en relief l'intégration de ses métaheuristiques dans notre méthode de classification.

6.3 Analyse génétique et classification hybride des communautés

L'extension de la méthode hiérarchique mixte (Toujani and Akaichi, 2018, oujani and Akaichi, 2018) vise a introduire une méthode de détection de communauté hiérarchique hybride génétique. En effet, nous proposons une formulation hybride génétique de classification hiérarchique pour la détection des communautés dans les réseaux sociaux. Dans cette classification, les techniques hiérarchiques ascendantes et descendantes sont combinées afin de produire la même structure de communauté g2n2tiaue. Par conséquent, la méthode développée suppose l'existence d'une structure de communauté initiale composée, à l'origine, de n partitions (groupes) (Toujani and Akaichi, 2017, oujani and Akaichi, 2017). Elle ne change pas le nombre des partitions, mais elle modifie la distribution initiale en optimisant la modularité optimale Q_{DS} dans un graphe. Généralement, toutes les approches heuristiques souffrent de l'inconvénient d'optimum local. Pour évaluer la fonction de modularité proposée, nous utilisons une approche heuristique génétique spécifique en évitant les pièges d'optimaux locaux. En fait, l'approche génétique de détection de communauté hiérarchique hybride est une heuristique qui combine les différentes méthodes hiérarchiques afin de trouver des clusters dans des réseaux complexes.
La première technique est la méthode hiérarchique génétique ascendante (en anglais Genetic Bottom-up method GB - up) basée sur l'opérateur d'agrégation. A chaque itération, (GB-up) assemble les deux communautés ayant la plus hausse valeur de Q_{DS}. Cependant, la deuxième méthode est une technique génétique hiérarchique descendante (en anglais Genetic top-down method (GT-down)) basée sur la suppression des arêtes ayant la moindre valeur de Q_{DS}. Enfin, la troisième méthode est l'algorithme génétique hybride qui combine $GB-up$ et $GT-down$. Le partitionnement génétique introduit est résumé par les étapes suivantes :

— **Générer la population initiale**

Le choix de la population initiale affecte les performances du partitionnement génétique. Elle peut être générée aléatoirement ou à l'aide de stratégies spécifiques. Comme déjà mentionné, la structure de communauté initiale de la partition génétique introduite est obtenue en utilisant la méthode décrite dans (Toujani and Akaichi, 2017, oujani and Akaichi, 2017).

— **Evaluation** A cette étape, nous mettons l'accent sur la fonction objective à optimiser (Q_{DS}) à chaque niveau hiérarchique.

— **Sélection** Le but de la première étape de sélection est d'associer, à chaque partition, sa fonction de qualité. Ainsi, chaque groupe dans la partition initiale reçoit une probabilité de reproduction qui dépend, d'une part, de sa propre valeur de la fonction objective et, d'autre part, des autres fonctions objectives de tous les autres groupes du graphe. En fait, nous associons, à chaque groupe C_i, une probabilité Pr_i proportionnelle à sa fonction objective Q_{DS} (C_i).
Soit q le nombre des groupes dont la probabilité de sélection (PrS) est calculée comme suit :

$$Pr_i = \frac{Q_{OpL}(C_i)}{\sum_{i=1}^{q} Q_{OpL}(C_i)} \tag{6.5}$$

— **Reproduction**
A ce stade, nous appliquons des opérateurs génétiques (croisement et mutation) à toutes les paires de groupes sélectionnées.
Supposons qu'on a un graphe G composé de n membres (m) placés sur k groupes. Chacun des groupes est étiqueté par un symbole entre 1 et k. Un placement donné de m est représenté par un vecteur de taille n de ces symboles. Comme montré dans le tableau 6.1, pour chaque sommet i, nous associons l'élément i du vecteur indiquant le groupe auquel il appartient et l'élément j représentant la valeur de Q_{DS}.

Cluster	1	2	3	3	2	4
Membre	1	2	3	4	5	6
Q_{DS}	0.56	0.75	0.45	0.27	0.89	0.5

TABLE 6.1: L'étape de placement

En outre, l'opérateur de mutation introduit est utilisé pour changer le placement des membres sur d'autres clusters ainsi que pour modifier l'emplacement des noeuds dans les clusters. Cependant, l'opérateur de croisement proposé est employé pour combiner deux placements donnés. Généralement, les deux opérateurs sont appliqués respectivement à chaque niveau hiérarchique de division et d'agglomération.

6.3.1 Analyse génétique ascendante

L'algorithme génétique Bottom-Up (GB-up) est un algorithme itératif reposant sur l'opérateur d'agrégation. En fait, à chaque itération, GB-up fusionne les deux communautés ayant la sélection probabiliste la plus élevée. Ainsi, un individu avec une fonction fitness plus élevée aura plus de chance d'être sélectionné qu'un autre avec une valeur de fitness inférieure. Les étapes importantes de cet algorithme génétique itératif sont résumées au dessous :

1. Initialement, en tant que population initiale, $GB-up$ considère chaque membre du réseau social m_i comme une partition p_i. Ainsi, la structure de la communauté C_s est présentée comme suit : $C_s = \underbrace{\{\{m_1\}\}}_{p_1}, \underbrace{\{\{m_2\}\}}_{p_2}, ..., \underbrace{\{\{m_n\}\}}_{p_i}$.
2. Calculer PrS de chaque partition. En fait, l'algorithme ascendant introduit passe d'un niveau k à un autre en effectuant une procédure d'agrégation qui consiste à mettre en cluster deux utilisateurs ayant la plus hausse PrS.
3. Sélectionnez un utilisateur du réseau social selon sa fonction objective et la sélection probabiliste.
4. A chaque niveau hiérarchique k, l'algorithme passe d'un niveau hiérarchique à un autre en effectuant un opérateur de mutation, en modifiant l'emplacement des utilisateurs des réseaux sociaux et en fusionnant chaque paire possible de partitions p_g et p_h ($g \neq h$).
5. Remplacez les utilisateurs avec une meilleure fonction de qualité et mettez à jour Q_{DS}.
6. L'algorithme se termine si tous les utilisateurs sont regroupés dans une partition et que la procédure de mutation n'est plus réalisable.

6.3.2 Analyse génétique descendante

L'algorithme génétique Top Down ($GT-down$) est similaire à ($GB-up$). Cependant, il procède par une construction hiérarchique opposée. Les étapes principales de l'algorithme $GT-down$ proposé sont mentionnées au dessous :

1. La méthode de partitionnement génétique descendante considère tous les utilisateurs de réseaux sociaux comme population initiale formant une seule communauté. $C_s = \underbrace{\{\{m_1, m_2, .., m_n\}\}}_{p_1}$
2. L'algorithme $GT-down$ introduit passe d'un niveau k à l'autre en effectuant une procédure de décomposition qui consiste à séparer deux utilisateurs ayant la plus faible PrS.
3. Sélectionnez un utilisateur du réseau social selon sa fonction objective et de la sélection probabiliste.

4. Appliquer l'opérateur génétique de croisement : fusionner toutes les paires possibles des partitions de cluster.
5. Remplacer les utilisateurs ayant la moindre de PrS et mettre à jour Q_{DS}.
6. L'algorithme s'arrête si la procédure d'éclatement n'est plus réalisable et chaque utilisateur du réseau social constitue une partition.

6.3.3 Analyse génétique mixte

L'algorithme génétique hybride, exploitant alternativement les deux algorithmes mentionnés précédemment, peut être résumé en ces étapes :

1. L'algorithme génétique hybride nécessite l'existence d'une structure communautaire initiale de n partitions.
2. Il procède par la combinaison successive des algorithmes $GB-up$ et $GT-down$. Ce dernier fournit une décomposition et un partitionnement de $(n-1)$ groupes sur lesquels il suffit d'appliquer l'opérateur d'agrégation utilisé dans $GB-up$ pour obtenir le nombre initial des groupes (n) et inversement.
3. L'algorithme s'arrête si une partition stable est obtenue.

Exemple Dans cet exemple illustratif, nous nous concentrons sur les étapes d'exécution de l'approche de partitionnement génétique hybride introduit. **Définir le codage du problème :** Nous considérons le graphe suivant modélisant le réseau social composé de 6 noeuds et 18 arrêtes (figure6.1).

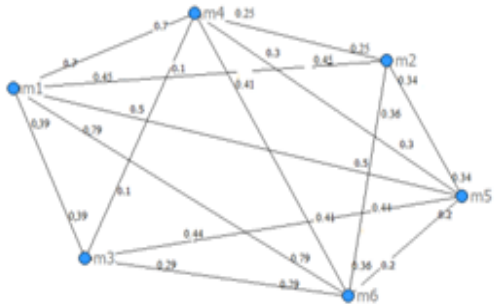


FIGURE 6.1: Echantillon pondéré du graphe non orienté

Les valeurs des poids indiquent la valeur $Score_{importantOp}$ entre les noeuds. Par conséquent, nous modélisons le réseau sous forme de matrice symétrique car la valeur de $Score_{importantOp}$ dépend à la fois des membres du réseau social m_i et m_j. Le tableau 6.2 illustre $Score_{importantOp}$ montrant le degré de leader d'opinion (OpLD)entre les membres du réseau social. shared opinion between social network members'.

	m1	**m2**	**m3**	**m4**	**m5**	**m6**
m1	0	0,45	0,39	0,7	0,5	0,79
m2	0,45	0	0	0,25	0,34	0,36
m3	0,39	0	0	0,1	0,44	0,29
m4	0,7	0,25	0,1	0	0,3	0,41
m5	0,5	0,34	0,44	0,3	0	0,2
m6	0,79	0,36	0,29	0,41	0,2	0

TABLE 6.2: Tableau initial décrivant $Score_{importantOp}$ entre les membres

Génération de la population initiale La méthode génétique introduite commence par la génération de la population initiale. En fait, en se référant à la méthode décrite dans (Toujani and Akaichi, 2017, oujani and Akaichi, 2017), cette dernière partition initiale est composée de trois groupes sous-détectés, à savoir $C = \underbrace{\{\{m_1, m_2\}\}}_{c_1} + \underbrace{\{\{m_3, m_4\}\}}_{c_2} + \underbrace{\{\{m_5, m_6\}\}}_{c_3}$.
Après avoir accordé à chaque partition sa fonction de qualité, nous appliquons, à chaque niveau hiérarchique, les étapes suivantes :

- Nous appliquons la sélection probabiliste (PrS) pour sélectionner les groupes qui seront fusionnés ou décomposés.
- En se référant à l'opérateur de mutation ou de croisement, nous choisissons certains membres de clusters qui seront fusionnés ou décomposés.

Dans cet exemple, la classification hiérarchique hybride introduite commence par l'utilisation de l'opérateur de division.
Niveau hiérarchique descendant 1
Nous élaborons le tableau 6.3 décrivant la PrS de la structure de communauté initiale composée de : $c_1 = \{m_1, m_2\}$, $c_2 = \{m_3, m_4\}$ and $c_3 = \{m_5, m_6\}$:

	$\{m_1, m_2\}$	$\{m_3, m_4\}$	$\{m_5, m_6\}$
$\{m_1, m_2\}$	0	0,32	0,49
$\{m_3, m_4\}$	-	0	0,39
$\{m_5, m_6\}$	-	-	0

TABLE 6.3: Probabilité de sélection des groupes initiaux détectés

Nous remarquons que $\underbrace{\{\{m_1, m_2\}\}}_{c_1}$ and $\underbrace{\{\{m_3, m_4\}\}}_{c_4}$ ont la plus petite valeur de PrS.
En conséquence, nous séparons, de ces sous-partitions, les membres ayant la moindre Q_{DS} détectées dans l'étape de placement résumée dans le tableau 6.4.

cluster	$\{m_1, m_2\}$	$\{m_1, m_2\}$	$\{m_3, m_4\}$	$\{m_3, m_4\}$	$\{m_5, m_6\}$	$\{m_5, m_6\}$
members (node)	m_1	m_2	m_3	m_4	m_5	m_6
OpLD	0,45	0,45	0,1	0,1	0,2	0,2

TABLE 6.4: Etape de placement du niveau hiérarchique descendant 1

En se référant à l'étape de placement, les membres m_3 et m_4 du réseau social ont la moins Q_{DS}. Nous reproduisons ces derniers membres en appliquant un opérateur de croisement et en modifiant leur emplacement. Dans ce cas, nous décomposons $\{m_3, m_4\}$ et chacun d'entre eux composera ultérieurement une partition séparée. Par conséquent, la nouvelle structure de communauté reproduite devient : $CS_1 = \underbrace{\{\{m_1, m_2\}\}}_{c_1}, \underbrace{\{\{m_3\}\}}_{c_2}, \underbrace{\{\{m_4\}\}}_{c_3}$ and $\underbrace{\{\{m_5, m_6\}\}}_{c_4}$.

Niveau hiérarchique ascendant 1

La structure de communauté générée au niveau hiérarchique descendant précédent constitue l'entrée de ce niveau hiérarchique ascendant. Le tableau 6.5 illustre les PrS qui convient :

	$\{m_1, m_2\}$	$\{m_3\}$	$\{m_4\}$	$\{m_5, m_6\}$
$\{m_1, m_2\}$	0	0,19	0,31	0,49
$\{m_3\}$	-	0	0,05	0,24
$\{m_4\}$	-	-	0	0,23
$\{m_5, m_6\}$	-	-	-	0

TABLE 6.5: Probabilité de sélection de CS_1

A ce niveau hiérarchique, on applique l'opérateur d'agrégation défini dans $GB - up$. En se référant au tableau 6.5, $\underbrace{\{\{m_1, m_2\}\}}_{c_1}$ and $\underbrace{\{\{m_5, m_6\}\}}_{c_4}$ ont la plus haute valeur de PrS. Ainsi, nous appliquons l'opérateur de croisement introduit et nous combinons ces deux partitions sous-détectées. En conséquence, la structure de la communauté générée, à ce niveau hiérarchique devient :

$CS_2 = \underbrace{\{\{m_1, m_2, m_5, m_6\}, \}}_{c_1}, \underbrace{\{\{m_3\}\}}_{c_2}$ and $\underbrace{\{\{m_4\}\}}_{c_3}$

Niveau hiérarchique descendant 2 La probabilité de sélection de CS_2 est définie dans le tableau 6.6 :

	$\{m_1, m_2, m_5, m_6\}$	$\{m_3\}$	$\{m_4\}$
$\{m_1, m_2, m_5, m_6\}$	0	0,22	0,33
$\{m_3\}$	-	0	0,05
$\{m_4\}$	-	-	0

TABLE 6.6: Probabilité de sélection de CS_2

A ce niveau hiérarchique descendant, nous isolons les groupes ayant la moindre sélection probabiliste. De toute évidence, les membres m_3 et m_4 sont déjà séparés. Nous isolons aussi, du groupe $\{m_1, m_2, m_5, m_6\}$, les membres du réseau social ayant la plus faible valeur Q_{DS} avec m_3. Par conséquent, le tableau 6.7 décrit l'étape de placement associé :

cluster	$\{m_3\}$	$\{m_3\}$	$\{m_3\}$	$\{m_3\}$
members (node)	m_1	m_2	m_5	m_6
$OpLD$	0,39	0	0,44	0,29

TABLE 6.7: Etape de placement au niveau hiérarchique descendant 2

L'étape de placement décrite dans le tableau 6.7 montre que les membres du réseau m_3 and m_6 ont la plus petite valeur Q_{DS}. Par conséquent, la structure de la communauté générée à ce niveau hiérarchique devient :
$CS_3 = \underbrace{\{\{m_1, m_2, m_5\},\}}_{c_1}, \underbrace{\{\{m_3\}\}}_{c_2}, \underbrace{\{\{m_4\}\}}_{c_3}$ and $\underbrace{\{\{m_6\}\}}_{c_4}$.

Niveau hiérarchique ascendant 2 La probabilité de sélection liée à CS_3 est présentée dans le tableau 6.8 :

	$\{m_1, m_2\, m_5\}$	$\{m_3\}$	$\{m_4\}$	$\{m_6\}$
$\{m_1, m_2, m_5\}$	0	0,27	0,31	0,3
$\{m_3\}$	-	0	0,1	0,29
$\{m_4\}$	-	-	0	0,41
$\{m_6\}$	-	-	-	0

TABLE 6.8: probabilité de sélection de Cs_3

De toute évidence, nous modifions l'emplacement de m_4 et m_6 et nous les avons rejoints car ils ont la valeur PrS la plus élevée. Ainsi, la structure de la communauté devient :
$Cs_4 = \underbrace{\{\{m_1, m_2, m_5\},\}}_{c_1}, \underbrace{\{\{m_3\}\}}_{c_2}, \underbrace{\{\{m_4, m_6\}\}}_{c_3}$.

Niveau hiérarchique descendant 3 La sélection probabiliste de CS_4 est définie dans le tableau 6.9 :

	$\{m_1, m_2\, m_5\}$	$\{m_3\}$	$\{m_4, m_6\}$
$\{m_1, m_2, m_5\}$	0	0,27	0,52
$\{m_3\}$	-	0	0,13
$\{m_4, m_6\}$	-	-	0

TABLE 6.9: Probabilité de sélection de CS_4

Comme indiqué dans le tableau 6.9, les deux partitions $\{m_4, m_6\}$ et $\{m_3\}$ ont la moindre valeur de PrS. Ainsi, nous isolons le membre générant la plus basse Q_{DS}. Pour cela, nous présentons, dans le tableau 6.10, le placement à ce niveau hiérarchique descendant.

cluster	$\{m_3\}$	$\{m_3\}$
members (node)	m_4	m_6
$OpLD$	0,1	0,29

TABLE 6.10: Etape de placement dans le niveau hiérarchique descendant 3

Il est clair que m_4 est décomposé et nous obtenons la même partition sous-détectée que celle du niveau hiérarchique descendant 2. Cette structure communautaire stable est : $\underbrace{\{\{m_1, m_2, m_5\}, \}}_{c_1}, \underbrace{\{\{m_3\}\}}_{c_2}, \underbrace{\{\{m_4\}\}}_{c_3}$ and $\underbrace{\{\{m_6\}\}}_{c_4}$.
Tant que la structure de la communauté générée reste constante et le partitionnement génétique hybride introduit produit la même structure de communauté, le critère d'arrêt est atteint.
La méthode génétique développée n'assure pas l'optimisation approfondie à chaque niveau hiérarchique, d'où la nécessité d'intégrer les techniques d'intelligence en essaim dans notre processus de classification. Ceci constitue le but de la section suivante.

6.4 Intelligence en Essaim et classification hybride des communautés

La nouvelle version de classification hiérarchique mixte intègre l'optimisation et utilise les techniques d'intelligence en essaim pour optimiser Q_{DS} et éviter la convergence vers une structure de communauté hiérarchique localement optimale. Nous choisissons d'utiliser l'intelligence en essaim parce que ses caractéristiques suivantes évitent la convergence vers un optimum local :

- Les essaims ont différents comportements en parallèle,
- Les interactions des individus entre eux et l'environnement renforcent les actions des individus,

— L'augmentation de la taille du problème est résolue en ajoutant plus d'individus.

6.4.1 Correspondance entre le comportement des essaims et celui des membres du réseau social

Généralement, l'analyse des comportements de communauté dans les réseaux sociaux et l'analyse des comportements d'essaims sont similaires. En établissant les similarités entre les deux problèmes, les sources de nourriture ayant la plus grande quantité de nectar et le bon passage de colonie à une source d'aliment ressemblent à la détection d'un groupe de membres ayant la fonction qualité (modularité (Q_{DS}) la plus haute. Nous assimilons la recherche des leaders parmi les membres de réseaux sociaux, d'une part, à la recherche de nectar par des abeilles et, d'autre part, aux premières fourmis qui cherchent de la nourriture. En outre, lorsque les premières fourmis retournent dans leur colonie et s'étendent sur une piste de phéromone, les fourmis de construction suivent cette dernière piste. Ainsi, les fourmis de construction et les abeilles sont similaires aux membres qui sonts influencés par les leaders d'opinion.
Les abeilles et les fourmis scoutes sont assimilées à des membres influencés par ceux qui dirigent la recherche, mais qui en cherchent un autre. Par conséquent, l'intelligence des essaims et l'analyse des réseaux sociaux ont pour objectif, respectivement, de rechercher la source d'alimentation la plus rentable et de déterminer la communauté présentant la valeur de modularité la plus élevée. Les analogies entre ces deux problèmes sont résumées dans le tableau suivant :

Colonie d'abeilles	**Colonie de fourmis**	**Utilisateurs des réseaux sociaux**
sources de nourriture	énergie gagnée (nourriture)	degré d'opinions partagées
La fonction objective indique la quantité de nectar	La fonction objective indique les traces de phéromones montrant le bon chemin	La fonction objective représente la Modularité Q_{DS} définie
Des abeilles	Fourmis	Membres de réseaux sociaux
— Abeilles butineuses — Abeilles employées — Abeilles scoutes	— Premières fourmis — Fourmis employés — Fourmis Scoutes	— Les membres tentent d'être leaders d'opinion — Les membres suivent les leaders d'opinion — Les membres suivent les leaders d'opinion mais recherchent un autre.
objectif : maximiser le nectar recueilli à partir de diverses fleurs	**objectif** : maximiser le compromis entre l'énergie gagnée (nourriture) et l'énergie dépensée pour emmener la proie au nid	**objectif** : détecter les k communautés qui optimisent la fonction de qualité Q_{DS}

TABLE 6.11: Comportements des utilisateurs des réseaux sociaux assimilés aux comportements des Swarms (colonie d'abeilles et de fourmis)

L'intégration des essaims dans notre classification mixte repose sur l'utilisation de la colonie des fourmis afin d'améliorer la $DivA$ introduite et définir ($AntCDivA$) ainsi que la colonie d'abeilles pour faire progresser l'AgA en proposant($BeeCAgA$). Comme le cas de la classification mixte de base, celle reposant sur l'application alternative d'$AntCDivA$ et de $BeeCAgA$ suppose l'existence d'une structure de communauté initiale. En fait, le comportement collectif et social des abeilles semble plus approprié au processus agglomérant suggéré. Comme la méthode d'optimisation des abeilles est employée pour identifier les aliments ayant un niveau de qualité élevé, nous cherchons, lors du processus d'agglomération proposé, à détecter les utilisateurs ayant un score $Score_{importantOp}$ élevé. Néanmoins, le principe des fourmis pour découvrir le chemin le plus court allant de leur nid à la

source de nourriture est proche du celui du processus de division introduit reposant sur la décomposition des utilisateurs des réseaux sociaux ayant un faible $Score_{importantOp}$.

6.4.2 Analyse mixte basée sur l'Intelligence en Essaim

Le nouvel algorithme hiérarchique mixte basée sur les essaims (en anglais Mixed Hierarchical Algorithm based Swarm(MHAS)) exploite alternativement les deux algorithmes $AntCDivA$ et de $BeeCAgA$. Nous appliquons $AntCDivA$o$BeeCAgA$ ou $BeeCAgA$o$AntCDivA$ dans un ordre régulier jusqu'à la stabilisation de la structure communautaire. En fait, l'algorithme s'arrête à l'obtention des même partitions en appliquant $BeeCAgA$ ou $AntCDivA$. Les principales étapes de MHAS sont résumées ci-dessous :

- MHAS nécessite l'existence d'une partition initiale (Toujani.R et Akaichi.J., 2017).
- Il procède par une combinaison successive des $AntCDivA$ et $BeeCAgA$ introduits.
- L'algorithme s'arrête si la communauté détectée obtenue en appliquant $AntCDivA$ est identique à celle obtenue par $BeeCAgA$.

Soit Cc_k la structure de communauté obtenue au niveau hiérarchique k. Le pseudo-code de MHAS est présenté par l'algorithme suivant.

Algorithm 6.1 *MHAS*

Require: input : Graph SN(V,E)
Ensure: output : k sub-detected colony communities

1: **repeat**
2: **repeat**
3: Cc_k =AntCDivAoBeeCAgA(Cc_k).
4: **until** ($AntCDivA$o$BeeCAgA$ (Cc_k))=Cc_k
5: **repeat**
6: Cc_k =BeeCAgAo$AntCDivA$(Cc_k).
7: **until** ($BeeCAgA$o$AntCDivA$(Cc_k))= Cc_k
8: **until** ($AntCDivA$o$BeeCAgA$(Cc_k))=($BeeCAgA$o$AntCDivA$(Cc_k))= Cc_k

Les principes des algorithmes $AntCDivA$ et $BeeCAgA$ sont détaillés dans les sous-sections suivantes.

Analyse par décomposition basée sur les colonies de fourmis

Soit $SN(V, E)$ le graphe de construction modélisant le réseau, γ désigne $GScore_{importantOp}$ correspondant à la quantité de phéromone et n représente le nombre des fourmis (sommet du graphe). L'idée de base de l'$AntCDivA$ proposé est décrite dans le pseudo-code de l'algorithme suivant.

Algorithm 6.2 AntCDivA

Require: le graphe $SN(V, E)$
Ensure: Structure optimale de communauté C^*
1.Phase d'initialisation
Initialiser les traces de phéromone sur les sommets du G par la valeur γ_0
2.2 Phase de construction de solution
Placer la fourmi k sur une partition de départ p_k
Initialiser sa mémoire privée M^k par p_k
while l'opérateur de décomposition est encore possible **do**
for $k = 1$ to n **do**
fourmi k Choisira de se déplacer de la partition courante p_i vers la partition p_j selon le principe $DivA$ décrit précédemment
end for
$M^k = p_j$
3.3. Phase d'évaluation et de mise à jour
for $k = 1$ to n **do**
Evaluer la structure de communauté obtenue C_k de la fourmi k
Mettre à jour la meilleure communauté C_*
end for
end while

Comme indiqué dans le pseudo-code précédent obtenu après l'initialisation de chaque sous-partition avec la valeur $GScore_{importantOp}$ (trace de phéromone), l'étape de construction a été lancée et gérée par une colonie de fourmis ayant visité les partitions adjacentes. Nous plaçons une fourmi initiale k sur une partition initiale p_k. Ensuite, les fourmis sont passées d'un niveau hiérarchique descendant à un autre en exécutant le principe $DivA$. Une fois qu'une fourmi a construit la meilleure structure de communauté, elle indique la quantité de phéromone à déposer reflétée par $GScore_{importantOp}$ en fonction de la partition trouvée. Ensuite, l'évaluation de cette dernière partition a eu lieu en mettant l'accent sur la fonction objective définit dans l'equation (7.2), à savoir la modularité reposant sur la dépendance et la similarité (Q_{DS}). En outre, les partitions ayant la moindre valeur de fonction de qualité ($DivQ_{DS}$) sont décomposées du communauté. Contrairement aux algorithmes de fourmis standards, $AntCDivA$ utilise deux informations heuristiques qui favorisent respectivement les couples de sommets minimisant la similarité entre les sous-partitions et la dépendance entre les couples des noeuds. Une trace de phéromone entre deux sous-partitions d'une communauté fait référence à la désirabilité que les deux sous-partitions appartiennent à la même partition. En se déplaçant, les fourmis construisent par incrément des solutions au problème d'optimisation introduit. Une fois qu'une fourmi a construit une communauté ou lorsque la communauté est en cours de construction, la fourmi évalue la partition (partielle) et dépose la phéromone dans la sous-partition qu'elle a utilisée.

Cette information de phéromone dirigera la recherche des futures fourmis. L'intensité de ces traces permet d'éviter la convergence vers une région sous-optimale, de diversifier ainsi la recherche et avoir plus de chance d'atteindre un optimum global.

Analyse par agrégation basée sur les colonies d'abeilles

Le processus d'agrégation introduit, basée sur la colonie d'abeilles, considère initialement chaque membre du réseau social comme une communauté détectée distincte et place le scout des abeilles dans la partition ayant $GScore_{importantOp}$ le plus élevé. La deuxième étape consiste à évalue la fonction objective des partitions visitées par les abeilles scoutes. En outre, à chaque itération, nous optimisons la fonction objective définie dans l'equation (6.2). Les abeilles ayant la meilleure fonction de qualité et les partitions visitées sont sélectionnées respectivement comme « les scoutes » et comme « voisinage de recherche ». En outre, l'analyse par agrégation améliore les recherches dans le voisinage des communautés les plus prometteuses. Cependant, pour chaque chemin, seulement l'abeille ayant la meilleure fonction objective est choisie pour former la prochaine population d'abeilles. Afin de mieux comprendre $BeeCAgA$, nous résumons, dans l'algorithme 6.3 , notre méthode de colonie d'abeilles.

Algorithm 6.3 *BeeCAgA*

Require: Graphe $SN(V, E)$
Ensure: Structure optimale de communauté C^*
1: $C = \{\{m_1\}, \{m_2\}, ..., \{m_n\}\}$ /*Initialisation*/
2: Initialiser les sous-partitions avec la valeur de $GScore_{importantOp}$ la plus élevée
3: Evaluez la fonction objective et placez la reine des abeilles parmi les membres ayant les liens les plus élevés
4: **while** l'opérateur d'agrégation est encore réalisablee **do**
5: **repeat**
6: 1. sélectionner la partition pour le voisinage de recherche
7: 2. sélectionner la partition ayant la valeur $Score_{importantOp}$ la plus élevée
8: 3. Recruter des abeilles pour la partition sélectionnée et évaluer leurs fonctions objectives
9: 4. Sélectionnez l'abeille reine de chaque patch
10: 5. Informer toutes les abeilles du changement au niveau de la structure communautaire
11: 6. Placez une autre abeille sur la prochaine partition importante non visitée
12: **until** (chaque colonie d'abeilles construit ses partitions)
13: **end while**

L'algorithme *BeeCAgA* commence par le placement des abeilles, à savoir les sous-groupes de la partition initiale, dans l'espace de recherche. Ensuite, la fonction de qualité des partitions visitées par les abeilles scoutes est évaluée. Les abeilles qui ont la meilleure fonction objective sont ultérieurement choisies comme des «abeilles sélectionnées» et les partitions visitées par celles-ci sont choisies pour le voisinage de recherche. Ainsi, l'algorithme effectue des recherches au voisinage des partitions sélectionnées en recrutant plus d'abeilles pour rechercher à proximité des meilleures partitions. Les abeilles sont choisies directement en fonction des conditions associées aux partitions qu'elles visitent. En outre, les valeurs de fonction qualité sont prises en considération pour déterminer la modularité basée sur la similarité et la dépendance. Les recherches à proximité de meilleures partitions, qui représentent des sous-communautés plus prometteuses, sont plus détaillées en recrutant plus d'abeilles pour les suivre que les autres abeilles sélectionnées. Cependant, pour chaque parcelle, seule l'abeille ayant la meilleure fonction objective sera sélectionnée pour former la prochaine population d'abeilles.

6.5 Conclusion

Dans ce chapitre, nous avons présenté notre méthode de classification hiérarchique mixte intégrant l'intelligence en essaim. L'optimisation globale d'un cluster nécessite que les sous-partitions soient optimisées indépendamment et en parallèle (Frenken and Mendritzki, 2012, renken and Mendritzki, 2012). Pour cette raison, nous avons défini une nouvelle fonction de modularité basée sur la similarité et la dépendance entre les sommets du graphe. La fonction de modularité introduite améliore la convergence vers un optimum global. Le niveau optimal de modularité déduit a minimisé le temps requis pour optimiser globalement le processus hiérarchique mixte suggéré.
Afin d'éviter la convergence vers un optimum local et pour détecter une structure de communauté hiérarchique globale, nous avons étendu notre technique de classification hiérarchique combinée et nous l'avons formulée comme un problème d'optimisation. Nous avons considéré la classification comme une méthode génétique dans laquelle nous avons appliqué l'opérateur génétique "crossover" et combiné deux sous-partitions pour produire des partitions avec des bits associes aux deux. En outre, l'opérateur de "mutation" développé a été utilisé pour changer le placement des sous-partitions sur d'autres partitions. La convergence de la classification génétique prouve que la structure communautaire détectée n'était pas forcement optimale, d'où l'intégration de la technique d'intelligence en essaim dans notre classification mixte qui a permis, à travers, la possession d'une mémoire, de former une solution optimale. En effet, le procédé d'agglomération basé sur les colonies d'abeilles introduit prouve l'optimisation parallèle des partitions. En effet, la fusion de sous-partitions ayant une grande modularité lors du processus d'agglomération reposant sur la colonie d'abeilles signifie que, si une mutation est effectuée dans l'une des partitions, la fonction objective de l'autre partition est également affectée. De plus, la possibilité de décomposition fournie dans le processus de division basé sur la colonie de fourmis montre qu'une partition est partitionnée en sous-partitions non chevauchante, de sorte qu'aucune interdépendance ne subsistera entre les partitions. Cela implique que les sous-partitions sont optimisées indépendamment et en parallèle.
Malgré l'optimalité de structure communautaire obtenue en combinant la classification et l'optimisation, l'interaction dynamique des utilisateurs des réseaux sociaux provoque la nécessité de suivre l'évolution des communautés au fils du temps. Il est important non seulement de détecter les communautés, mais également de suivre l'évolution de la structure du réseau au fil du temps. Naturellement, les réseaux sociaux évoluent fréquemment. Le suivi de cet évolution sera étudié dans le chapitre suivant.

Chapitre 7

Analyse des événements dans les réseaux sociaux

7.1 Introduction

Un réseau social dynamique est un ensemble d'observations des réseaux sociaux à différents intervalles, soulignant l'évolution des relations dans le temps(Falkowski et al., 2008, alkowski et al., 2008). La principale différence entre les communautés dynamiques et celles statiques réside dans l'aspect temporel lors de l'analyse de la structure et des interactions. Suivre et comprendre l'évolution sont deux tâches difficiles qui nécessitent l'identification des événements importants pouvant survenir aux communautés et aux individus. La plupart des réseaux sociaux évoluent avec le temps en raison des changements fréquents : des noeuds peuvent rejoindre ou quitter des communautés ; des nouveaux liens peuvent être créés et d'autres peuvent disparaître et probablement former des nouvelles communautés. En outre, les utilisateurs des réseaux sociaux se joignent à divers les événements. Pour cette raison, la participation à des événements communs peut développer des nouvelles communautés. En d'autres termes, les informations sur les événements permettent de déduire des sous-groupes, d'où la nécessité de suivre les variations des événements.
Dans ce chapitre, nous présentons dans la section 7.3notre approche de suivi de l'évolution des communautés et dans la section **??** leurs mobilité durant les événements dans les réseaux sociaux(Toujani et al., 2018b, oujani et al., 2018b) (Toujani et al., 2018a, oujani et al., 2018a)(Toujani and Akaichi, 2019b, oujani and Akaichi, 2019b).

7.2 Analyse de mobilité des événements dans les réseaux sociaux

Les réseaux sociaux représentent une source abondante d'événements. En fait, la mobilité d'événement inclut le changement d'opinion, la vélocité de diffusion d'information et la localisation des utilisateurs. Nous modélisons l'évolution des communautés en fonction des événements en appliquant les techniques d'apprentissage automatique pour analyser la mobilité des événements. Pour cette raison, nous développons la méthode de l'arbre de décision pour prédire le changement d'opinion par rapport à sa cooccurrence spatiale et temporelle. Par conséquent, la mesure d'entropie définie repose principalement sur des attributs spatio-temporels tels que les attributs de branchement.
Comme l'illustre la figure ci-dessous, la méthode d'arbre de décision pour la mobilité et l'exploration des événements se base sur trois processus principaux : Collecte des données brutes, pipeline du traitement pour l'extraction d'événements temporels et le processus de mobilité des événements.

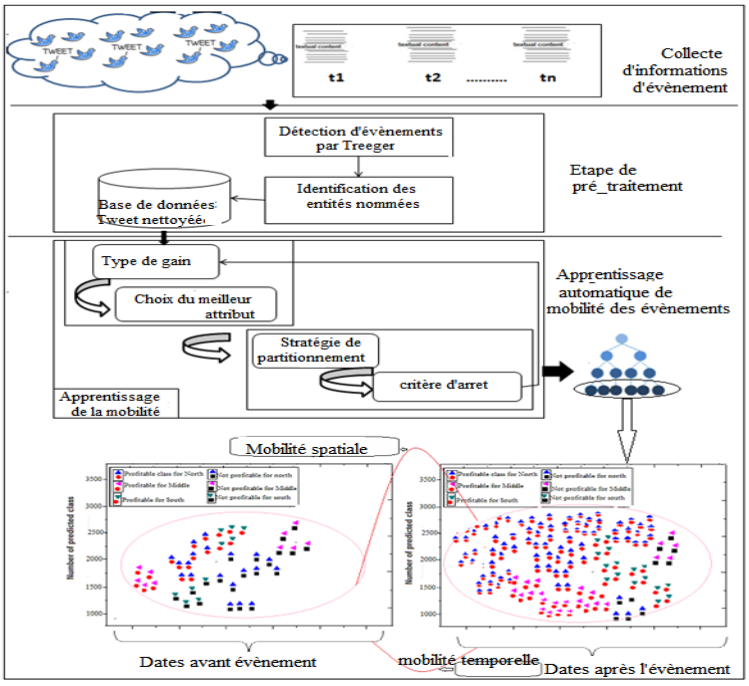


FIGURE 7.1: Architecture de l'approche de mobilité des événements

Dans ce qui suit, nous allons détailler le processus de l'approche de mobilité des événements [Toujani et al2019]. Après la collecte et le prétraitement des évènements, nous présentons notre stratégie d'apprentissage de mobilité des évènements reposant sur les arbres de décision. En effet, les arbres de décision font partie des méthodes les plus populaires en apprentissage supervisé et constituent une technique préliminaire puissante de

data mining. Cette méthode est souvent utilisée pour les tâches de classification et de prédiction. Ses avantages sont : la construction rapide de l'arbre et l'interprétation facile des règles de décision générées. En fait, un arbre de décision est un organigramme représenté sous forme d'arbre hiérarchique qui se compose de trois éléments fondamentaux :

— Un noeud de décision qui correspond à un attribut. Cependant, le noeud racine est celui situé à la partie supérieure de l'arbre ; il n'a pas de noeud parent. Dans notre travail, il s'agit d'un tweet, statut ou commentaire publié sur un réseau social.
— Une branche qui correspond à l'une des valeurs des attributs possibles (le résultat du test). Elle produit un autre noeud de décision ou à une feuille.
— Une feuille qui inclut des objets généralement appartenant à la même classe, ou des objets très similaires. Elle est étiquetée par une classe unique (classe d'opinion positive, negative ou floue (fuzzy)

La représentation graphique des arbres de décision aide à rendre cette technique plus facile à interpréter, même lorsque les arbres sont assez complexes. En fait, dans ce cas, les procédures graphiques peuvent être développées pour simplifier l'interprétation de ces arbres complexes. De toute évidence, la représentation graphique, la nature hiérarchique et la flexibilité des arbres de décision rendent ce classificateur plus populaire.
La méthode développée de l'arbre de décision utilise la même représentation qu'un arbre de décision standard. Cependant, sur chaque noeud de l'arbre, nous intégrons le type de gain à appliquer afin de choisir l'attribut de sélection, plus particulièrement les attributs spatio-temporels.

7.2.1 Collecte des données évènementielles

Le service de localisation permet aux utilisateurs des réseaux sociaux de publier des messages contenant des informations géo-spatiales. En fait, l'émergence de ce service a poussé les chercheurs à extraire les informations spatio-temporelles afin d'obtenir les données des événements géo spatiaux en temps réel. Les méthodes de détection d'événements en temps réel présentent certaines faiblesses. Par exemple, les utilisateurs des réseaux sociaux ne fournissent pas d'informations de localisation dans leurs profils, d'où l'ambiguïté d'emplacement extrait des messages contextuels. Ainsi, afin de garantir la pertinence de la localisation contextuelle extraite, nous nous concentrons sur l'extraction de la localisation à partir des postes contextuels temporels et nous explorons la mobilité des événements en utilisant la technique d'apprentissage automatique. Pour extraire les publications contextuelles, nous avons appliqué le Framework Twitter 140dev [1]. Les données d'entrée représentent un ensemble de postes temporaires. Nous avons collecté une liste de publications textuelles (Tp) obtenues à une date précise t.

1. http ://140dev.com/free-twitter-api-source-code-library/

7.2.2 Phase de prétraitement

La langue du texte décrivant un évènement joue un rôle primordial dans la tâche de prétraitement ainsi que celle de classification. Par conséquent, il est nécessaire de se concentrer sur un seul langage pour permettre un prétraitement efficace des publications, en extrayant ses caractéristiques et en créant le modèle de classification. Ainsi, une traduction en anglais a été appliquée à toutes les publications extraites. Nous avons, ensuite, utilisé un lemmatiseur pour annoter nos messages textuels non structurées et nous avons employé le lemmatiseur TreeTagger développé par Helmut dans le cadre du projet TC [2]. En fait, nous avons effectué le parcours du contenu du message publié en fonction du résultat d'annotation généré par TreeTagger. Nous avons également identifié les entités nommées de tous les objets texte, telles que DATE, PERSON, LOCATION. En outre, en plus de l'attribut d'opinion extraite, ces entités nommées «extraites», en particulier celles liées à LOCATION et DATE, servent comme attributs à chaque message extrait.
Cette étape se termine par l'obtention d'une base de données nettoyée et injectée en tant qu'entrée dans le processus introduit de classification, à savoir la méthode de l'arbre de décision définit.

7.2.3 Processus d'apprentissage basée sur la mobilité des évènements

La modélisation de mobilité des évènements suit le principe de la technique la plus utilisée de construction d'un arbre : Top Down Induction of Decision Tree(TDIDT). Ce principe est décrit dans la figure suivante :

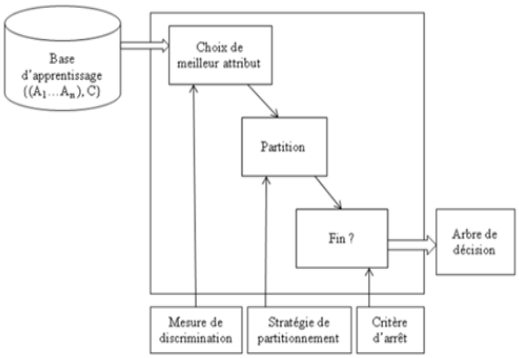


FIGURE 7.2: Principe général d'apprentissage de mobilité

1. Condition d'arrêt : si tous les éléments du noeud sont dans la même classe, alors le noeud est une feuille étiquetée par cette classe. Les classes, dans notre contexte, indiquent

2. http ://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

si le message est profitable pour une région bien déterminée ou non. **2.** Sinon, faire les étapes suivantes :
(a) Choisir un attribut et l'assigner au noeud courant. En fait, la méthode proposée repose principalement sur l'exploitation de trois attributs : lieu, date et opinion extraite [].
(b) Partitionner les publications dans des sous ensembles selon les valeurs de cet attribut.La liste des nouveaux noeuds est ajoutée aux fils du noeud.
(c) Appliquer la procédure récursivement sur les nouveaux noeuds Par conséquent, le modèle de classification proposé repose sur une analyse à la fois temporelle et spatiale et décrit le mouvement d'extraction d'événements à un moment défini. La présentation considérée dans l'arbre de décision utilisé dans la méthode proposée est la même que celle employé dans l'arbre de décision standard.
Cependant, sur chaque noeud de l'arbre, nous avons intégré le type de gain à appliquer afin de choisir l'attribut de sélection, plus particulièrement les attributs spatio-temporels. L'algorithme d'arbres de décision définit les composants suivants :
Choix du meilleur attribut et gain d'information :
Cet élément, généralement basé sur la théorie de l'information, représente un élément essentiel dans la construction de l'arbre de décision (Quinlan, 1986, uinlan, 1986). A chaque noeud de décision, nous choisissons ,selon un critère de choix parmi une liste d'attributs, l'attribut ayant le pouvoir le plus discriminant sur les classes. Par conséquent, le choix du meilleur attribut est l'étape la plus importante dans l'arbre de décision. Les mesures les plus appliquées sont celles proposées par Quinlan, à savoir le gain d'information et le rapport du gain(Quinlan, 2014, uinlan, 2014). Le but de cette étape est de sélectionner le type du gain qui va être appliqué sur chaque noeud de l'arbre pour choisir le meilleur attribut. En fait, la partie relative à la sélection des attributs, étroitement liée à la quantité d'information exprimée par la valeur d'un attribut donné, influe la complexité de l'arbre de décision.
Nous formalisons, dans ce travail, l'idée à la base de la mesure d'entropie introduite pour calculer le gain d'information de la manière suivante :
Nous présentons le jeu des données d'entrée sous cette forme :

$$D = (T, A \cup C) \tag{7.1}$$

où :

$T = \{t_1, t_2, ..., t_n\}$ désigne l'ensemble des échantillons de données (tweet temporellement extrait),

$A = \{a_1, a_2, ..., a_m\}$ correspond à l'ensemble des attributs de condition. Ces attributs présentent des informations spatiales, des informations temporelles et des opinions extraites, L'attribut d'opinion extraite est spécifié en appliquant la méthode d'analyse des sentiments définie dans(Toujani and Akaichi, 2016, oujani and Akaichi, 2016).

$C = \{c_1, c_2, ..., c_n\}$ est l'attribut d'étiquette de classe qui a q valeurs distinctes définissant j classes distinctes. Le tableau suivant présente un exemple d'échantillon.

Tweet	Location (attribut spatial)	Date (attribut temporel)	Opinions extraites	Classe
1	Nord	Avant l'événement		profitableN
2	Sud	Après l'événement	Négative	NotprofitableS
3	Milieu	Après l'événement	Positive	profitableM
4	Nord	Après l'événement	Positive	profitableN
5	Nord	Après l'événement	Positive	profitableN
6	Nord	Avant l'événement	Positive	profitableN
7	Sud	Après l'événement	Flou	Not profitableS
8	Nord	Avant l'événement	Négative	profitableN
9	Milieu	Avant l'événementt	Positive	profitable

TABLE 7.1: Représentation des publications échantillons

Dans l'exemple ci-dessus, les attributs « Location » et «Date» sont extraits au cours du processus d'identification des entités nommées. Alors que l'attribut «opinions extraites» est défini en référence à notre méthode d'analyse des sentiments présentée dans (Toujani and Akaichi, 2016, oujani and Akaichi, 2016). En outre, le label de «classe» indique si un message est profitable ou non pour la prise de décision. Dans ce cas, l'entropie est calculée :

$$entropy(T) = -\sum_{i=1}^{k} p_i \log_2 p_i \tag{7.2}$$

où p_i indique la probabilité que le message d'échantillon arbitraire appartienne à une classe étiquetée et k représente le nombre total des messages d'échantillon. De plus, chaque attribut a_i possède v valeur distincte p_1, p_2,...,p_v. Cet attribut peut être appliqué pour diviser T en v sous-ensembles S_1, S_2,..., S_v. S_{ij} est le nombre d'échantillons de classe c_i dans un sous-ensemble S_j. L'entropie de l'attribut a_i est donnée par :

$$E(a_i) = \sum_{j=1}^{v} \sum \frac{s_{1,j} + ... + s_{m,j}}{S} I(s_{1,j}, ..., s_{m,j}) \tag{7.3}$$

L'attribut avec le gain d'information le plus élevé est sélectionné comme attribut de branchement. Pour un sous-ensemble donné S_j, le gain d'information est calculé de la manière suivante :

$$I(s_{1,j}, ..., s_{m,j}) = -\sum_{i=1}^{m} p_{i,j} \log_2 p_{i,j} \tag{7.4}$$

où $p_{ij} = s_(i,j)/S$ est la probabilité qu'un échantillon de j appartienne à la classe c_i. Par conséquent, le gain d'information de l'attribut a_i est donné par :

$$Gain(a_i) = I(s_{1,j}, ..., s_{m,j}) - E(a_i) \tag{7.5}$$

Nous calculons le gain d'information de chaque attribut de condition. L'attribut avec le gain d'information le plus élevé est celui le plus informatif et le plus discriminant de l'ensemble donné.

Stratégie de partitionnement :

L'ensemble d'apprentissage est divisé selon l'attribut sélectionné. Dans l'arbre de décision classique, la stratégie de partitionnement consiste à partitionner le jeu de données en fonction de toutes les valeurs possibles d'attribut. Cependant, lors de la construction de l'arbre de décision introduit, nous proposons une alternative pour partitionner la base des messages collectées. Cette dernière alternative nécessite la considération des attributs spatiaux (A_s) et temporels (A_t) avec P_v valeur possible pour partitionner la base d'apprentissage et construire le noeud N. Par conséquent, l'ensemble d'apprentissage T composé de messages échantillon e_i est partitionné en P_v sous-ensemble T_v^N tel que :

$$\forall e_i \in T^N, if e_i(A_s, A_t) = v_{lk}, then e_i \in T_k^N, 1 < k < P_v \tag{7.6}$$

où v_{lk} est une valeur appartenant à l'ensemble des valeurs possibles de A_s ,A_t, et

$$T^N = \bigcup_{k=1,...,P_v} T_k^N \tag{7.7}$$

$$\forall x, k = 1, ..., P_v, x \neq k, T_x^N \bigcap T_k^N \tag{7.8}$$

Conditions d'arrêt : Les critères d'arrêt déterminent les conditions d'arrêt du processus de partitionnement. Il permet de décider s'il est nécessaire de continuer, pour un ensemble d'exemple de la base d'apprentissage, à développer l'arbre. Les raisons de l'arrêt peuvent être un nombre trop faible d'exemples dans l'ensemble considéré ou bien si tous les exemples de l'ensemble ont la même classe.

Les critères d'arrêt sont les mêmes pour l'apprentissage de mobilité des événements en adaptant l'évaluation de ces critères.

Le choix de ces composantes d'arbre (test de type du gain, choix du meilleur attribut, stratégie de partitionnement, critères d'arrêt) fait la différence majeure entre les algorithmes d'arbre de décision. L'algorithme utilisé dans notre approche d'apprentissage de mobilité des événements est l'algorithme C4.5 [56]. Il permet d'obtenir un modèle de prédiction représenté sous forme d'arbre de décision facilement compréhensible et interprétable.

La prédiction du changement des opinions et de structure communautaire nécessite l'analyse de cette évolution pour améliorer les décisions précises pour la gestion des catastrophes et la planification des secours. A ce propos, dans la section suivante, nous allons montrer que les réseaux sociaux jouent un rôle important à aider les gens en cas de catastrophe naturelle.

7.3 Analyse de l'évolution des événements pendant les catastrophes naturelles

La gestion des risques de catastrophes naturelles est améliorée en explorant les données diffusées dans les réseaux sociaux. La nécessité d'extraire rapidement des informations significatives à partir des grandes quantités de données générées par les réseaux sociaux est en augmentation, en particulier pour faire face aux catastrophes naturelles. De ce fait, des nouvelles méthodes doivent être développées pour soutenir profondément l'analyse immersive des données sociales.

Les réseaux sociaux ont non seulement révolutionné la manière par laquelle les gens communiquent et interagissent, mais ils jouent également un rôle primordial dans les systèmes de gestion des catastrophes. En cas de catastrophe naturelle, les médias sociaux sont généralement utilisés afin de :

- Prendre les précautions nécessaires. Bien que personne ne puisse prévoir l'apparition d'un séisme, des messages instantanés peuvent alerter les personnes en cas d'urgence.
- Effectuer les interventions nécessaires, à travers la conversation ou communication synchrone, pendant et après la catastrophe.
- Se récupérer après une catastrophe naturelle.

Fondamentalement, le processus de détection des événements implique la transformation des publications non structurées perdant des informations pertinentes dans leurs formats originaux. En ce sens, il est important de gérer le flou et l'incertitude des connaissances réelles. La signification du contenu textuel du message des événements vari selon les domaines d'application. En outre, les indices de subjectivité des informations sur les événements utilisés avec des sens objectifs constituent une source importante de détection des erreurs sur les événements. Ainsi, en exploitant les avantages des approches floues, il est possible de développer des mécanismes plus puissants pour représenter les informations sur les événements car ces mécanismes nécessitent la vérification des concepts d'incertitude dans le modèle de détection des informations sur les événements finaux et la désambiguïsation du sens des mots.

Dans cette thèse, les publications sur les événements nécessitent des connaissances incertaines, comme les autres tâches de détection des événements impliquant un concept imprécis. En fait, le concept 'degré de danger' que nous utilisons est sous forme linguistique (proches, forts, très forts, faibles, moyens, etc.). Ainsi, la logique floue est efficace face aux variables linguistiques. En fait, notre méthode de détection des événements et de suivi de structure communautaire (Toujani and Akaichi, 2019b, oujani and Akaichi, 2019b) peut être divisée en trois étapes principales (voir la figure7.3).

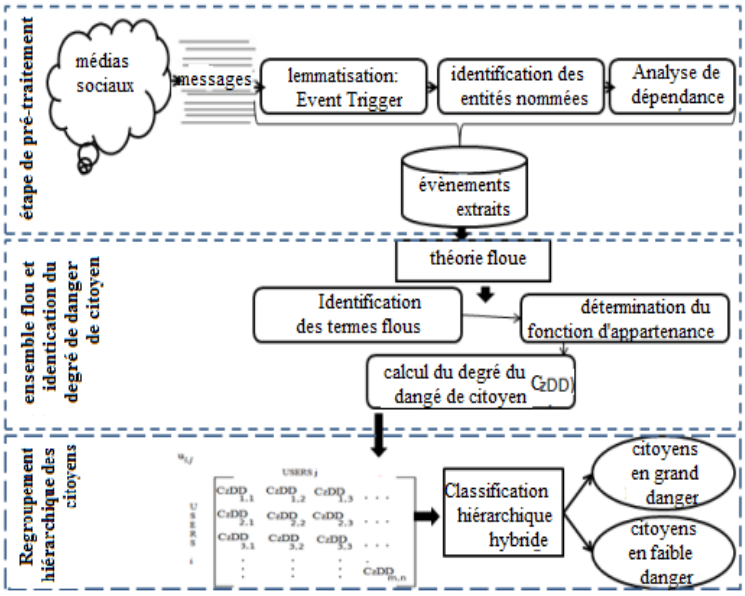


FIGURE 7.3: Architecture de l'approche de détection d'événements et de suivi de structure communautaire

Dans ce qui suit, nous allons détailler le processus de l'approche proposée de détection des événements et de suivi de structure communautaire. Dans la première partie, nous commençons par le prétraitement des messages extraits des médias sociaux. Ensuite, nous définissons le degré de danger en utilisant la théorie floue. En fait, cette étape est composée de trois sous-étapes : (1) identification des termes flous, (2) détermination de la fonction d'appartenance (3) calcul du degré de danger des citoyens (en anglais Citizen Danger Degree (CzDD)). La troisième étape consiste à trouver des groupes de citoyens présentant des degrés de danger proches.

7.3.1 Processus d'extraction des publications d'événements

Notre processus de détection des événements a la forme d'un pipeline. Dans notre contexte, ce dernier processus est composée de trois composants principaux :

i) Déclencheur d'événements d'actualité (Event news Trigger en anglais) : décrit avec précision l'occurrence d'un événement.

ii) Arguments des publications d'événements : sont les détails impliqués dans un événement, c'est-à-dire ceux ayant un déclencheur d'événements d'entités associées. La méthode

introduite pour extraire les événements à partir des publications de médias sociaux commence par la détermination de la mention de l'événement qui représente un déclencheur d'événement. Dans la deuxième étape, nous extrayons les différents arguments tels que la date de publication des événements et les noms des participants concernés. En conséquence, le processus d'extraction d'événements est divisé en trois sous-étapes principales :

1. Identification et extraction des déclencheurs d'événements liés aux catastrophes naturelles reposant sur un lexique créé contenant un ensemble de déclencheurs d'événements possibles de catastrophes naturelles.

2. Identification des entités nommées liées aux événements identifiés.

3. Analyse de dépendance entre les entités extraites et le message de l'événement (déclencheur de l'événement). Nous réalisons une analyse de dépendance sémantique-syntaxique.

1. **Détection de déclencheur des évènements** Afin d'extraire les événements liés aux risques naturels et décrits dans un texte non structuré, nous appliquons, d'abord, la technique de lemmatisation pour trouver la forme canonique des mots. En fait, nous créons une liste de lemmes considérés comme des déclencheurs possibles d'événements potentiels décrivant des dangers naturels. Ensuite, nous utilisons un lemmatiseur pour annoter nos données textuelles non structurées en appliquons le lemmatiseur TreeTagger [3].
   Considérant l'exemple suivant, "In 2013, an anonymous ignited a forest in the north of Tunisia. " extrait des publications sociales non structurées. TreeTagger, appliqué à cet exemple, donne les résultats présentés dans la figure 7.4.

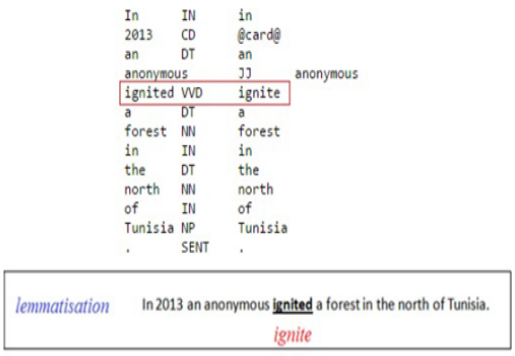


FIGURE 7.4: Résultats obtenus en utilisant TreeTagger

En fait, TreeTagger renvoie la forme canonique de chaque mot. Par exemple, le mot "ignited" (verbe) devient "ignite" (Lemma). Ensuite, en considérant les résul-

3. http ://www.cis.uni-muenchen.de/ schmid/tools/TreeTagger/

tats d'annotation générés par TreeTagger, nous parcourons le contenu de la publication. Nous détectons, à chaque mot, un lemme apparaissant simultanément accompagné d'une liste de lemmes (lexique créé) et du texte traité. En outre, nous récupérons le mot correspondant à chaque lemme en tant que déclencheur possible d'événement lié à un risque naturel potentiel ou à un nom d'événement.

2. **Identification des entités nommées liées aux déclencheurs des événements identifiés**

La reconnaissance des entités nommées est une sous-tâche d'extraction des informations. Nous identifions généralement les entités nommées comme tous les objets texte faisant référence aux noms des personnes, des lieux et des sociétés présentés dans les publications sociales. Dans cette étude, nous avons utilisé une version personnalisée de l'outil de reconnaissance des entités nommées de Stanford afin d'améliorer la précision de la reconnaissance des entités nommées (en anglais « Named Entities Recognition » (NER)). L'application du système NER sur le même exemple a donné le résultat suivant :
In <DATE>2013< /DATE> <PERSON> an anonymous <PERSON> ignited a forest <LOCATION> in the north of Tunisia< /LOCATION>.
Nous remarquons que le système a reconnu trois différentes entités nommées <DATE>, <PERSONNE> et <EMPLACEMENT> considérées dans la méthode introduite en tant que participants de l'événement liés au déclencheur principal de publication de l'événement. Cependant, ces entités extraites sont liées au déclencheur de message d'événement principal afin de confirmer la cohérence des informations extraites. Pour cette raison, nous avons réalisé une analyse sémantique ou une analyse lexicale de dépendance.

3. **Analyse de dépendance entre les entités nommées extraites et les déclencheurs d'événements récupérés**

Après l'extraction des publications sociales nécessaires, nous montrons que les données extraites sont liées à l'événement principal. En d'autres termes, nous prouvons qu'il existe une relation de dépendance entre les différentes informations extraites et le déclencheur de l'événement principal détecté. Pour cela, nous avons utilisé le système Stanford Parser pour générer une description des relations grammaticales entre les mots d'une phrase, comme illustré dans la figure 7.5. Nous avons exploité, ensuite, cette description pour déterminer s'il existait des liens de dépendance entre les différents termes constitutifs de la phrase analysée, y compris les entités extraites et les déclencheurs des événements.

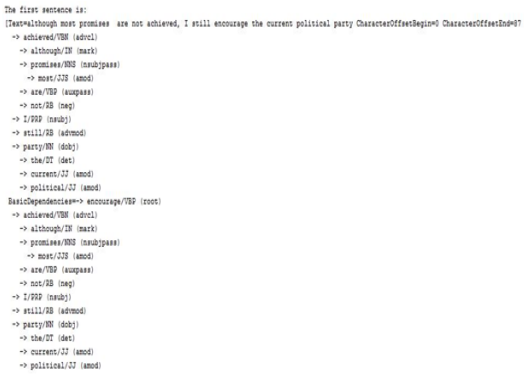


FIGURE 7.5: Résultat d'analyse de dépendance utilisant Stanford Parser

7.3.2 Théorie des ensembles flous et calcul du degré de danger des citoyens

Etape 1 : Identification des concepts flous

L'objectif de cette étape est de définir les parties floues de la liste des événements détectés dans le processus d'extraction des événements stockés dans un thésaurus (THS). Après une analyse détaillée des événements extraits, nous récupérons, en fonction d'une valeur particulière d'une variable linguistique, les différents concepts flous dans un nouveau dictionnaire de termes flous nommés DFT. Ainsi, les variables linguistiques sont des termes utilisés lors de la description d'une situation, d'un phénomène ou d'un processus, tels que la température, la vitesse, etc. Par exemple, Faible, Moyen et Elevé sont les valeurs de la variable linguistique de température. En fait, la définition d'une variable linguistique inclut des informations numériques et linguistiques. Les termes linguistiques sont considérés comme des ensembles flous de l'univers du discours utilisant des valeurs numériques. Après l'extraction de la liste (THS), nous générons une base des variables linguistiques destinées aux risques naturels.

Cette dernière base de variables est ensuite utilisée pour identifier les concepts flous. La phase de détermination des concepts flous prend, en entrée, tous les événements extraits. Au cours de cette étape, nous vérifions l'existence du concept dans la base des termes flous qui produit la liste des concepts vagues. Le tableau suivant présente des exemples de concepts flous pour le cas de deux risques naturels : tremblements de terre et tempêtes.

Type d'évènement	Les termes flous
Tremblement de terre	Micro, Very minor, Minor, Light, Moderate, Strong, Very Strong, Major, Devastator
Tempêtes	Low, Moderate, Strong, Very Strong, Devastating

TABLE 7.2: Des concepts confus de deux catastrophes naturelles : tremblements de terre et tempêtes

Etape 2 : Calcul du degré de danger des citoyens (CzDD) : fonctions d'appartenance

Nous associons les fonctions d'appartenance pour les termes flous extraits des publications sociales. Les fonctions d'appartenance les plus utilisées dans la logique floue sont : la fonction triangulaire, la fonction croissante monotone, la fonction décroissante monotone et la fonction trapézoïdale [24].

Dans ce travail, nous utilisons les trois premières fonctions. Nous n'employons pas la fonction trapézoïdale, car elle génère généralement des degrés d'appartenance de valeur 1. Ainsi, chaque terme a sa propre fonction d'appartenance et le degré de danger des citoyens est obtenu comme suit :

$$CzDD_T(EI) = \begin{cases} 0, & \text{if } EI \leq \ell \text{ ou } EI \geq \beta \\ \frac{EI-\ell}{\alpha-\ell}, & \text{if } \ell < EM \leq \alpha \\ \frac{\beta-\ell}{\beta-\alpha}, & \text{if } \alpha < \ell < \beta. \end{cases} \tag{7.9}$$

où EI correspond aux informations (I) extraites de l'évènement (E), ℓ indique une valeur basse de EI, β est une valeur élevée de EI et EM représente la valeur modale. Les tableaux 7.3 et 7.4 montrent la partition des fonctions floues pour chaque événement de description des tremblements de terre.

TABLE 7.3: Concepts flous pour tremblement de terre

Concepts flous pour tremblement de terre				
Micro	$\leq 1,9$	$---$	$---$	Fonction décroissante monotone
very minor	2	2.5	2.9	triangular function
Minor	3	3.5	3.9	fonction triangulaire
Light	4	4.5	4.9	fonction triangulaire
Moderate	5	5.5	5.9	fonction triangulaire
strong	6	6.5	6.9	fonction triangulaire
Very stron	7	7.5	7.9	fonction triangulaire
Major	8	8.5	8.9	fonction triangulaire
Devastator	$\geq 9,0$	$---$	$---$	Fonction croissante monotone

TABLE 7.4: Concepts flous du "tempête"

Concepts flous du "tempête"				
Low	≤ 100	$---$	$---$	Fonction décroissante monotone
Moderate	100	150	180	fonction triangulaire
strong	180	200	210	fonction triangulaire
Very strong	210	220	250	fonction triangulaire
Devastator	≥ 250	$---$	$---$	Fonction croissante monotone

Etape 3 : Détermination du degré d'appartenance Afin de déterminer le degré de danger des citoyens pour chaque publication sociale extraite, nous mettons l'accent sur le degré d'appartenance de l'information floue exprimant un événement aux ensembles des concepts flous identifiés.

Exemple 1

"According to the records analyzed yesterday, wind gusts reached $138km/h$ in Montauban, Monday at $8:12$y pm."

$138km/h \in [100, 180]$: fonction triangulaire $Czdd_{EN}(Wind - Montauban)$

$= ((138 - 100)/(150 - 100)) = 0.76$

$\Rightarrow$ La vitesse du vent à Montauban présente un danger modéré avec un degré de $0,76$.

Exemple 2

"In the north of the island of Balagne in Cap Corse, in the late afternoon of Saturday, the wind will reach $140 - 170km/h$. At midnight, it will turn north again with violent gusts across the north of the island. "

$170km/h \in [150, 200]$: fonction triangulaire

$CzDD_{DF}(Wind - Balagne) = ((170 - 150)/(180 - 150)) = 0.7$

⇒ La vitesse du vent sur l'île de Balagne présente un fort danger avec un degré de 0,7.

Exemple 3
$9km/h \in [0, 150]$: fonction décroissante monotone
$CzDD_{EN}(Wind - Tozeur) = 9 < 150$
⇒ La vitesse du vent à Tozeur présente un faible danger avec un degré de 1.

7.3.3 Regroupement hiérarchique des citoyens selon le degré de danger

L'analyse de la communauté est utilisée pour filtrer les événements et développer un modèle de gestion des catastrophes et de planification des secours. En fait, après une catastrophe, les médias sociaux aident les membres d'une communauté ainsi que les différents organismes gouvernementaux à discuter l'événement, à partager des informations, à coordonner les efforts de reconstruction et à obtenir des informations sur l'aide nécessaire.
En fait, nous modélisons le réseau social par un graphe $G = (V, E, D)$ où V représente les citoyens dans les réseaux des médias, E désigne les diverses interactions entre eux, à savoir les ensembles des informations échangées, et D est le degré de danger obtenu en appliquant la méthode floue introduite. Nous pondérons chaque arête par la valeur de CzDD définie dans l'équation 7.9.
Ensuite, nous appliquons notre méthode de regroupement mixte des citoyens associant des opérateurs de classification hiérarchique, à savoir les opérateurs d'agglomération et ceux de division.En fait, la technique d'agglomération est basée sur un processus d'agrégation qui réunit deux citoyens ayant la plus grande valeur $CzDD$, tandis que l'opérateur de division repose sur un processus de décomposition basé sur la division d'un groupe de citoyens en deux sous-groupes ayant la plus faible valeur $CzDD$. Pour le processus descendant suggéré, les communautés des citoyens sont divisées itérativement en deux parties en supprimant les arêtes entre les sommets exprimant les plus faibles degrés de danger. Cependant, le processus ascendant commence si chaque noeud ou citoyen constitue une communauté séparée et aboutit à considérer le graphe entier comme une communauté.
Par conséquent, la méthode mixte proposée nécessite l'existence d'une partition initiale de k communautés des citoyens. Il procède par une combinaison successive des opérateurs ascendants et descendants et se termine si une partition stable est obtenue en appliquant l'opérateur d'agrégation ou celui de décomposition. La structure de communauté initiale est obtenue en utilisant la méthode développée dans [25]. Cette technique considère la partition initiale comme un problème d'optimisation combinatoire et le résout en employant la métaheuristique de recherche Taboue. Ainsi, la partition initiale générée est injectée comme entrée à la méthode de classification mixte des citoyens (Mixed Citizens Classification Method (MCCM) en anglais).

— **Algorithme de classification ascendante des citoyens (Bottom-up Citizens Clustering algorithm (B - upCCA) en anglais)** Au début, chaque sommet représente une communauté distincte. $B - upCCA$ considère initialement que chaque citoyen (utilisateur du réseau social) constitue une communauté :

$C = \{\{Cz_1\}, \{Cz_2\}, \{Cz_3\}, ..., \{Cz_k\}\}$
A chaque itération, le $B - upCCA$ introduit passe d'un niveau n à l'autre en effectuant une procédure d'agrégation qui consiste à regrouper, dans la même communauté, deux citoyens présentant le niveau de danger le plus élevé. L'algorithme se termine si tous les citoyens sont regroupés dans la même communauté et si la procédure d'agrégation n'est plus réalisable.
En fait, pour un niveau donné n, $B-upCCA$ examine toutes les paires de groupes constituant la communauté c_k et trouve les deux groupe c_α et c_β ayant la plus grande valeur de $CzDD$. Le pseudo-code de $B - upCCA$ est est obtenu en appliquant l'algorithme 7.1.

Algorithm 7.1 $B - upCCA$

Require: $G(V, E, D)$
Ensure: output : k groupes sous-détectés de citoyens
1: $K := n - 1$
2: **repeat**
3: $C_k = C_k \backslash \{c_\alpha, c_\beta\} \cup \{c_\alpha \cup c_\beta\}$ such that.
$CzDD(c_\alpha, c_\beta) = \max(CzDD_{i,j}(cz_i, cz_j))$.
$CzDD(C_k) := CzDD(C_k - CzDD(c_\alpha, c_\beta))$ $k := k - 1$
4: **until** K=2

Où C_k représente les k groupes des citoyens générés, c_k désigne les k sous-groupes détectés, c_z est l'ensemble des citoyens du réseau social constituant c_k et n correspond au niveau hiérarchique d'agglomération relatif.

— **Algorithme de classification descendante des citoyens (Top-Down Citizens Clustering Algorithm ($Top - DCCA$) en anglais)** Le $Top - DCCA$ introduit est basé sur le processus descendant et le principe de décomposition. En fait, le $Top - DCCA$ introduit est similaire au $B - upCCA$. Toutefois, il est procédé par une construction hiérarchique opposée s'appuyant sur les divisions successives de communauté des citoyens en deux groupes ayant le moins important $CzDD$. En fait, $Top-DCCA$ considère initialement que tous les citoyens des réseaux sociaux constituent une communauté et passent d'un niveau hiérarchique descendant à un autre en effectuant une procédure de décomposition : séparations successives des citoyens ayant la plus petite valeur $CzDD$ en deux groupes. La condition d'arrêt est atteinte lorsque la procédure d'éclatement n'est plus réalisable et que chaque citoyen du réseau social constitue un groupe séparé. Par conséquent, dans l'algorithme 7.2, nous décrivons le pseudo-code de la méthode $Top - DCCA$ proposée.

Algorithm 7.2 $Top - DCCA$

Require: $G(V, E, D)$

Ensure: k groupes sous-détectés de citoyens

1: $K := 1$

2: **repeat**

3: $C_k = C_k \backslash g_k \cup g_k \backslash cz^*, cz^*$ Such that.
$(g_\alpha \backslash m^*, m^*) < Score_{CzDD}(c_\alpha \backslash cz, cz) \forall g_\alpha \in C_k, \forall cz \subset c_\alpha$.
$CzDD(Top - DCCA(C_K)) = CzDD(C_K + CzDD(g_\alpha \backslash cz^*, cz^*))$ $k = k + 1$

4: **until** K=n-1

où C_k représente la communauté généréede citoyens , g_k désigne le k groupe sous-détecté de citoyens à chaque niveau hiérarchique, cz correspond au citoyen de réseau social constituant g_k et n est le niveau hiérarchique associé.
Le nombre des partitions explorées par $Top - DCCA$ est plus important que celui exploré par $B - upCCA$. En fait, pour trouver la meilleure division d'une communauté avec p partitions en deux sous-partitions, nous explorons $2^{(p-1)} - 2$ possibilités.

— **Algorithme de classification mixte des citoyens (Mixed Hierarchical Citizens Clustering method (MHCCM) en anglais)** Le $MHCCM$ développé procède par une combinaison successive des techniques $Top - DCCA$ et $B - upCCA$. En fait, le processus de division reposant sur l'opérateur de décomposition fournit une partition de $(k - 1)$ groupes sur lesquels il suffit d'appliquer l'opérateur d'agrégation pour obtenir le numéro du groupe initial (K) et inversement.
$MHCCM$ se termine si les communautés sous-détectées obtenues restent constantes. Ainsi, le processus de stabilisation est atteint et nous obtenons les mêmes partitions en appliquant la méthode ascendante ou la technique descendante. Par conséquent, le pseudo-code du MHCCM introduit est décrit dans l'algorithme suivant :

Algorithm 7.3 $MHCCM\ (C_k)$

Require: structure communautaire initiale des citoyens (C_k)
Ensure: structure de communauté de citoyens hiérarchisée mixte $(C_k^{'})$
repeat
repeat
C_k =B-upCCA∘Top-DCCA(C_k).
until $(B - upCCA \circ Top - DCCA(C_k\ ))$=$C_k$
repeat
C_k =Top-DCCA∘B-upCCA(C_k).
until $(Top - DCCA \circ B - upCCA\ (C_k\ ))= C_k$
until $(B - upCCA \circ Top - DCCA(C_k\ ))=(Top - DCCA \circ B - upCCAC_k\ ))= C_k$

7.4 Conclusion

Tout au long de notre histoire, les humains ont dû faire face à des événements inattendus, naturels ou provoqués par l'homme, qui ont eu des conséquences graves sur la vie et la propriété de l'homme. Pour cette raison, les gouvernements ont tenté de gérer l'impact de ces événements et d'éviter, ou du moins d'atténuer, leurs conséquences désastreuses. Dans ce chapitre, nous avons défini une méthode permettant de déterminer les cas critiques lors des catastrophes naturelles. Les points les plus importantes sont : i) la détection et le suivi de l'évolution des événements de risque naturel à partir des contenus de publications sociales, ii) l'introduction d'un traitement flou générant le degré de danger pour chaque événement extrait, iii) Le développement d'un processus efficace pour la classification des citoyens en combinant un regroupement hiérarchique par division et agglomération.
Nous allons implémenter et valider, dans le chapitre suivant, l'approche proposée pour l'analyse et modélisation des réseaux sociaux, à savoir l'analyse des sentiments, la classification hiérarchique des communautés et la détection et le suivi de l'évolution des évènements. Ensuite, nous montrons les résultats des différentes expérimentations que nous avons menées et qui nous ont permis d'évaluer et de valider les contributions introduites.

Chapitre 8

Expérimentation

8.1 Introduction

Dans les chapitres du deuxième partie, nous avons présenté nos contributions d'analyse et modélisation des réseaux sociaux, notamment l'analyse des sentiments(Toujani and Akaichi, 2016, oujani and Akaichi, 2016)(Toujani Radhia and Jalel, 2015, oujani Radhia and Jalel, 2015), la classification hiérarchique des communautés(Toujani and Akaichi, 2019a, oujani and Akaichi, 2019a) ainsi que la détection et le suivi de l'évolution des évènements dans ces réseaux(Toujani and Akaichi, 2019b, oujani and Akaichi, 2019b).
Cependant, nous nous intéressons, dans les sections de ce chapitre, à présenter les principaux résultats obtenus suite à l'application de ces contributions sur des réseaux artificiels et réels. Dans la section 8.2 nous présentons l'évaluation de l'approche d'analyse des sentiments. L'évaluation de la classification hiérarchique mixte fait l'objet de la section 8.3. Dans la section 8.4 nous mettons l'accent sur l'évaluation de l'approche de détection et suivi des évènements : Gestion des catastrophes naturelles.

8.2 Evaluation de l'approche d'analyse des sentiments

Cette section décrit l'évaluation expérimentale de notre approche de classification des sentiments. Premièrement, les données de nos expérimentations sont collectées et analysées et la méthode manuelle de classification des termes sont expliquées. Ensuite, les résultats de classificateur défini sont discutés et comparés à l'aide des mesures de performance.

8.2.1 Extraction des statuts Facebook

Nous avons collecté des données sur Facebook et nous avons concentré sur le contenu du statut contenant une référence à un parti politique. Nous avons choisi une population aléatoire d'utilisateurs Facebook. Cette population comprend des hommes et des femmes tunisiens d'âges et de professions différents. L'extraction des status Facebook consiste à extraire les mises à jour des statuts de chaque utilisateur lors de l'élection tunisienne du 26 octobre 2014 en utilisant l'application I Told You ce qui nous a permis d'obtenir environ 260 statuts.
L'algorithme d'apprentissage supervisé introduit nécessite que les données d'apprentissage soient déjà classées en trois catégories : positives, négatives et floues. Cette classification doit être étiquetée manuellement et avant que l'analyse des sentiments soit effectuée. Cette étape produit un ensemble de données étiqueté POS, si le statut exprime un sentiment positif, et deux autres étiquetés NEG, si le statut exprime un négatif, ou FUZ si le statut exprime une opinion floue.
Pour la comparaison avec d'autres méthodes, nous avons utilisé l'implémentation de Naïve Bayes et nous avons choisi le modèle multivarié de Bernoulli à plusieurs variables de Naïve Bayes. Ainsi, nous avons utilisé la machine à vecteurs de support (SVM), implémenté par le noyau SMO (Sequential Minimal Optimization) qui divise le grand problème d'optimisation quadratique en plus petits problèmes d'optimisation quadratique pouvant être résolus analytiquement (Pang et al., 2002, ang et al., 2002).

8.2.2 Résultats et Discussions

Pour évaluer les performances de la classification de sentiment effectuée, nous avons utilisé les métriques de performance de classification standard utilisées dans les études précédentes de Recherche d'informations et de classification de texte : exactitude, précision, rappel et F-mesure.
Nous comparons les performances du SVM Flou proposé avec celles de la SVM de base. Ainsi, le Tableau8.1 décrit les performances de classification des sentiments en utilisant le SVM de base. En plus, le Tableau8.2 se concentre sur les performances du classificateur flou proposé. Comme indiqué dans le Tableau8.3, nous comparons nos résultats avec ceux obtenus par le classificateur naïve de Bayes.

Sentiment	**Précision**	**Rappel**	**F-measure**
Positive	74	73,1	74
Negative	71,2	72	71,2
Fuzzy	-	-	-

TABLE 8.1: Les performance de SVM de base

Sentiment	**Précision**	**Rappel**	**F-measure**
Positive	88,2	70,2	88,2
Negative	71,2	72	71,2
Fuzzy	63	82,3	63

TABLE 8.2: Les performance de SVM Flou introduit

Sentiment	**Précision**	**Rappel**	**F-measure**
Positive	77	74,2	77
Negative	73	71,4	73
Fuzzy	-	-	-

TABLE 8.3: Les performances de Naïve Bayes

Le SVM proposé se concentre essentiellement sur l'opinion floue qui n'est pas traitée par SVM de base et l'approche naïve de Bayes. En appliquant SVM de base, les publications sociales sont correctement identifiées avec 73,1% des rappels pour la classification des opinions positif. Néanmoins, le classificateur SVM de base surperforme le SVM flou introduit avec 71,2% de rappel pour le classement positif, tandis que le classement positif de l'approche Naïve Bayes est de 70%. La précision des opinions négatives extraites par la méthode introduite est de 71,2%, tandis que la classification négative de naïve Bayes est de l'ordre de 73%.
Par rapport à l'approche Naïve Bayes, la précision du SVM flou indique une haute exactitude et une meilleure qualité de classification (82,3% pour la classification négative). La performance de ces résultats montre la dominance de publications sociales positives par rapport à celles négatives. Lors des élections tunisiennes, les utilisateurs du réseau social expriment leur satisfaction vers les plans politiques proposés. Par conséquent, nous pouvons conclure que la plupart des statuts publiés sur Facebook après la révolution tunisienne reflètent des opinions politiques positives. Cela serait surprenant en raison des conditions difficiles que les utilisateurs ont endurées.

8.3 Evaluation et Optimisation de la classification hiérarchique mixte

Cette section a pour objectif de valider et évaluer la performance des méthodes proposées. L'objectif de nos expériences est d'améliorer la qualité des résultats de notre approche de classification. Les résultats expérimentaux de la méthode de classification hiérarchique

mixte introduite sont comparés à ceux basés uniquement soit sur les techniques de colonie de fourmis ou d'abeilles et aux résultats obtenus en appliquant l'algorithme hiérarchique mixte introduit intégrant les méthodes de l'intelligence des essaims. En outre, nous comparons la technique suggérée avec INFOMAP (Gonzalez-Pardo et al., 2017, onzalez-Pardo et al., 2017) et avec LOUVAIN (Moradi and Rostami, 2015, oradi and Rostami, 2015) incorporant les méthodes de l'intelligence des essaims.

8.3.1 Description du jeu de données

Afin d'obtenir des résultats complets et décisifs, nos expérimentations ont été menées sur deus type de réseaux : réels et artificiels.
Réseaux artificiels : Ce sont des réseaux synthétiques basés sur Lancichinetti Fortunato Radicchi (LFR) benchmark (Fortunato, 2017, ortunato, 2017). Le générateur LFR a plusieurs paramètres, tels que le nombre des noeuds (N), le degré moyen des arrêtes entrants (k), le degré maximal des arrêtes entrants ($maxk$), la fraction entre les arrêtes entrants et sortants à l'intérieur d'une communauté ainsi que (μ), la taille minimale de la communauté ($minc$), et la taille maximale de la communauté ($maxc$).
Réseaux réels : Comme réseaux réels à petite échelle, nous utilisons le club de karaté de Zachary (Zachary, 1977, achary, 1977), le réseau des dauphins de Lusseau (Lusseau et al., 2003, usseau et al., 2003) et le réseau des équipes de football universitaires américaines (Girvan and Newman, 2002, irvan and Newman, 2002). En tant que réseaux à grande échelle, nous utilisons le réseau PGP (Boguná et al., 2004, oguná et al., 2004), le réseau Netscience (Newman, 2006a, ewman, 2006a) et les jeux de données tweeter en collectant de tweets publics publiés pendant une semaine via le framework 140dev Twitter [1] . Notre collection est composée de tweets publiés du 10 au 17 juillet 2017. En fait, nous modélisons les tweets extraits par un réseau pondéré avec le degré de dépendance et de similarité entre les noeuds. En outre, les paramètres des réseaux réels sont décrits dans le tableau8.4 :

Network	number of Nodes	number of Edge
Karate	34	78
Dolphin	62	159
Football	115	613
PGP	10680	24346
Netscience	1589	2742
tweeter	1120	2780

TABLE 8.4: Paramètres des réseaux réels

1. http ://140dev.com/free-twitter-api-source-code-library/

8.3.2 Résultats et Discussions

Nous commençons, dans cette section, par la validation du processus de modélisation du graphe. Puis, nous évaluons, d'une part, la qualité du regroupement pour les réseaux artificiels et celle pour les réseaux réels.

Validation de la modélisation du graphe

Le processus de modélisation graphique introduit, comme poids des arrêtes, une mesure de covariance ainsi qu'une similarité de Jaccard bien fondées entre les noeuds d'un réseau pondéré. En fait, la covariance et la mesure de Jaccard sont coordonnées efficacement. La figure 8.1 présente la courbe de distribution de la covariance et de la similarité de Jaccard basée sur le réseau réel ainsi que le réseau artificiel.

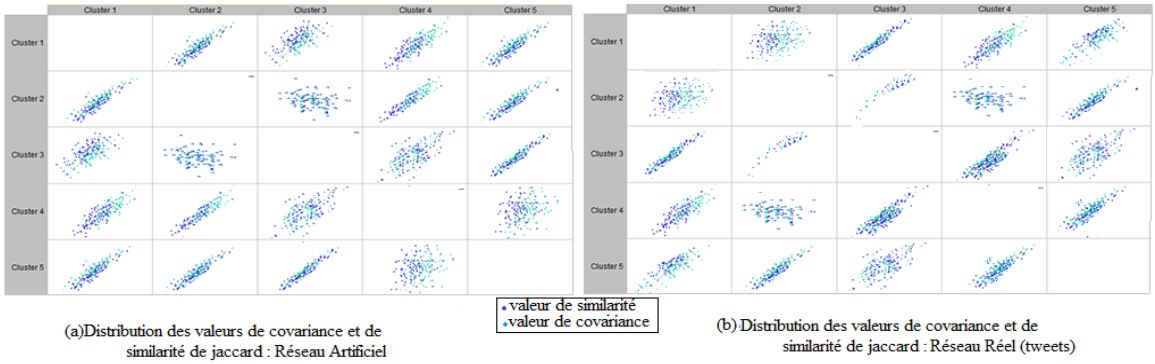


FIGURE 8.1: Distribution des mesures de covariance et de Jaccard

Comme le montre la figure 8.1, les valeurs de covariance et de Jaccard sont similaires. Ainsi, les membres des réseaux sociaux dépendants partagent, dans la plupart des cas, les mêmes opinions. Cette similarité est observée sur le regroupement de LFR benchmark, en tant que base de données artificiel, et dans les données collectées de Tweeter en tant que réseau réel. Par conséquent, la combinaison des valeurs de covariance et de similarité de jaccard montre plus clairement le degré d'importance entre les noeuds.

Qualité du regroupement pour le réseau artificiel

Pour évaluer un partitionnement donné, nous considérons les critères de performance suivants : CEC et NMI.

— **Evaluation du regroupement du réseau artificiel en utilisant la CEC**
Nous définissons la micro-communauté comme suit : $minc = 20$, $maxc = 40$. Nous considérons comme macro-communauté $minc = 40$, $max = 90$.

La figure 8.2 décrit la performance comparative en termes des valeurs de CEC des essaims intégrés dans les méthodes LOUVAIN et INFOMAP par rapport à la performance de notre méthode introduite basée uniquement sur une colonie de fourmis ou d'abeilles ou sur des essaims. Dans les deux cas, (a) représente les petites communautés et (b) correspond aux les grandes communautés.

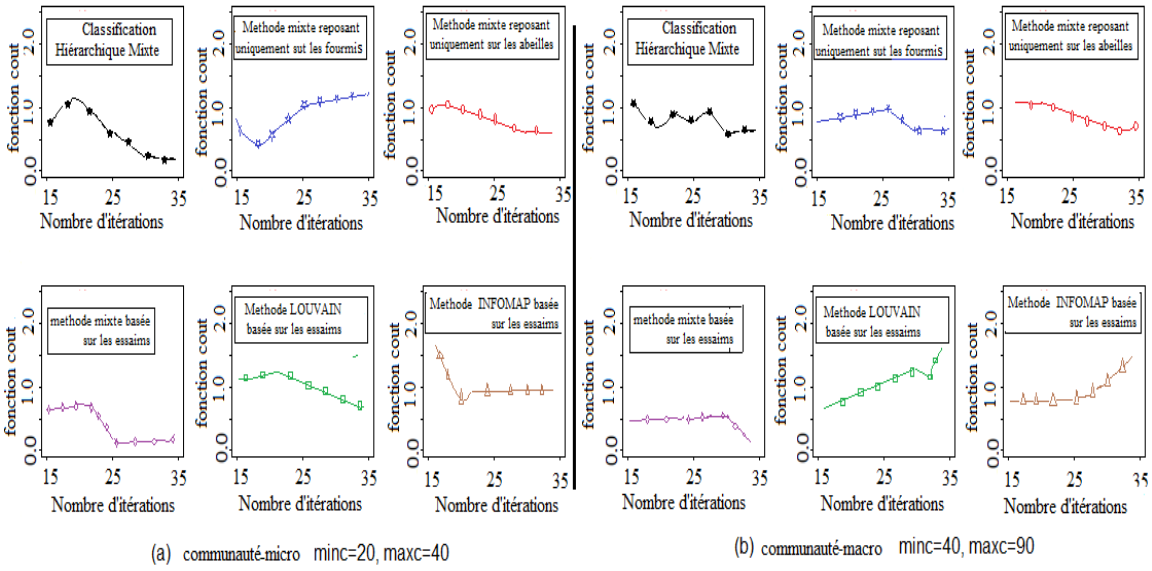


FIGURE 8.2: Comparaison de la qualité du regroupement en termes de CEC pour le réseau artificiel

Comme le montre la figure 8.2, dans le cas de micro-communautés, la fonction coût de notre méthode mixte introduite augmente au début des itérations. Ensuite, elle diminue à partir de l'itération 20, indiquant une meilleure qualité du regroupement. En intégrant uniquement les fourmis dans le processus du regroupement mixte introduit, la fonction d'énergie augmente de manière polynomiale avec l'augmentation du le nombre d'itérations. Cependant, lorsque la colonie d'abeilles est utilisée pour le regroupement mixte, la fonction de coût amoindrit avec la diminution du nombre d'itérations (à l'itération 17, E = 1.01 ; à l'itération 20, E = 0.98 ; à l'itération 30, E = 0.59). Nous remarquons que la méthode hiérarchique mixte introduite est plus performante que celles intégrant uniquement les colonies des abeilles ou des fourmis, d'où la nécessité de combiner les essaims dans la classification introduite. Evidemment, notre technique de classification mixte optimale améliore la qualité du regroupement. Pour les grandes communautés, la fonction coût de la méthode mixte proposée basée sur les essaims est presque constante dans les premières itérations (15 <nombre d'itérations <20, E = 0 :35). Par conséquent, la fonction d'énergie diminue entre les itérations 20 et 25 et reste constante lorsque le nombre d'itérations augmente.
En résumé, le regroupement hiérarchique mixte développé dans ces deux versions (basées sur des essaims ou non) présente une qualité du regroupement supérieure, par rapport au

regroupement mixte de base utilisant uniquement des colonies de fourmis ou d'abeilles.
En outre, comparée aux techniques d'intelligence d'essaims incorporées dans les méthodes INFOMAP et LOUVAIN, notre méthode, reposant sur les essaims, génère la fonction énergétique la plus faible (à l'itération 30, E de la méthode introduite basée sur des essaims = 0,25 ; E de LOUVAIN reposant sur les essaims = 0,95 ; E de INFOMAP basé sur des essaims = 0,99). Dans le cas de micro-communautés, la fonction de coût de notre méthode mixte, dans ces deux versions (à savoir, le regroupement hiérarchique mixte de base et la hiérarchie mixte basés sur les essaims), est inférieure à celle de la fonction de coût des techniques d'intelligence d'essaims incorporées à LOUVAIN et INFOMAP (à l'itération 25, E de la méthode introduite basée sur les essaims = 0,35 ; E de LOUVAIN intégrant les essaims = 0,98 ; E d'INFOMAP basée sur des essaims = 0,75).
Les qualités du regroupement en termes de CEC de la classification hiérarchique mixte introduite, comparées à celles des techniques intégrant uniquement les colonies d'abeille ou de fourmis, sont proches, en particulier dans les itérations complexes (à l'itération 30, E de la méthode de classification de base proposée = 0,53 ; E du regroupement de base utilisant uniquement la colonie de fourmis = 0,53 ; E du regroupement de base employant seulement la colonie d'abeilles = 0,53). Evidement, l'intégration de l'optimisation, dans notre processus de classification hiérarchique mixte, améliore la qualité du regroupement dans les macro et micro-communautés. Ainsi, la méthode introduite génère une fonction coût relativement faible quelque soit le nombre d'itérations, indiquant les meilleurs résultats du regroupement.
Dans la figure 8.3, on compare la distribution, fournie par la fonction CEC, des techniques d'essaim intégrées dans les méthodes LOUVAIN et INFOMAP avec les performances de notre méthode mixte basée uniquement sur des colonies de fourmis ou d'abeilles ou sur des essaims où les partitions introduites, générées par le benchmark LFR, sont regroupées en quatre groupes.

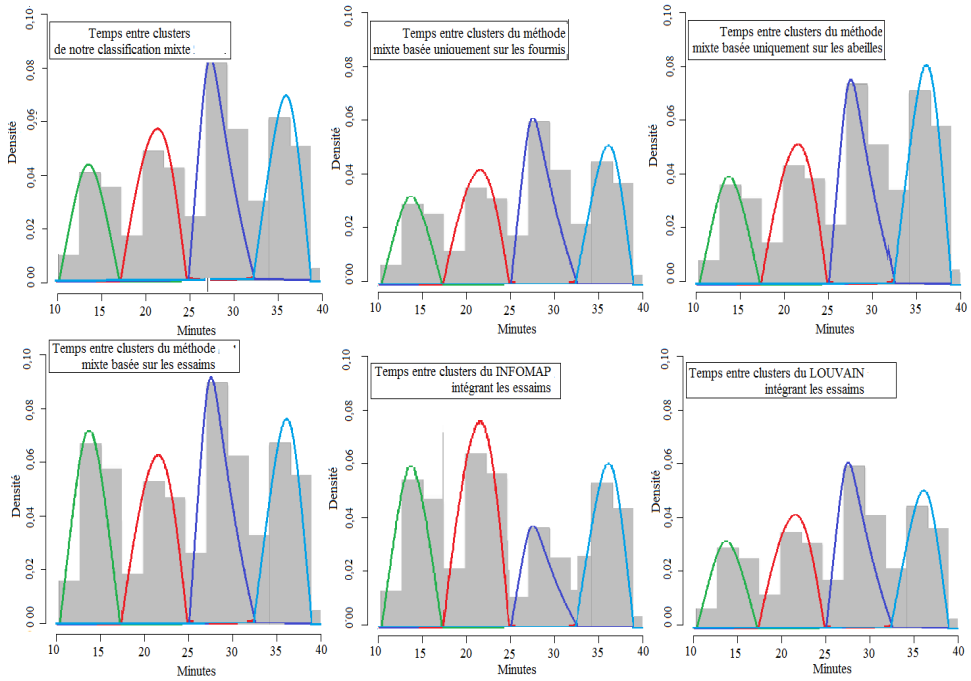


FIGURE 8.3: Histogramme avec approximation de la densité CEC de quatre groupes

En fait, cette distribution est basée sur la densité obtenue pour chaque cluster dans le temps. Nous remarquons que la méthode hiérarchique mixte génère le meilleur partitionnement et détecte les communautés les plus denses dans le plus court temps.
En fait, cette distribution est basée sur la densité obtenue pour chaque cluster dans le temps. Nous remarquons que la méthode hiérarchique mixte génère le meilleur partitionnement et détecte les communautés les plus denses dans le plus court temps. En conséquence, comme susmentionné dans l'histogramme présenté à la figure 8.3 la mesure de CEC place la barrière dans un endroit plus raisonnable. Les groupes produits par notre processus de classification hiérarchique mixte sont plus denses, comparés aux groupes obtenus en appliquant notre classification mixte utilisant uniquement la colonie des abeilles ou des fourmis, à l'exception du quatrième groupe. Dans ce dernier groupe, les partitions fournies par notre méthode mixte employant les abeilles sont les plus denses (avec une densité de classification mixte de base = 0,06 ; celle de la classification basé uniquement sur les fourmis = 0,04 ; la densité de la classification basée seulement sur les abeilles = 0,07). Les techniques d'essaims, incorporées à INFOMAP, génèrent des partitions plus denses que les méthodes d'essaims incorporées à LOUVAIN. La méthode de classification mixte introduite, basée sur des essaims, présente une densité élevée quelque soit le temps de génération des groupes.

— **Evaluation du regroupement du réseau artificiel en utilisant NMI**

La figure 8.4 illustre la performance comparative de la méthode des essaims incorporée dans les algorithmes LOUVAIN et INFOMAP par rapport à celle du regroupement mixte en utilisant uniquement des colonies de fourmis ou d'abeilles, pour les petites et les grandes communautés en termes des valeurs de MNI ainsi qu'en termes des valeurs de paramètre de mixage.

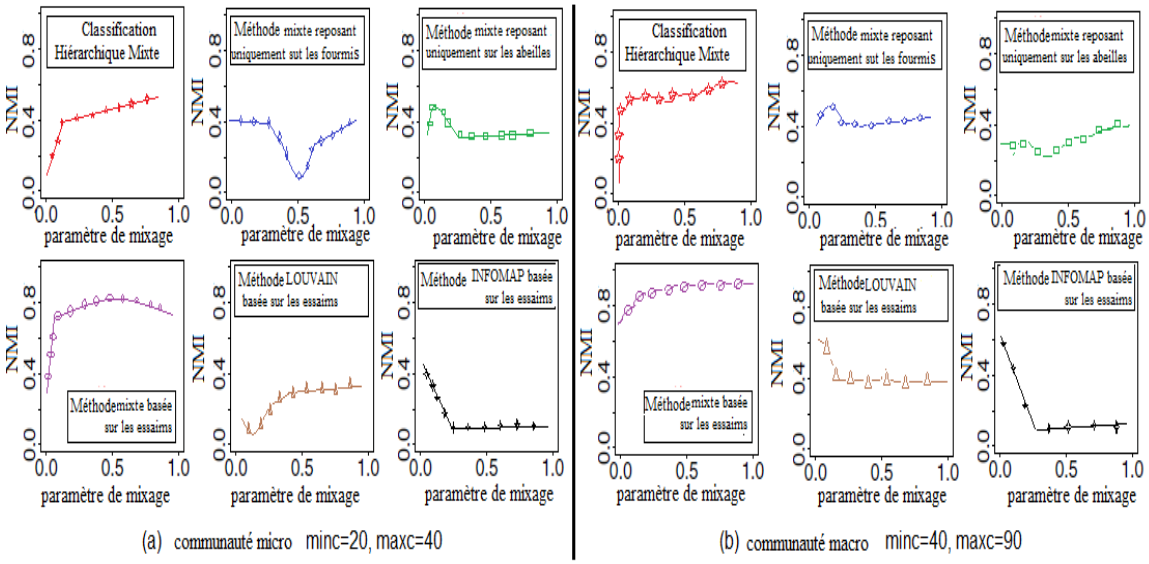


FIGURE 8.4: Comparaison de la qualité du regroupement en termes des valeurs NMI pour le réseau artificiel

Nous remarquons que la classification mixte introduite, basée sur les essaims, est performante pour les petites et les grandes communautés. Par exemple, avec un paramètre de mixage supérieur à 0,5, le regroupement mixte proposé fournit une meilleure qualité du regroupement que LOUVAIN et INFOMAP reposant sur les essaims. En utilisant uniquement des colonies de fourmis ou d'abeilles, notre processus de classification mixte donne une valeur de NMI inférieure à celle fournie par la classification mixte introduite de base, indiquant une faible performance de classification. Dans ce contexte, il est nécessaire d'intégrer l'optimisation dans notre processus de classification. Toutefois, notre méthode mixte offre une meilleure qualité du regroupement pour des valeurs raisonnables des paramètres de mixage dans les macro et micro-communautés. En outre, le nombre des communautés calculé par la méthode mixte introduite est très proche de celui des communautés réelles obtenu par le LFR benchmark. Cela peut s'expliquer par le fait que la méthode mixte développée est basée sur le concept de dépendance et de similarité entre les partitions.

Performance temporelle en appliquant LFR benchmark à grande échelle

Dans la figure 8.5, nous comparons les temps d'exécution spécifiques des macro-communautés du générateur LFR (minc = 40 ; maxc = 90) de la méthode proposée, basée sur les colonies de fourmis, d'abeilles ou les deux essaims, à ceux des techniques LOUVAIN et INFOMAP reposant sur les essaims.

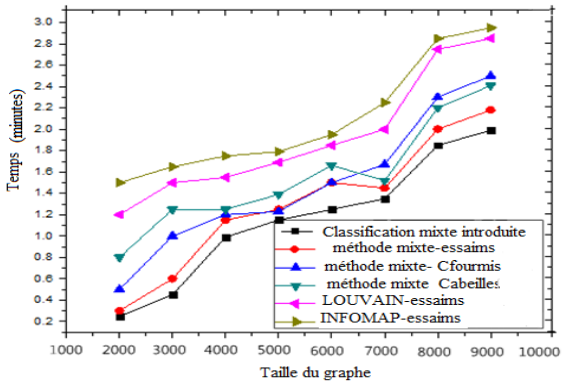


FIGURE 8.5: Comparaison de la qualité du regroupement en termes du temps d'exécution pour un graphe artificiel à grande échelle

Comme mentionné dans la courbe, le temps d'exécution de la classification mixte introduite est assez faible, indiquant les meilleurs résultats du regroupement dans un temps très rapide (1 minute pour un graphe de 4000 noeuds). De plus, le temps d'exécution de notre méthode mixte basée uniquement sur la colonie des fourmis ou d'abeilles ou les deux essaims est supérieur à celui de LOUVAIN et INFOMAP reposant sur les essaims (avec une taille de graphe> 8000, temps d'exécution de classification mixte = 1,4 seconde ; temps d'exécution de classification mixte reposant sur des essaims = 1,6 seconde ; temps d'exécution de classification mixte basée uniquement sur les fourmis = 1,98 seconde ; temps d'exécution de classification mixte basée uniquement sur les abeilles = 1,99 secondes ; temps d'exécution de LOUVAIN basée sur essaims = 2,7 secondes ; temps d'exécution d'INFOMAP basée sur essaims = 3 secondes). Généralement, le gain de temps d'exécution de la méthode mixte introduite est très significatif et augmente polynomialement avec la taille du graphe.

Qualité du Regroupement pour les réseaux réels

Nous prenons en compte les indices Silhouette et DBI en tant que critères de performance.

— **Qualité du Regroupement reposant sur la valeur Silhouette**

La silhouette plot fournit des informations concernant : "cluster", "voisin", et "sil-width". Les figures suivantes affichent le tracé du résultat de l'indice de silhouette du réseau réel utilisé.

La figure 8.6 au dessous présente une comparaison de la performance du réseau de football du collège américain, en termes de l'indice silhouette, de la classification mixte introduite avec celle intégrant uniquement les colonies des fourmis ou d'abeilles, par rapport à la performance des essaims incorporés aux méthodes LOUVAIN et INFOMAP.

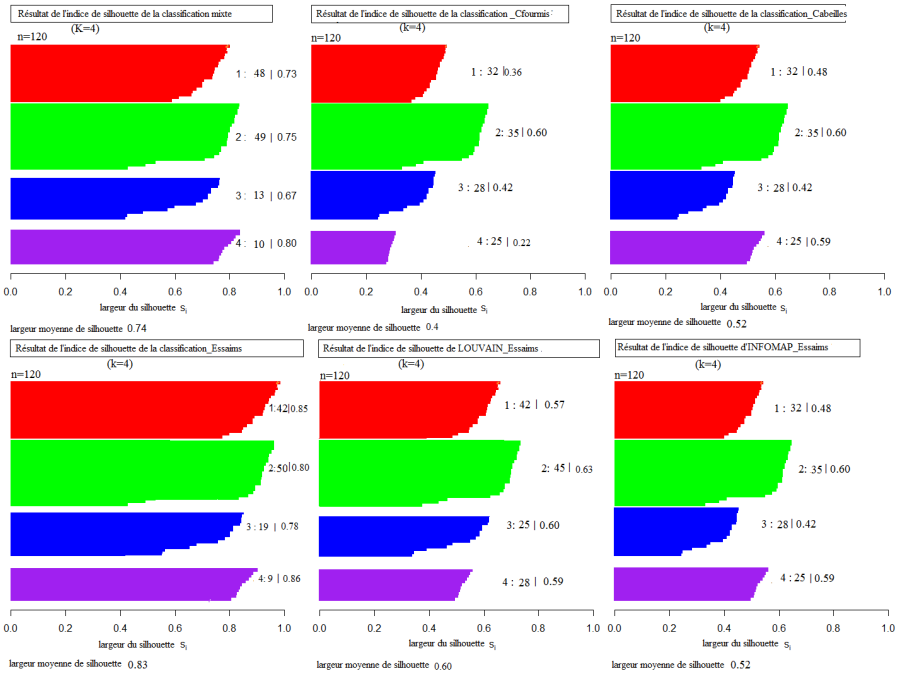


FIGURE 8.6: Comparaison de la qualité du regroupement en termes des valeurs de silhouette pour le réseau du Collège de football American

Comme le montre la figure 8.6, la classification hiérarchique mixte introduite est parfaitement fonctionnée. Sa qualité est généralement supérieure à celle des essaims intégrés dans les algorithmes LOUVAIN et INFOMAP et de la technique de classification de base introduite utilisant uniquement des colonies de fourmis ou d'abeilles. Par conséquent, une structure communautaire forte est détectée en appliquant la méthode introduite basée sur les essaims (avec une largeur moyenne de silhouette = 0,83). La plupart des utilisateurs des réseaux sociaux sont bien classés et appartiennent aux groupes appropriés. Cependant, en se référant à la largeur moyenne de silhouette des méthodes LOUVAIN et INFOMAP reposant sur des essaims (0,60 pour LOUVAIN et 0,52 pour INFOMAP), nous remarquons que ces deux dernières méthodes génèrent une structure communautaire raisonnable. En utilisant seulement les colonies de fourmis, la structure de la communauté devient faible et artificielle (avec une largeur moyenne de silhouette = 0,4). En outre, la largeur moyenne de l'indice silhouette de la méthode mixte employant uniquement des abeilles représente une structure de partition raisonnable (avec une largeur moyenne de silhouette = 0,52).
La figure 8.7 représente une étude comparative de la performance du réseau Dolphin en termes de tracé de la silhouette de la classification hiérarchique mixte avec celles basées uniquement sur les colonies d'abeille ou de fourmis, par rapport à la performance des techniques des essaims incorporées aux méthodes LOUVAIN et INFOMAP. Pour le réseau Dolphin, nous remarquons que les clusters détectés par notre processus de classification ont une bonne structure et chaque sous-partition est bien classée dans la partition appropriée.

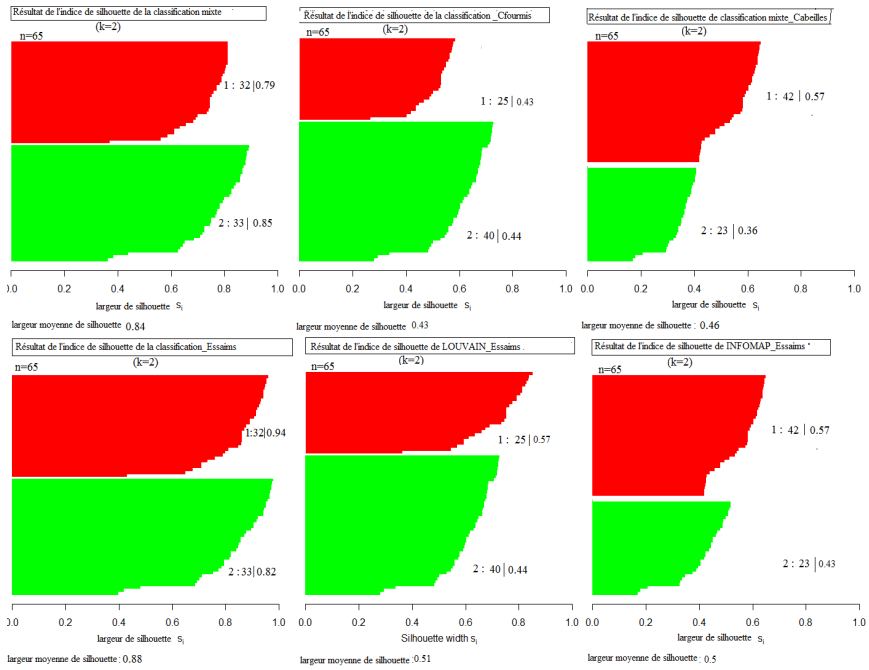


FIGURE 8.7: Comparaison de la qualité du regroupement en termes des valeurs de silhouette pour le réseau Dolphin

Le tracé de la silhouette de la classification hiérarchique mixte introduit reposant sur les essaims pour le réseau Dolphin montre que l'appartenance des membres à ses propres sous-partitions est supérieure à celle des autres partitions. En fait, la valeur de largeur de silhouette de la première sous-partition est presque égale à 1 (0,94), ce qui indique que les membres du réseau social sont bien placés dans les groupes convenables. Néanmoins, le tracé de la silhouette des méthodes INFOMAP et LOUVAIN basé sur les essaims fournit des valeurs de largeur de silhouette correspondant à une structure communautaire raisonnable. La classification hiérarchique mixte dans laquel seules les colonies d'abeilles ou de fourmis sont utilisées, résulte en une largeur de silhouette faible (largeur moyenne de silhouette du regroupement mixte utilisant uniquement les fourmis = 0,5 ; largeur moyenne de silhouette du regroupement mixte employant seulement les abeilles = 0,51) prouvant que les membres du réseau social n'étaient pas placés sur leurs sous-partitions adéquates et pourraient vraiment appartenir à d'autres partitions.
De plus, la figure 8.8 illustre les performances du réseau de clubs de karaté de Zachary, comparées, en termes du résultat de silhouette de la classification hiérarchique mixte introduite, avec celles des méthodes utilisant uniquement les colonies de fourmis ou d'abeilles, par rapport aux performances des techniques d'essaims incorporées avec les méthodes LOUVAIN et INFOMAP.

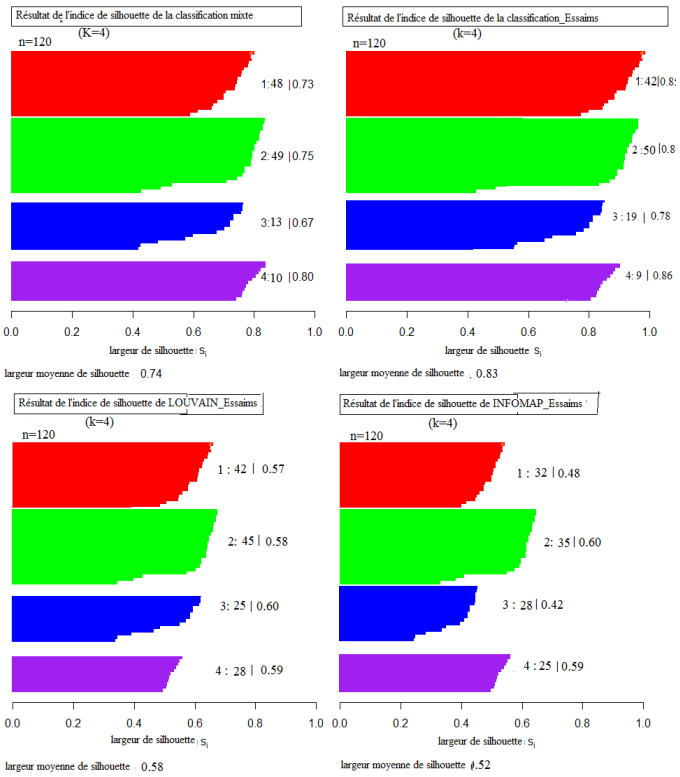


FIGURE 8.8: Comparaison de la qualité du regroupement en termes des valeurs de silhouette pour le réseau du club de karaté de Zachary

Comme indiqué dans la figure 8.8, le résultat de silhouette pour le réseau du club de karaté de Zachary révèle des valeurs proches de 1 pour chaque partition générée de la classification hiérarchique mixte introduite, ce qui montre que l'adéquation entre les membres et les sous-partitions est bonne (avec une largeur de silhouette de 0,80 pour la troisième et la deuxième partitions du groupement mixte proposé fondé sur les essaims). Cependant, le résultat de silhouette des méthodes LOUVAIN et INFOMAP basées sur les essaims pour le réseau club de karaté de Zachary prouve que la structure du regroupement obtenue en utilisant ces méthodes est acceptable (avec une largeur moyenne de silhouette de 0,55 pour les essaims incorporés à la méthode LOUVAIN et 0,53 pour la méthode INFOMAP reposant sur les essaims). Cependant, la structure de communauté obtenue en appliquant uniquement les colonies d'abeilles ou de fourmis est faible (avec une largeur moyenne de silhouette de 0,49 pour le regroupement mixte employant seulement la colonie de fourmis).
Les expérimentations sur les réseaux réels à grande échelle en termes de silhouette sont présentées dans les tableaux 8.5, 8.6 et 8.7.

	Le réseau *PGP*	
	Taille de partition(k)	largeur moyenne de silhouette
classification hiérarchique mixte de base	4	0.71
classification mixte-colonies de fourmis	4	0.69
classification mixte -colonies d'abeilles	4	0.64
classification mixte-les Essaims	4	0.79
LOUVAIN-Essaims	4	0.57
INFOMAP- Essaims	4	0.53

TABLE 8.5: Résultat de silhouette pour le réseau *PGP*

	le réseau Netscience	
	Taille de partition(k)	largeur moyenne de silhouette
classification hiérarchique mixte de base	3	0.82
classification mixte-colonies de fourmis	3	0.55
classification mixte-colonies d'abeilles	3	0.62
classification mixte-Essaims	3	0.89
LOUVAIN-Essaims	3	0.58
INFOMAP-Essaims	3	0.55

TABLE 8.6: Résultat de silhouette pour le réseau Netscience

	Le réseau Tweeter	
	Taille de partition(k)	largeur moyenne de silhouette
classification hiérarchique mixte de base	2	0.75
classification mixte-Colonies de fourmis	2	0.61
classification mixte-colonies d'abeilles	2	0.59
classification mixte- Essaims	2	0.86
LOUVAIN basée-Essaims	2	0.45
INFOMAP-Essaims	2	0.48

TABLE 8.7: Résultat de silhouette pour le réseau Tweeter

Nous remarquons que, pour tous les réseaux réels à grande échelle, la méthode de regroupement hiérarchique mixte introduite fournit une structure forte indiquant la meilleure adéquation entre les membres et ses clusters appropriés (largeur moyenne de silhouette du regroupement de base pour le réseau PGP = 0,71 ; largeur moyenne de silhouette de la classification introduite basée sur des essaims pour le réseau Tweeter = 0,86).
La méthode du regroupement hiérarchique suggérée utilisant uniquement les colonies de fourmis ou d'abeilles détecte une structure de communautaire raisonnable (largeur moyenne de silhouette du regroupement mixte utilisant uniquement une colonie de fourmis pour le réseau Netscience = 0,55 ; largeur moyenne de silhouette du regroupement mixte employant seulement la colonie d'abeilles pour le réseau Tweeter = 0,59). Pour tous les réseaux à grande échelle, les méthodes LOUVAIN et INFOMAP reposant sur les essaims génèrent une structure acceptable, à l'exception du réseau Tweeter où la structure de la communauté obtenue en appliquant ces techniques est faible (avec la largeur moyenne de silhouette des essaims incorporés dans LOUVAIN = 0,45).

— **Qualité du Regroupement basé sur la valeur de** DBI

La figure 8.9 illustre les performances comparatives, en termes de DBI de la classification hiérarchique mixte introduite avec celles obtenues en utilisant uniquement les colonies d'abeilles ou de fourmis, par rapport à la performance des méthodes LOUVAIN et INFOMAP reposant sur les essaims par rapport à des différentes tailles de partitions.

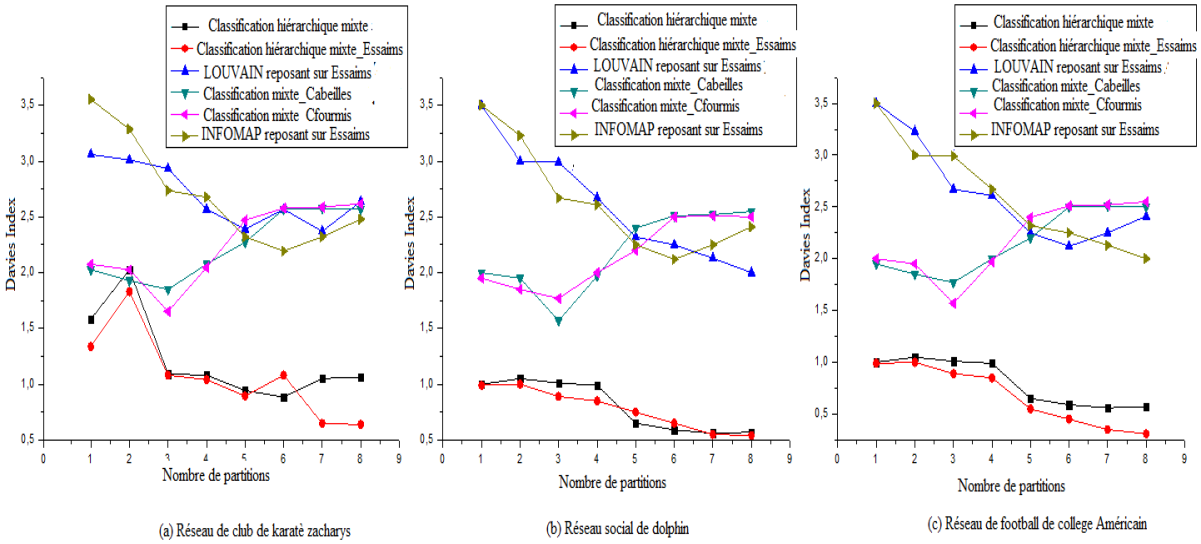


FIGURE 8.9: Comparaison de la qualité du regroupement en termes des valeurs DBI pour les réseaux à petite échelle

La figure 8.9 décrit l'évolution de DBI pour les réseaux réels de petite taille en fonction du nombre des clusters. Nous remarquons, à partir de cette figure, que la méthode de classification hiérarchique mixte développée a le meilleur score DBI et produit des groupes avec une similarité élevée au sein du même groupe et une similarité faible entre les groupes. Pour le réseau de karaté et avec une taille de partitions > 6, la classification mixte introduite et celle basée sur les essaims ont le DBI le plus bas (avec DBI de classification mixte introduite = 1,02 ; DBI de classification mixte reposant sur les essaims = 0,52 ; DBI de la classification mixte reposant uniquement sur la colonie de fourmis = DBI de la classification mixte utilisant seulement la colonie d'abeilles = 2,52 ; DBI des essaims incorporés à LOUVAIN = DBI des essaims incorporés à INFOMAP = 2.12).
Nous comparons maintenant les réseaux de dauphins et de football. Nous constatons que la méthode suggérée, basée sur l'optimisation, possède une qualité de classification supérieure à celle des autres techniques. Pour une taille de cluster> 4, la différence est négligeable entre la classification hiérarchique mixte de base et celle utilisant uniquement la colonie de fourmis ou la colonie d'abeilles, les essaims incorporés à INFOMAP et les essaims incorporés à LOUVAIN. Nous constatons que notre méthode de classification génère, pour les réseaux à petite échelle, une mesure de DBI assez faible quelle que soit la taille de la groupe, indiquant les meilleurs regroupements.
La figure 8.10 présente les performances comparées en termes de DBI pour les réseaux à grande échelle par rapport aux différentes tailles de cluster.

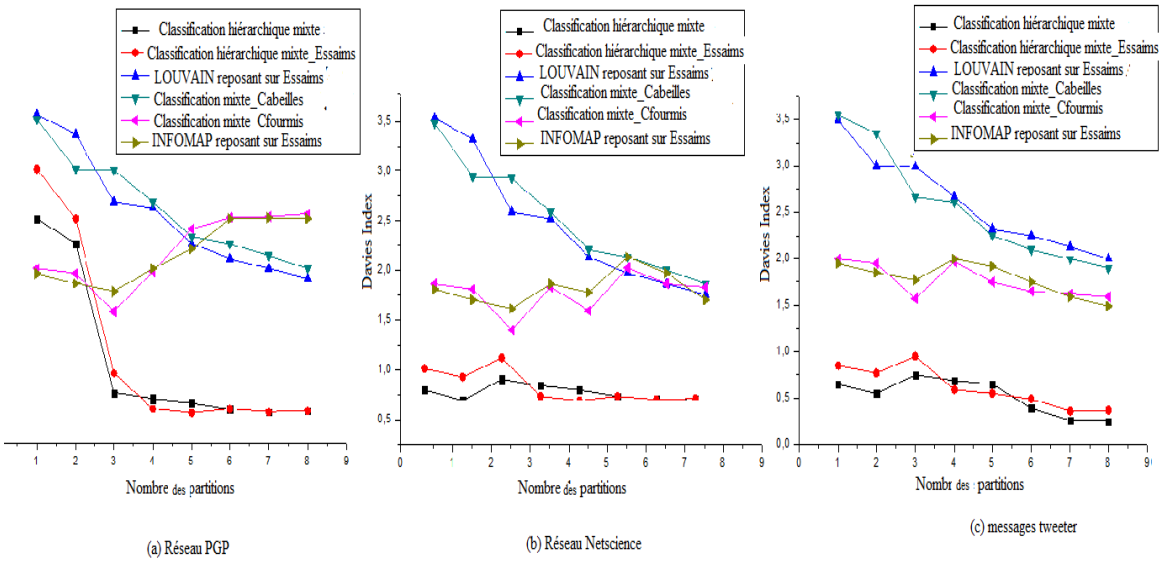
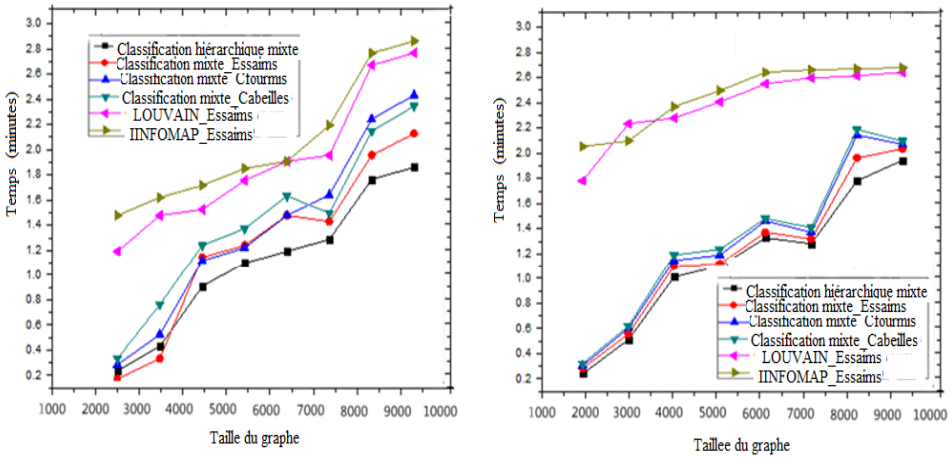


FIGURE 8.10: Comparaison de la qualité du regroupement en termes des valeurs d'indice de Davies pour les réseaux à grande échelle

Comme illustré dans la figure 8.10, la méthode de classification mixte introduite est la meilleure en termes de la moyenne de similarité entre chaque partitions et celles qui sont plus semblables car elle produit, pour les réseaux à grande échelle, une collection de groupes avec la plus petite valeur DBI (avec un score DBI proche de 0,5 pour la classification mixte de base et celle basée sur des essaims pour le réseau PGP et le réseau Netscience).
Pour le réseau Tweeter, les performances de toutes les techniques sont élevées. La méthode introduite, dans ses deux versions, a la valeur DBI la plus basse et fournit des clusters compacts et séparés. Les performances de la classification mixte utilisant uniquement la colonie de fourmis et INFOMAP reposant sur les Essaims sont proches (avec DBI pour la classification mixte employant seulement la colonie de fourmis = 1,75 et DBI pour INFOMAP basée sur les Essaims = 1,76, pour une taille de grappe = 6). De même, pour le réseau Tweeter, la différence entre la qualité du regroupement de la méthode mixte proposée appliquant uniquement la colonie d'abeilles et celle d'essaims incorporés à LOUVAIN, est négligeable. Les valeurs de DBI de la méthode mixte utilisant seulement la colonie d'abeilles et les essaims incorporés à LOUVAIN augmentent de manière polynomiale avec la taille du groupe, mais présente les performances les plus défavorables, comparées à celles des autres méthodes. Nous remarquons que notre approche fonctionne presque parfaitement (avec DBI> 1), en particulier lorsque la taille des partitions augmente.

— **Performances temporelles des réseaux réels PGP et de NetScience**

Les figures 8.11 (a) and 8.11 (b) décrivent respectivement les temps d'exécution spécifiques des réseaux NetSience et PGP.



(a) Comparaison Temporelle pour le réseau NetScience

(b) Comparison temporelle pour le réseau PGP

FIGURE 8.11: Comparaison de la qualité du regroupement en termes de temps d'exécution pour les graphes réels à grande échelle

Nous constatons que le gain de temps d'exécution de la classification mixte introduite est très significatif, à l'exception de la taille de graphe ≤ 4000 pour le la classification mixte reposant sur les essaims. Pour le réseau PGP, la différence de temps d'exécution de la méthode mixte basée uniquement sur les fourmis, sur la colonie d'abeilles ou les deux essaims est négligeable. Cependant, LOUVAIN et INFOMAP basées sur les essaims ont le temps d'exécution le plus long, indiquant la pire complexité temporelle. Généralement, la méthode développée a un temps d'exécution assez court, représentant les meilleurs résultats du regroupement dans un temps plus rapide.

8.4 Evaluation de l'approche de détection et suivi des évènements : Gestion des catastrophes naturelles

Dans cette section, nous validons et évaluons la performance de la méthode proposée de détection et de suivi des évènements pour la gestion des catastrophes naturelles dans les réseaux sociaux.

8.4.1 Description du jeu de données

Afin d'obtenir des résultats complets et décisifs, l'évaluation a été menée sur des réseaux réels et artificiels.

Réseau réel : pour évaluer les performances de la méthode de détection des événements, nous avons utilisé la base de données CrisisLexT26 (Olteanu et al., 2014, lteanu et al., 2014), y compris les tweets collectés au cours de 26 événements de crise survenus en 2012 et 2013. Chaque crise contient environ 1 000 tweets annotés pour un total d'environ 28 000 tweets avec des étiquettes indiquant si un tweet est lié à un événement de crise ou non. Nous avons trouvé plus d'informations sur la base de données CrisisLexT26 sur le site Web CrisisLex [2].
Réseau artificiel : Nous avons employé le benchmark SNOW (Social News On the Web en anglais) de 2014, qui a permis de créer un benchmark public et d'évaluer une ressource pour le problème de la détection des sujets dans des flux du contenu social (Papadopoulos et al., 2014, apadopoulos et al., 2014). Ce repère a été utilisé pour extraire des histoires ou des sujets d'actualité, pour plusieurs intervalles de temps supérieurs à 24 heures, où chaque intervalle de temps dure 15 minutes.

8.4.2 Résultats et Discussions

L'efficacité de la méthode proposée a été validée en la comparant à celles des classificateurs Naïves Bayes et de la machine à vecteurs de support (Support Vector Machine (SVM)). En plus, comme l'objectif principal de notre travail est d'analyser les tweets liés à un événement de danger naturel, les performances de la méthode introduite sont comparées à celles des techniques proposées par Verma et al. (Verma et al., 2011, erma et al., 2011). Le choix d'une mesure d'évaluation appropriée dépend des propriétés du cluster. Par conséquent, pour représenter la distribution homogène des groupes, nous avons employé la mesure CEC (Hajto et al., 2017, ajto et al., 2017) et celle de précision-rappel comme critères d'évaluation. En outre, pour prouver que les groupes détectés sont compacts et bien séparés, nous avons utilisés l'indice DBI et celui de Silhouette (Moshtaghi et al., 2018, oshtaghi et al., 2018).
Nous avons appliqué CEC et Precision-Recall comme critères externes de qualité du regroupement pour le benchmark SNOW, car ce dernier dispose d'une partition modèle et d'un jeu de données étiquetées. Toutefois, lorsque les mesures de qualité sont basées sur des exemples, nous avons appliqué l'indice de la silhouette et DBI comme critères internes pour la base de données CrisisLexT26.

— **Qualité de regroupement basée sur le Rappel-Précision pour le réseau artificiel**

Dans la figure 8.12, nous comparons les performances (en termes de précision / rappel) des classificateurs SVM, de Naïves bayes et de la méthode Verma avec celle de notre technique de détection des événements et du suivi de la structure de communauté des citoyens.

2. CrisisLex T26 Dataset, http ://www.crisislex.org/data-collections.html#CrisisLexT26.

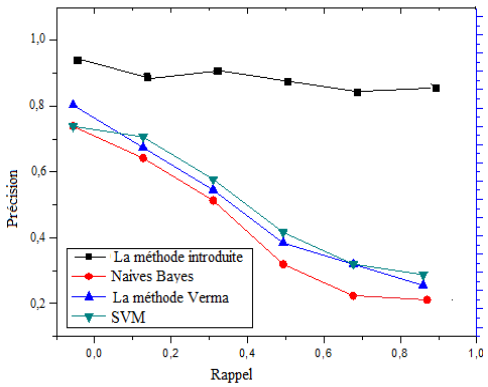


FIGURE 8.12: Comparaison de la qualité du regroupement en termes de Précision-Rappel

D'après les courbes présentées ci-dessus, nous remarquons que notre méthode de détection et du suivi des événements permet d'identifier avec succès la communauté des citoyens en danger. Il est aussi clair que notre approche fonctionne parfaitement (avec une précision> 0,89) et que sa qualité est généralement meilleure que celles des classificateurs SVM, Naïves Bayes et Verma en raison de la grande efficacité du regroupement hiérarchique mixte introduit. En fait, comme le processus mixte défini dans (Toujani and Akaichi, 2019a, oujani and Akaichi, 2019a) permet d'ajuster la partition des groupes, le processus de classification hiérarchique des citoyens est plus flexible.
Le classificateur Bayes a un taux de rappel élevé avec une précision de 0,22 et un rappel de 0,89), ce qui indique une pire performance de classification. En outre, la méthode Verma et la classification du SVM fournissent des classes de citoyens de danger non-équilibrées pour identifier les tweets liés à une crise avec une précision = 0,35 et un rappel = 0,70. Généralement, la performance de ces approches diminue de manière significative lors de l'identification des groupes des citoyens en danger. Toutefois, la méthode introduite de regroupement mixte des citoyens représente le meilleur modèle pour identifier les communautés des citoyens en danger et les tweets liés aux crises. Ainsi, la classification des citoyens en danger génère des classes de citoyens équilibrées et fournit un équilibre entre précision et rappel avec une précision supérieure à des valeurs de rappel plus élevées (> 80).

— **Qualité de regroupement basée sur le *CEC* pour le réseau artificiel**

Dans ce travail, nous utilisons le CEC comme critère de densité basé sur l'entropie pour vérifier l'efficacité du hiérarchique mixte des citoyens et pour trouver le nombre optimal des groupes en supprimant automatiquement les groupes des citoyens dont le coût de l'information est négatif. La figure 8.13 présente une comparaison des performances des classificateurs SVM, Naive Bayes et de la méthode Verma (en termes des valeurs de CEC), avec celle de notre technique de détection et du suivi des communautés des citoyens.

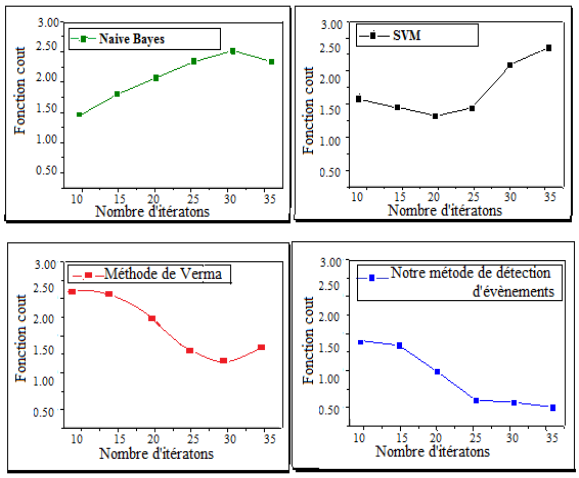


FIGURE 8.13: Comparaison de la qualité du regroupement en termes de valeur CEC

Comme le montre la figure 8.13, la fonction du coût de notre méthode de détection des évènement et de classification des citoyens est inférieure à celles des autres techniques. En fait, sa fonction d'énergie augmente au début des itérations. Ensuite, elle diminue à partir de l'itération 15, indiquant une meilleure qualité de détection d'évènement. Néanmoins, la valeur de la fonction d'énergie du classificateur Naïve Bayes accroît proportionnellement avec le nombre d'itérations, montrant la pire qualité d'identification des groupes des citoyens en danger. Pour le classificateur SVM, la fonction énergie diminue de manière significative pour détecter les événements de danger (avec une fonction de coût égale à 1,25 à l'itération 20). Lorsque le nombre des itérations est supérieur à 20, la valeur de la fonction d'énergie amplifie, ce qui reflète des pires performances. Nous remarquons également que la fonction du coût de la méthode de Verma se dégrade considérablement avec le nombre d'itérations. Mais, elle est supérieure à la fonction d'énergie des autres méthodes, révélant la pire qualité de performance. Généralement, notre méthode de détection d'événement génère une fonction de coût assez faible quelque soit le nombre des itérations, indiquant des meilleurs résultats du regroupement des citoyens en danger. En effet, notre technique de détection d'événement et d'identification des citoyens en danger est presque parfaite en termes d'homogénéité et sa qualité est plus meilleure que celles des autres méthodes, notamment avec l'augmentation du nombre d'itérations (avec une fonction de coût = 1,0 et un nombre d'itérations compris entre 20 et 30 itérations).
La figure 8.14 illustre la répartition, fournie par la fonction CEC, de la méthode d'identification des citoyens en danger par rapport à celle obtenue par les classificateurs Naïve Bayes, SVM et Verma lors de l'identification de quatre groupes de citoyens en danger.

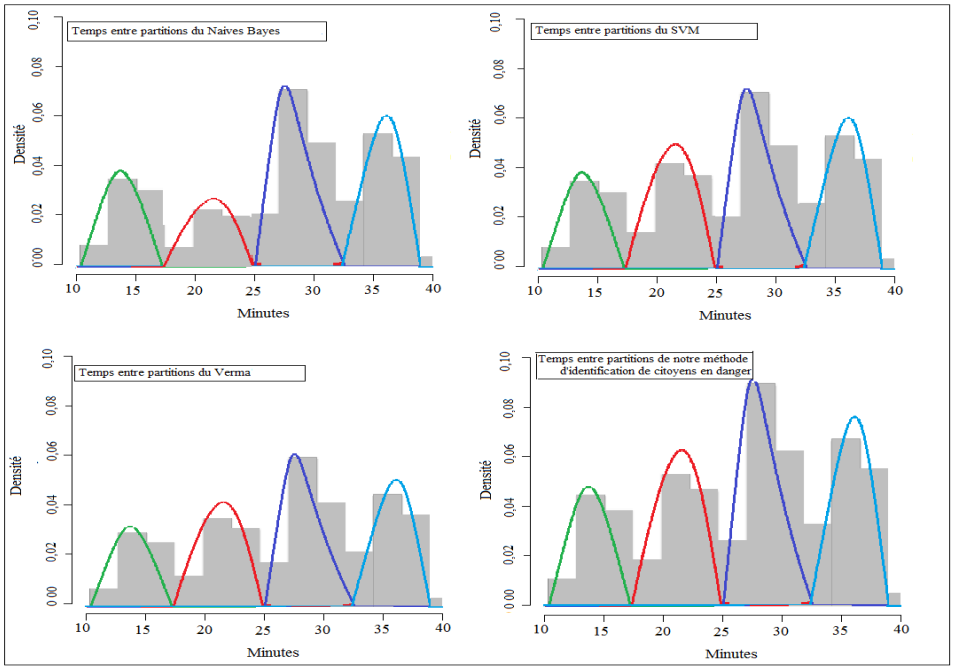


FIGURE 8.14: Histogramme avec approximation de la densité CEC fournie pour quatre groupes sur divers plages horaires.

Comme montré dans la figure 8.14, la répartition des groupes des citoyens est basée sur la densité obtenue pour chaque partition au cours d'une période. En comparant la méthode proposée avec Verma et les classificateurs SVM et Naives Bayes, nous constatons que notre méthode présente une meilleure identification des citoyens en danger et produit des communautés plus denses dans un temps plus court. En conséquence, la CEC place la barrière dans un endroit plus raisonnable. Par exemple, le troisième groupe obtenu par notre technique contient les meilleures partitions denses avec une valeur de densité égale à 0,08. La différence entre la distribution des tweets des citoyens par tranche de temps des classificateurs Naïve Bayes et SVM est négligeable, à l'exception du deuxième groupe où le Naïve Bayes fournit les groupes des citoyens les plus denses. En outre, la densité de la méthode Verma est faible quelque soit l'intervalle du temps.

— **Qualité du regroupement basée sur l'indice de silhouette pour le réseau réel**

La figure 8.15 révèle le tracé de la silhouette résultant du réseau réel utilisé.

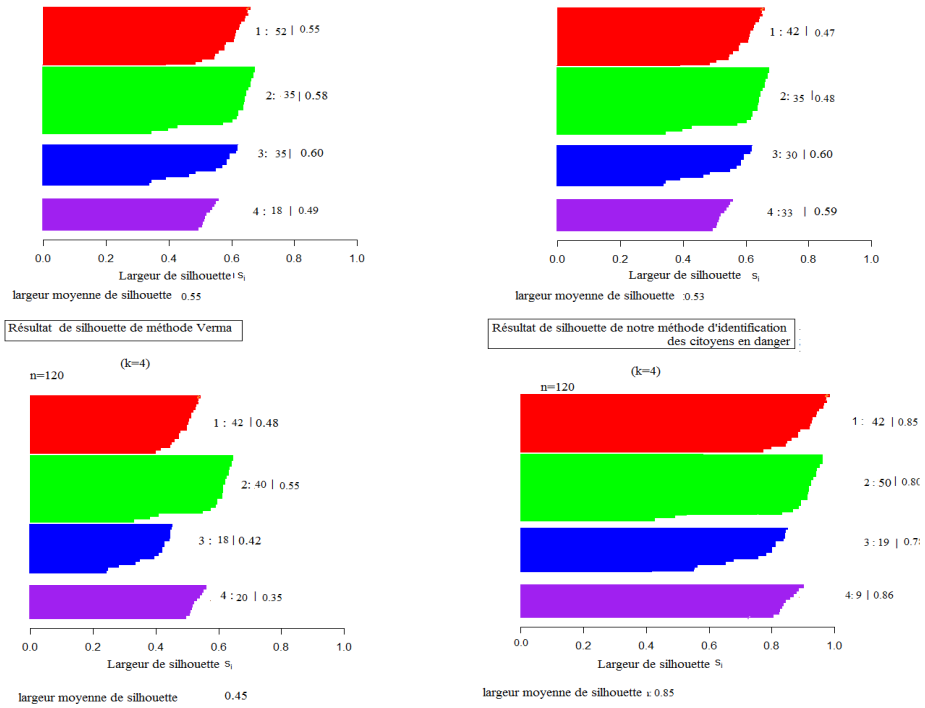


FIGURE 8.15: Comparaison de la qualité du regroupement en termes des valeurs de silhouette

D'après la figure 8.15, notre méthode d'identification des citoyens fonctionne parfaitement et surpasse généralement la qualité des autres techniques. De toute évidence, l'appartenance des citoyens à ses propres sous-partitions est supérieure à celle aux autres partitions, ce qui indique que les citoyens du réseau social, présentant le même danger, sont bien placés dans leurs communautés détectées. De plus, une structure solide des groupes des citoyens est formée en appliquant la méthode introduite de classification des citoyens avec une largeur de silhouette moyenne = 0,83. En outre, la plupart des sous-partitions sont bien classées et appartiennent à la partition appropriée. Cependant, en se référant à la largeur de silhouette moyenne des classificateurs Naïve Bayes et SVM (0,53 pour Naïve Bayes et 0,52 pour SVM), nous constatons l'existence d'une structure communautaire raisonnable des citoyens. La précision entre les citoyens et les sous-partitions utilisant la technique Verma est faible (largeur de la silhouette = 0,45), ce qui révèle que les citoyens des réseaux sociaux ne sont pas placés dans leur groupe adéquate et peuvent appartenir à d'autres groupes.

— **Qualité du regroupement basée sur l'indice Davies Bouldin pour le réseau réel**

Dans la figure 8.16, nous comparons les performance (en termes d'indice de Davies-Bouldin) des approches SVM, Naive Bayes et Verma à celle de notre méthode d'identification des citoyens par rapport aux différentes tailles du groupe.

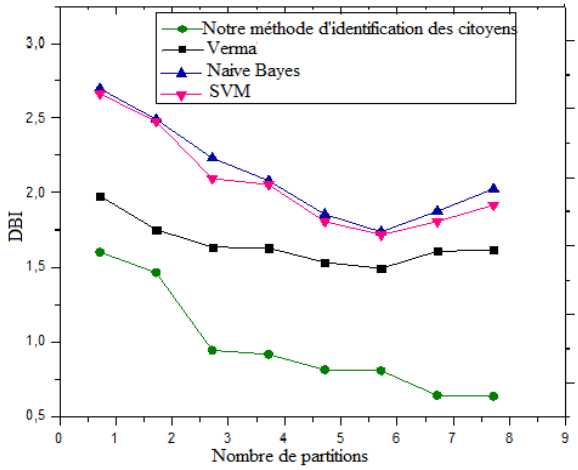


FIGURE 8.16: Comparaison de la qualité du regroupement en termes des valeurs de l'indice Davies

Considérant la figure 8.16, qui met l'accent sur l'index de Davies Bouldin pour le jeu de données CrisisLexT26, la méthode de détection de citoyens en danger génère une meilleure structure (avec DBI = 0,22 et le nombre de cluster = 8). En outre, notre technique fournit la meilleure qualité du regroupement, comparée aux autres approches. Nous observons également que la différence entre la qualité du regroupement de SVM et celle du classificateur Naïves bayes est négligeable, indiquant les plus mauvaises performances. Par ailleurs, les partitions des citoyens fournies par la méthode Verma sont mal séparées. Généralement, la méthode introduite produit des partitions avec une faible distance intra-groupe et une grande distance inter-groupe. Elle fournit aussi les valeurs d'indice de Davies les plus basses et des partitions avec une similarité élevée au sein d'un groupe et une similarité faible entre les groupes.

8.4.3 Visualisation

Les résultats fournis par la technique du regroupement des citoyens pour la base de données CrisisLex T26 ont été visualisés en explorant l'outil Power BI à pour l'analyse visuelle de la répartition des citoyens en danger. La visualisation est efficace pour annoncer les actualités et transmettre rapidement des nouvelles informations, telles que l'emplacement d'une crise et le nombre de victimes. Ainsi, un analyseur peut approfondir un sujet et offrir une nouvelle perspective pour aider le décideur à voir l'événement d'une manière totalement nouvelle.
Comme le montre la figure8.17, les cartes Power BI permettent aux utilisateurs des réseaux sociaux d'analyser les données à l'aide d'un tableau de bord simplement généré, qui indique les zones les plus infectées et fournit une sensibilisation puissante pouvant faciliter les opérations d'urgence.

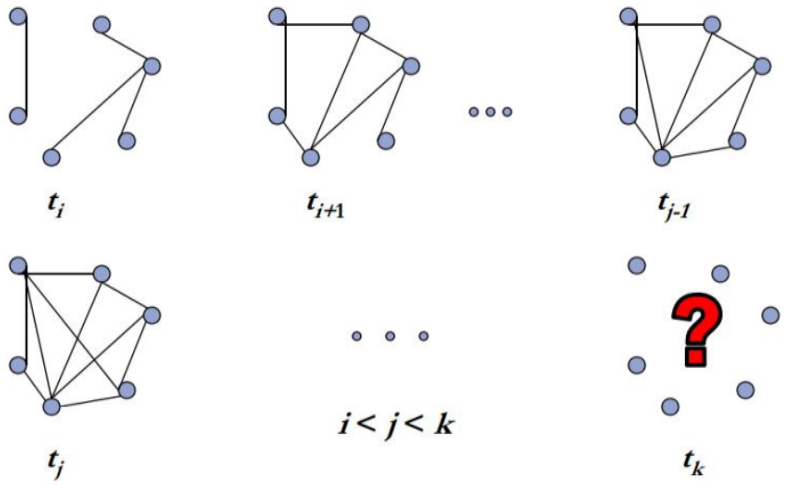


FIGURE 8.17: Visualisation des groupes de citoyens à l'aide des cartes Power BI

En fait, les cartes «3D» des systèmes de gestion des catastrophes peuvent accélérer l'évaluation et l'analyse. Ces systèmes contiennent une quantité considérable d'informations détaillées sur l'emplacement et le degré de danger des citoyens par région.

8.5 Conclusion

Dans ce chapitre, nous avons présenté l'expérimentation de l'approche proposé d'analyse et de modélisation des réseaux sociaux. Cette approche est constituée des contributions suivantes : la classification hiérarchique mixte, l'intégration des métaheuristiques dans notre processus de classification, d'analyse des évènements dans les réseaux sociaux. Afin d'évaluer les performances de nos contributions, nous avons utilisé des indices externes et internes à savoir CEC, rappel-précision, silhouette et DBI.

Conclusion générale et perspectives

L'explosion des réseaux sociaux a rendu indispensable leur analyse et leur exploration, notamment pour la détection des communautés. Dans le cadre de cette thèse, nous présentons une nouvelle approche pour la détection de communautés dans les réseaux sociaux dont la principale originalité est de regrouper dans un même groupe les utilisateurs ayant les mêmes opinions. D'où la nécessité de développer, dans un premier temps, une technique d'analyse des sentiments et une approche de détection des communautés partageant les mêmes opinions, dans un second temps. Comme la taille massive des sentiments exprimés dans les réseaux sociaux pose un sérieux défi pour l'évolutivité des algorithmes de classification classique des graphes et d'évaluation des communautés découvertes, nous avons introduit, dans cette thèse, une nouvelle approche pour découvrir la structure communautaire hiérarchique dans les grands réseaux. La classification hiérarchique hybride proposée combine les avantages des techniques de classification hiérarchique ascendante et descendante. En fait, le premier type de classification est efficace pour identifier les petits groupes, tandis que le deuxième type de classification fonctionne parfaitement à grand échelle.

Notre technique de classification hiérarchique mixte est basée sur l'hypothèse qu'il existe une solution initiale composée de k partitions et sur la combinaison des deux opérateurs de classification hiérarchique. Pour éviter la génération aléatoire de la partition initiale de notre algorithme mixte, nous avons défini une heuristique qui vise à modéliser le problème de détection de communautés sous forme d'un problème d'optimisation, à savoir un problème de sac à dos multi-objectifs. En outre, pour analyser ce problème, nous avons suggéré une méthode de recherche Tabou.

À la fin du processus de regroupement hybride introduit, un point fixe, représentant un optimum local de fonction de coût qui mesure le degré d'importance entre deux partitions, a été obtenu. Par conséquent, notre modèle combiné a conduit à l'émergence d'une structure communautaire locale. Pour éviter cet optimum local et assurer la convergence des structures de communauté vers l'optimum global de la fonction de coût, la détection

des structures de communauté, dans cette thèse, n'a pas été considérée uniquement comme un problème de classification, mais aussi comme un problème d'optimisation. D'une part, nous avons proposé une modélisons génétique pour notre classification hiérarchique mixte. Nous avons intégré aussi les techniques d'intelligence en Essaims dans notre processus de regroupement hiérarchique.

Nous avons amélioré la modularité de Newman en ajoutant le degré de dépendance et similarité entre les membres des réseaux sociaux et définissant (QDS). En combinaison avec les techniques d'optimisation et la classification hiérarchique mixte, QDS a permis d'améliorer la précision des graphes, même les plus denses et les plus confus.

Nous avons déduit le niveau optimal de modularité, ce qui a minimisé le temps requis pour améliorer globalement le processus hiérarchique mixte suggéré. En outre, notre modèle hiérarchique mixte a donné de bons résultats pour les communautés de petites et grandes tailles. Le regroupement hiérarchique mixte introduit a fourni une fonction d'énergie (E) basse des partitions du benchmark LFR pour les micro et macro-communautés, indiquant une bonne qualité de regroupement. De plus, comme le processus mixte mis en place a permis d'ajuster la répartition des groupes, le processus de classification hiérarchique a été plus flexible pour les petites et les grandes communautés. En effet, la combinaison alternative de l'agrégation et de la décomposition a réduit la fonction de coût de la classification. De plus, les mesures de validation, à savoir la CEC et le NMI, appliquées au benchmark LFR, ont montré une forte dépendance entre le regroupement mixte et la classification manuelle. L'objectif principal de notre classification hiérarchique mixte est de fournir des partitions cohésives et séparées.

Pour atteindre cet objectif, nous nous sommes référés aux critères internes appliqués pour mesurer la qualité d'une structure de regroupement sans référence à des informations externes. Dans ce travail de recherche, nous avons utilisé les indices de silhouette et de DBI pour évaluer les partitions de la méthode introduite basée sur des réseaux à petite et grande échelle. Les valeurs obtenues de DBI ont prouvé que la méthode hiérarchique mixte est capable de fournir des partitions ayant des opinions très proches avec des membres compacts. De plus, la largeur moyenne de silhouette pour un groupe indique que la méthode introduite génère des partitions bien séparées et cohésives.

De toute évidence, les résultats obtenus ont montré une bonne qualité de regroupement de la méthode mixte proposée pour les deux indices d'évaluation, à savoir les critères externes (critères CEC et NMI) et les critères internes (silhouette et DBI).

La détection des communautés hiérarchiques peut être efficace pour connaître les caractéristiques cachées et avoir une idée claire de la structure du réseau. Néanmoins, pour accomplir cette tâche, nous avons traité les problèmes suivants. Premièrement, il est essentiel non seulement de déterminer les communautés, mais également d'identifier les modifications apportées à la structure du réseau au fil du temps. Bref, les réseaux sociaux changent régulièrement en raison des changements suivants : création des nouveaux liens ou disparition des nouveaux liens existants, formation des nouvelles communautés, etc. A ce propos, nous mettrons l'accent, dans notre contribution suivante, sur l'analyse de la mobilité des communautés et des évènements.

En fait, les réseaux sociaux semblent être une source riche pour découvrir la mobilité des événements et analyser leurs tendances. Dans cette étude, la méthode de l'arbre de décision a été discutée pour analyser la mobilité d'événement dans un réseau social. En fait, le modèle de mobilité proposé a été conçu pour décrire le mouvement des opinions et montrer ses évolutions au fil du temps. Par conséquent, la mesure d'entropie développée a mis en évidence les attributs spatiaux et temporels associés à chaque attribut d'opinions détectées. Par une analyse expérimentale approfondie, nos résultats montrent que la mobilité des opinions dépend fortement du comportement à la fois temporel et spatial. De plus, les résultats préliminaires révèlent que notre modèle de classification de la mobilité a le potentiel de détecter et de sensibiliser les événements. En outre, Les points abordés dans notre contribution d'analyse des évènements sont les suivantes : i) la détection et l'extraction des événements de risque naturel à partir des contenus de publications sociales. ii) l'introduction d'un traitement flou générant le degré de danger pour chaque événement extrait. iii) Le développement d'un processus efficace pour évaluer les communautés des citoyens en appliquant notre classification hiérarchique mixte. iv) faciliter le partage, l'intégration et l'analyse des évènements détectés en appliquant un processus de visualisation. En fait, les citoyens classés ont été injecté comme input aux tableaux de bord Power BI à des fins d'analyse visuelle.

Dans cette thèse, nous avons appliqué l'aspect incertain pour déterminer uniquement le degré de danger des citoyens. Pour améliorer l'approche proposée, nous avons inclus l'aspect incertain pour l'analyse et la modélisation des réseaux sociaux.

Le problème de prédiction de liens est un domaine de recherche crucial considéré pour l'analyse des réseaux sociaux. Il s'agit de chercher les liens dans le réseau en considérant son état présent. De nombreuses techniques ont été introduites pour résoudre ce problème. Cependant, la majorité de ces méthodes le traitent dans un cadre certain. Il est constaté que les données des réseaux sociaux sont généralement incomplètes et bruitées. Pour cette raison, il est important de gérer l'incertitude pendant le processus de prédiction.

Références

Aggarwal, C. C. (2011). An introduction to social network data analytics. In Social network data analytics, pages 1–15. Springer.

Aiello, L. M., Petkos, G., Martin, C., Corney, D., Papadopoulos, S., Skraba, R., Goker, A., Kompatsiaris, I., and Jaimes, A. (2013). Sensing trending topics in twitter. IEEE Transactions on Multimedia, 15(6) :1268–1282.

Akcora, C. G., Carminati, B., and Ferrari, E. (2013). User similarities on social networks. Social Network Analysis and Mining, 3(3) :475–495.

Alfaro, C., Cano-Montero, J., Gómez, J., Moguerza, J. M., and Ortega, F. (2016). A multi-stage method for content classification and opinion mining on weblog comments. Annals of Operations Research, 236(1) :197–213.

Anderson, A., Huttenlocher, D., Kleinberg, J., and Leskovec, J. (2012). Effects of user similarity in social media. In Proceedings of the fifth ACM international conference on Web search and data mining, pages 703–712. ACM.

Aone, C. and Ramos-Santacruz, M. (2000). Rees : a large-scale relation and event extraction system. In Proceedings of the sixth conference on Applied natural language processing, pages 76–83. Association for Computational Linguistics.

Arenas, A., Díaz-Guilera, A., and Pérez-Vicente, C. J. (2006). Synchronization reveals topological scales in complex networks. Physical review letters, 96(11) :114102.

Arenas, A., Fernandez, A., and Gomez, S. (2008). Analysis of the structure of complex networks at different resolution levels. New journal of physics, 10(5) :053039.

Atefeh, F. and Khreich, W. (2015). A survey of techniques for event detection in twitter. Computational Intelligence, 31(1) :132–164.

Aynaud, T., Blondel, V. D., Guillaume, J.-L., and Lambiotte, R. (2011). Optimisation locale multi-niveaux de la modularité.

Aynaud, T., Fleury, E., Guillaume, J.-L., and Wang, Q. (2013). Communities in evolving networks : definitions, detection, and analysis techniques. In Dynamics On and Of Complex Networks, Volume 2, pages 159–200. Springer.

Bakliwal, A., Foster, J., van der Puil, J., O'Brien, R., Tounsi, L., and Hughes, M. (2013). Sentiment analysis of political tweets : Towards an accurate

classifier. Association for Computational Linguistics.

Bedi, P. and Sharma, C. (2016). Community detection in social networks. Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery, 6(3) :115–135.

Benhardus, J. and Kalita, J. (2013). Streaming trend detection in twitter. International Journal of Web Based Communities, 9(1) :122–139.

Berger-Wolf, T. Y. and Saia, J. (2006). A framework for analysis of dynamic social networks. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 523–528. ACM.

Bhattacharyya, P., Garg, A., and Wu, S. F. (2011). Analysis of user keyword similarity in online social networks. Social network analysis and mining, 1(3) :143–158.

Bjorne, J., Ginter, F., Pyysalo, S., Tsujii, J., and Salakoski, T. (2010). Complex event extraction at pubmed scale. Bioinformatics, 26(12) :i382–i390.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan) :993–1022.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. Journal of statistical mechanics : theory and experiment, 2008(10) :P10008.

Boguná, M., Pastor-Satorras, R., Díaz-Guilera, A., and Arenas, A. (2004). Models of social networks based on social distance attachment. Physical review E, 70(5) :056122.

Boididou, C., Middleton, S. E., Jin, Z., Papadopoulos, S., Dang-Nguyen, D.-T., Boato, G., and Kompatsiaris, Y. (2018). Verifying information with multimedia content on twitter. Multimedia Tools and Applications, 77(12) :15545–15571.

Brandes, U. (2008). On variants of shortest-path betweenness centrality and their generic computation. Social Networks, 30(2) :136–145.

Carpenter, B. (2005). Scaling high-order character language models to gigabytes. In Proceedings of the Workshop on Software, pages 86–99. Association for Computational Linguistics.

Castrillo, E., León, E., and Gómez, J. (2017). Fast heuristic algorithm for multi-scale hierarchical community detection. In Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, pages 982–989. ACM.

Cavnar, W. B., Trenkle, J. M., et al. (1994). N-gram-based text categoriza-

tion. In Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval, volume 161175. Citeseer.

Chaabani, Y., Toujani, R., and Akaichi, J. (2018). Sentiment analysis method for tracking touristics reviews in social media network. In International conference on intelligent interactive multimedia systems and services, pages 299–310. Springer.

Chen, J. (2008). The construction and application of chinese emotion word ontology. Dalian University of Technology, Liaoning.

Choudhury, N. and Uddin, S. (2016). Time-aware link prediction to explore network effects on temporal knowledge evolution. Scientometrics, 108(2) :745–776.

Chowdhary, A. (2017). Community detection : Hierarchical clustering algorithms. Int. J. Creat. Res. Thoughts, 5(4) :2320–2882.

Chun, H.-W., Hwang, Y.-S., and Rim, H.-C. (2004). Unsupervised event extraction from biomedical literature using co-occurrence information and basic patterns. In International Conference on Natural Language Processing, pages 777–786. Springer.

Clauset, A., Moore, C., and Newman, M. E. (2008). Hierarchical structure and the prediction of missing links in networks. Nature, 453(7191) :98.

Collingsworth, B. and Menezes, R. (2014). A self-organized approach for detecting communities in networks. Social Network Analysis and Mining, 4(1) :169.

Combe, D. (2013). Détection de communautés dans les réseaux d'information utilisant liens et attributs. David Combe.

Commission, E. T. et al. (2013). New media trend watch. World Usage Patterns and Demographics [Internet].

Creusefond, J. (2017). Caractériser et détecter les communautés dans les réseaux sociaux. PhD thesis, Normandie Université.

Cui, H., Mittal, V., and Datar, M. (2006). Comparative experiments on sentiment classification for online product reviews. In AAAI, volume 6, page 30.

da Silva Soares, P. R. and Prudêncio, R. B. C. (2012). Time series based link prediction. In The 2012 international joint conference on neural networks (IJCNN), pages 1–7. IEEE.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery : Opinion extraction and semantic classification of product reviews. In Proceedings of the 12th international conference on World Wide Web, pages 519–528. ACM.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. IEEE transactions on pattern analysis and machine intelligence, (2) :224–227.

De Montgolfier, F., Soto, M., and Viennot, L. (2011). Asymptotic modularity of some graph classes. In International Symposium on Algorithms and Computation, pages 435–444. Springer.

Denecke, K. (2008). Using sentiwordnet for multilingual sentiment analysis. In 2008 IEEE 24th International Conference on Data Engineering Workshop, pages 507–512. IEEE.

Dragoni, M., Tettamanzi, A. G., and da Costa Pereira, C. (2014). A fuzzy system for concept-level sentiment analysis. In Semantic web evaluation challenge, pages 21–27. Springer.

Dugué, N. and Perez, A. (2014). Social capitalists on twitter : detection, evolution and behavioral analysis. Social Network Analysis and Mining, 4(1) :178.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.

Durant, K. T. and Smith, M. D. (2006). Mining sentiment classification from political web logs. In Proceedings of Workshop on Web Mining and Web Usage Analysis of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (WebKDD-2006), Philadelphia, PA.

Dutta, S., Ghatak, S., Roy, M., Ghosh, S., and Das, A. K. (2015). A graph based clustering technique for tweet summarization. In 2015 4th international conference on reliability, infocom technologies and optimization (ICRITO)(trends and future directions), pages 1–6. IEEE.

Esuli, A. and Sebastiani, F. (2005). Determining the semantic orientation of terms through gloss classification. In Proceedings of the 14th ACM international conference on Information and knowledge management, pages 617–624. ACM.

Esuli, A. and Sebastiani, F. (2006a). Determining term subjectivity and term orientation for opinion mining. In 11th Conference of the European Chapter of the Association for Computational Linguistics.

Esuli, A. and Sebastiani, F. (2006b). Sentiwordnet : A publicly available lexical resource for opinion mining. In LREC, volume 6, pages 417–422. Citeseer.

Falkowski, T., Bartelheimer, J., and Spiliopoulou, M. (2006). Mining and visualizing the evolution of subgroups in social networks. In Proceedings of the 2006 IEEE/WIC/ACM International Conference

on Web Intelligence, pages 52–58. IEEE Computer Society.

Falkowski, T., Barth, A., and Spiliopoulou, M. (2008). Studying community dynamics with an incremental graph mining algorithm. AMCIS 2008 Proceedings, page 29.

Farzindar, A. and Roche, M. (2013). Les défis de l'analyse des réseaux sociaux pour le traitement automatique des langues. Revue TAL-Traitement Automatique des Langues, 54(3) :7–16.

Fortunato, S. (2017). Benchmark graphs to test community detection algorithms. URL https ://sites. google. com/site/santofortunato/inthepress2.

Fortunato, S. and Hric, D. (2016). Community detection in networks : A user guide. Physics reports, 659 :1–44.

Frenken, K. and Mendritzki, S. (2012). Optimal modularity : a demonstration of the evolutionary advantage of modular architectures. Journal of Evolutionary Economics, 22(5) :935–956.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. Proceedings of the national academy of sciences, 99(12) :7821–7826.

Gonzalez-Pardo, A., Jung, J. J., and Camacho, D. (2017). Aco-based clustering for ego network analysis. Future Generation Computer Systems, 66 :160–170.

Guille, A. and Favre, C. (2014). Mention-anomaly-based event detection and tracking in twitter. In 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2014), pages 375–382. IEEE.

Guimera, R., Sales-Pardo, M., and Amaral, L. A. N. (2004). Modularity from fluctuations in random graphs and complex networks. Physical Review E, 70(2) :025101.

Gutiérrez-Esparza, G. O., Vallejo-Allende, M., and Hernández-Torruco, J. (2019). Classification of cyber-aggression cases applying machine learning. Applied Sciences, 9(9) :1828.

Ha, H., Han, H., Mun, S., Bae, S., Lee, J., and Lee, K. (2019). An improved study of multilevel semantic network visualization for analyzing sentiment word of movie review data. Applied Sciences, 9(12) :2419.

Haccianella, S., Esuli, A., and Sebastiani, F. S. (2010). 3.0 : An enhanced lexical resource for sentiment analysis and opinion mining. In Proceedings of the Seventh conference on International Language Resources and Evaluation.

Hajto, K., Kamieniecki, K., Misztal, K., and Spurek, P. (2017). Split-and-merge tweak in cross entropy clustering. In IFIP International Conference on Computer Information Systems and Industrial Management, pages 193–204. Springer.

Hastings, M. B. (2006). Community detection as an inference problem. Physical Review E, 74(3) :035102.

Herrmann, S., Ochoa, G., and Rothlauf, F. (2016). Communities of local optima as funnels in fitness landscapes. In Proceedings of the Genetic and Evolutionary Computation Conference 2016, pages 325–331. ACM.

Hogenboom, F., Frasincar, F., Kaymak, U., De Jong, F., and Caron, E. (2016). A survey of event extraction methods from text for decision support systems. Decision Support Systems, 85 :12–22.

Hu, Y., John, A., Seligmann, D. D., and Wang, F. (2012). What were the tweets about ? topical associations between public events and twitter feeds. In Sixth International AAAI Conference on Weblogs and Social Media.

Hung, S.-H., Lin, C.-H., and Hong, J.-S. (2010). Web mining for event-based commonsense knowledge using lexico-syntactic pattern matching and semantic role labeling. Expert Systems with Applications, 37(1) :341–347.

Hurst, M. F. and Nigam, K. (2003). Retrieving topical sentiments from online document collections. In Document Recognition and Retrieval XI, volume 5296, pages 27–34. International Society for Optics and Photonics.

IJntema, W., Sangers, J., Hogenboom, F., and Frasincar, F. (2012). A lexico-semantic pattern language for learning ontology instances from text. Web Semantics : Science, Services and Agents on the World Wide Web, 15 :37–50.

Jabreel, M. and Moreno, A. (2019). A deep learning-based approach for multi-label emotion classification in tweets. Applied Sciences, 9(6) :1123.

Jarvis, R. A. and Patrick, E. A. (1973). Clustering using a similarity measure based on shared near neighbors. IEEE Transactions on computers, 100(11) :1025–1034.

Jefferson, C., Liu, H., and Cocea, M. (2017). Fuzzy approach for sentiment analysis. In 2017 IEEE international conference on fuzzy systems (FUZZ-IEEE), pages 1–6. IEEE.

Jeh, G. and Widom, J. (2002). Simrank : a measure of structural-context

similarity. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 538–543. ACM.

Joachims, T. and Sebastiani, F. (2002). Guest editors' introduction to the special issue on automated text categorization. Journal of Intelligent Information Systems, 18(2) :103–105.

Kang, H., Yoo, S. J., and Han, D. (2012). Senti-lexicon and improved naïve bayes algorithms for sentiment analysis of restaurant reviews. Expert Systems with Applications, 39(5) :6000–6010.

Kernighan, B. W. and Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. Bell system technical journal, 49(2) :291–307.

Khan, M. A. H., Bollegala, D., Liu, G., and Sezaki, K. (2013). Multi-tweet summarization of real-time events. In 2013 International Conference on Social Computing, pages 128–133. IEEE.

Kim, B., Kim, J., and Yi, G. (2017). Analysis of clustering evaluation considering features of item response data using data mining technique for setting cut-off scores. Symmetry, 9(5) :62.

Kim, H. and Jeong, Y.-S. (2019). Sentiment classification using convolutional neural networks. Applied Sciences, 9(11) :2347.

Koschützki, D., Lehmann, K. A., Tenfelde-Podehl, D., and Zlotowski, O. (2005). Advanced centrality concepts. In Network analysis, pages 83–111. Springer.

Krings, G. and Blondel, V. D. (2011). An upper bound on community size in scalable community detection. arXiv preprint arXiv :1103.5569.

Lancichinetti, A. and Fortunato, S. (2011). Limits of modularity maximization in community detection. Physical review E, 84(6) :066122.

Lau, J. H., Collier, N., and Baldwin, T. (2012). On-line trend analysis with topic models :# twitter trends detection topic model online. Proceedings of COLING 2012, pages 1519–1534.

Lee, C.-S., Chen, Y.-J., and Jian, Z.-W. (2003). Ontology-based fuzzy event extraction agent for chinese e-news summarization. Expert Systems with Applications, 25(3) :431–447.

Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. (2008). Statistical properties of community structure in large social and information networks. In Proceedings of the 17th international conference on World Wide Web, pages 695–704. ACM.

Li, C., Sun, A., and Datta, A. (2012). Twevent : segment-based event detection from tweets. In Proceedings of the 21st ACM international

conference on Information and knowledge management, pages 155–164. ACM.

Li, F., Sheng, H., and Zhang, D. (2002). Event pattern discovery from the stock market bulletin. In International Conference on Discovery Science, pages 310–315. Springer.

Liben-Nowell, D. and Kleinberg, J. (2007). The link-prediction problem for social networks. Journal of the American society for information science and technology, 58(7) :1019–1031.

Liu, H. and Cocea, M. (2017). Fuzzy rule based systems for interpretable sentiment analysis. In 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI), pages 129–136. IEEE.

Liu, H., Hu, Z., Haddadi, H., and Tian, H. (2013). Hidden link prediction based on node centrality and weak ties. EPL (Europhysics Letters), 101(1) :18004.

Liu, W. and Lü, L. (2010). Link prediction based on local random walk. EPL (Europhysics Letters), 89(5) :58007.

Lu, Z., Savas, B., Tang, W., and Dhillon, I. S. (2010). Supervised link prediction using multiple sources. In 2010 IEEE international conference on data mining, pages 923–928. IEEE.

Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E., and Dawson, S. M. (2003). The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behavioral Ecology and Sociobiology, 54(4) :396–405.

Mamdani, E. H. and Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. International journal of man-machine studies, 7(1) :1–13.

Manning, C. D., Manning, C. D., and Schütze, H. (1999). Foundations of statistical natural language processing. MIT press.

Mao, X., Chang, S., Shi, J., Li, F., and Shi, R. (2019). Sentiment-aware word embedding for emotion classification. Applied Sciences, 9(7) :1334.

Meladianos, P., Xypolopoulos, C., Nikolentzos, G., and Vazirgiannis, M. (2018). An optimization approach for sub-event detection and summarization in twitter. In European Conference on Information Retrieval, pages 481–493. Springer.

Melville, P., Sindhwani, V., and Lawrence, R. (2009). Social media analytics : Channeling the power of the blogosphere for marketing insight. Proc. of the WIN, 1(1) :1–5.

Menon, A. K. and Elkan, C. (2011). Link prediction via matrix factoriza-

tion. In Joint european conference on machine learning and knowledge discovery in databases, pages 437–452. Springer.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv :1301.3781.

Miller, G. A. (1998). WordNet : An electronic lexical database. MIT press.

Mitra, B., Nalisnick, E., Craswell, N., and Caruana, R. (2016). A dual embedding space model for document ranking. arXiv preprint arXiv :1602.01137.

Mitrović, M. and Tadić, B. (2009). Spectral and dynamical properties in classes of sparse networks with mesoscopic inhomogeneities. Physical Review E, 80(2) :026123.

Mohammad, S. M., Kiritchenko, S., and Zhu, X. (2013). Nrc-canada : Building the state-of-the-art in sentiment analysis of tweets. arXiv preprint arXiv :1308.6242.

Moncla, L., Renteria-Agualimpia, W., Nogueras-Iso, J., and Gaio, M. (2014). Geocoding for texts with fine-grain toponyms : an experiment on a geoparsed hiking descriptions corpus. In Proceedings of the 22nd acm sigspatial international conference on advances in geographic information systems, pages 183–192. ACM.

Montoro, A., Olivas, J. A., Peralta, A., Romero, F. P., and Serrano-Guerrero, J. (2018). An anew based fuzzy sentiment analysis model. In 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pages 1–7. IEEE.

Moradi, P. and Rostami, M. (2015). Integration of graph clustering with ant colony optimization for feature selection. Knowledge-Based Systems, 84 :144–161.

Moshtaghi, M., Bezdek, J. C., Erfani, S. M., Leckie, C., and Bailey, J. (2018). Online cluster validity indices for streaming data. arXiv preprint arXiv :1801.02937.

Mucha, P. J., Richardson, T., Macon, K., Porter, M. A., and Onnela, J.-P. (2010). Community structure in time-dependent, multiscale, and multiplex networks. science, 328(5980) :876–878.

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., and Stoyanov, V. (2016). Semeval-2016 task 4 : Sentiment analysis in twitter. In Proceedings of the 10th international workshop on semantic evaluation (semeval-2016), pages 1–18.

Newman, M. E. (2006a). Finding community structure in networks using the

eigenvectors of matrices. Physical review E, 74(3) :036104.

Newman, M. E. (2006b). Modularity and community structure in networks. Proceedings of the national academy of sciences, 103(23) :8577–8582.

Newman, M. E. (2016). Community detection in networks : Modularity optimization and maximum likelihood are equivalent. arXiv preprint arXiv :1606.02319.

Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. Physical review E, 69(2) :026113.

Nicosia, Vincenzo Mangioni, G. C. (2009). Extending the definition of modularity to directed graphs with overlapping communities. Journal of Statistical Mechanics : Theory and Experiment, 2009(03) :P03024.

Noack, A. and Rotta, R. (2009). Multi-level algorithms for modularity clustering. In International Symposium on Experimental Algorithms, pages 257–268. Springer.

Nzali, A. A. M. D. T., Christian, J. A. S. B., and Poncelet, L. C. M. P. Advanse : Analyse du sentiment, de l'opinion et de l'émotion sur des tweets francais.

Okamoto, M. and Kikuchi, M. (2009). Discovering volatile events in your neighborhood : Local-area topic extraction from blog entries. In Asia Information Retrieval Symposium, pages 181–192. Springer.

Olteanu, A., Castillo, C., Diaz, F., and Vieweg, S. (2014). Crisislex : A lexicon for collecting and filtering microblogged communications in crises. In Eighth International AAAI Conference on Weblogs and Social Media.

Palla, G., Barabási, A.-L., and Vicsek, T. (2007). Quantifying social group evolution. Nature, 446(7136) :664.

Pang, B. and Lee, L. (2004). A sentimental education : Sentiment analysis using subjectivity summarization based on minimum cuts. In Proceedings of the 42nd annual meeting on Association for Computational Linguistics, page 271. Association for Computational Linguistics.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up ? : sentiment classification using machine learning techniques. In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, pages 79–86. Association for Computational Linguistics.

Papadopoulos, S., Corney, D., and Aiello, L. M. (2014). Snow 2014 data challenge : Assessing the performance of news topic detection methods in social media. In Snow-dc@ www, pages 1–8.

Parikh, R. and Karlapalem, K. (2013). Et : events from tweets. In Proceedings of the 22nd international conference on world wide web, pages 613–620. ACM.

Piskorski, J., Tanev, H., and Wennerberg, P. O. (2007). Extracting violent events from on-line news for ontology population. In International Conference on Business Information Systems, pages 287–300. Springer.

Pons, P. and Latapy, M. (2006). Computing communities in large networks using random walks. J. Graph Algorithms Appl., 10(2) :191–218.

Porter, M. A., Onnela, J.-P., and Mucha, P. J. (2009). Communities in networks. Notices of the AMS, 56(9) :1082–1097.

Prat-Pérez, A., Dominguez-Sal, D., and Larriba-Pey, J.-L. (2014). High quality, scalable and parallel community detection for large real graphs. In Proceedings of the 23rd international conference on World wide web, pages 225–236. ACM.

Pujari, M. and Kanawati, R. (2012). Link prediction in complex networks by supervised rank aggregation. In 2012 IEEE 24th International Conference on Tools with Artificial Intelligence, volume 1, pages 782–789. IEEE.

Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1) :81–106.

Quinlan, J. R. (2014). C4. 5 : programs for machine learning. Elsevier.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D. (2004). Defining and identifying communities in networks. Proceedings of the national academy of sciences, 101(9) :2658–2663.

Ratkiewicz, J., Conover, M. D., Meiss, M., Gonçalves, B., Flammini, A., and Menczer, F. M. (2011). Detecting and tracking political abuse in social media. In Fifth international AAAI conference on weblogs and social media.

Ravi, K. and Ravi, V. (2015). A survey on opinion mining and sentiment analysis : tasks, approaches and applications. Knowledge-Based Systems, 89 :14–46.

Reichardt, J. and Bornholdt, S. (2006). Statistical mechanics of community detection. Physical Review E, 74(1) :016110.

Riloff, E. and Wiebe, J. (2003). Learning extraction patterns for subjective expressions. In Proceedings of the 2003 conference on Empirical methods in natural language processing, pages 105–112.

Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. Proceedings of the National

Academy of Sciences, 105(4) :1118–1123.

Saoud, Z., Faci, N., Maamar, Z., and Benslimane, D. (2014). A fuzzy clustering-based credibility model for trust assessment in a service-oriented architecture. In 2014 IEEE 23rd International WETICE Conference, pages 56–61. IEEE.

Sarkar, P., Chakrabarti, D., and Moore, A. W. (2011). Theoretical justification of popular link prediction heuristics. In Twenty-Second International Joint Conference on Artificial Intelligence.

Sebastiani, F. (2002). Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1) :1–47.

Shalev-Shwartz, S., Singer, Y., and Ng, A. Y. (2004). Online and batch learning of pseudo-metrics. In Proceedings of the twenty-first international conference on Machine learning, page 94. ACM.

Shamma, D. A., Kennedy, L., and Churchill, E. F. (2011). Peaks and persistence : modeling the shape of microblog conversations. In Proceedings of the ACM 2011 conference on Computer supported cooperative work, pages 355–358. ACM.

Shi, L.-l., Liu, L., Wu, Y., Jiang, L., and Hardy, J. (2017). Event detection and user interest discovering in social media data streams. IEEE Access, 5 :20953–20964.

Shiha, M. and Ayvaz, S. (2017). The effects of emoji in sentiment analysis. Int. J. Comput. Electr. Eng.(IJCEE.), 9(1) :360–369.

Siddiqua, U. A., Ahsan, T., and Chy, A. N. (2016). Combining a rule-based classifier with weakly supervised learning for twitter sentiment analysis. In 2016 International Conference on Innovations in Science, Engineering and Technology (ICISET), pages 1–4. IEEE.

Spiliopoulou, M., Ntoutsi, I., Theodoridis, Y., and Schult, R. (2006). Monic : modeling and monitoring cluster transitions. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 706–711. ACM.

Staudt, C. L. and Meyerhenke, H. (2015). Engineering parallel algorithms for community detection in massive networks. IEEE Transactions on Parallel and Distributed Systems, 27(1) :171–184.

Su, Y., Wang, B., and Zhang, X. (2017). A seed-expanding method based on random walks for community detection in networks with ambiguous community structures. Scientific reports, 7 :41830.

Sun, Y., Tang, J., Han, J., Gupta, M., and Zhao, B. (2010). Community evolution detection in dynamic heterogeneous information networks.

In Proceedings of the Eighth Workshop on Mining and Learning with Graphs, pages 137–146. ACM.

Talbi, M. (2013). Une nouvelle approche de dÃ©tection de communautÃ©s dans les rÃ©seaux sociaux. PhD thesis, UniversitÃ© du QuÃ©bec en Outaouais.

Tanev, H., Piskorski, J., and Atkinson, M. (2008). Real-time news event extraction for global crisis monitoring. In International Conference on Application of Natural Language to Information Systems, pages 207–218. Springer.

Tang, F. (2017). Link-Prediction and its Application in Online Social Networks. PhD thesis, Victoria University.

Tang, J., Chang, Y., and Liu, H. (2014). Mining social media with social theories : a survey. ACM Sigkdd Explorations Newsletter, 15(2) :20–29.

Toujani, R. and Akaichi, J. (2016). Fuzzy sentiment classification in social network facebook'statuses mining. In 2016 7th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), pages 393–397. IEEE.

Toujani, R. and Akaichi, J. (2017). Optimal initial partitionning for high quality hybrid hierarchical community detection in social networks. In 2017 4th International Conference on Control, Decision and Information Technologies (CoDIT), pages 0395–0403. IEEE.

Toujani, R. and Akaichi, J. (2018). Ghhp : Genetic hybrid hierarchical partitioning for community structure in social medias networks. In 2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), pages 1146–1153. IEEE.

Toujani, R. and Akaichi, J. (2019a). An approach based on mixed hierarchical clustering and optimization for graph analysis in social media network : toward globally hierarchical community structure. Knowledge and Information Systems, pages 1–41.

Toujani, R. and Akaichi, J. (2019b). Event news detection and citizens community structure for disaster management in social networks. Online Information Review, 43(1) :113–132.

Toujani, R., Chaabani, Y., Dhouioui, Z., and Bouali, H. (2018a). The next generation of disaster management and relief planning : Immersive

analytics based approach. In International Conference on Immersive Learning, pages 80–93. Springer.

Toujani, R., Dhouioui, Z., and Akaichi, J. (2018b). Mobility based machine learning modeling for event mining in social networks. In International Conference on Intelligent Interactive Multimedia Systems and Services, pages 311–322. Springer.

Toujani Radhia, D. Z. and Jalel, A. (2015). Machine learning and metaheuristic for sentiment anaysis in social networks. In proceedings of the metaheuristic internatianal conference(MIC'15).

Turney, P. D. (2002). Thumbs up or thumbs down ? : semantic orientation applied to unsupervised classification of reviews. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 417–424. Association for Computational Linguistics.

Valkanas, G. and Gunopulos, D. (2013). How the live web feels about events. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management, pages 639–648. ACM.

Valverde-Rebaza, J. and de Andrade Lopes, A. (2013). Exploiting behaviors of communities of twitter users for link prediction. Social Network Analysis and Mining, 3(4) :1063–1074.

van den Broek-Altenburg, E. M. and Atherly, A. J. (2019). Using social media to identify consumers' sentiments towards attributes of health insurance during enrollment season. Applied Sciences, 9(10) :2035.

Van Laarhoven, T. and Marchiori, E. (2014). Axioms for graph clustering quality functions. The Journal of Machine Learning Research, 15(1) :193–215.

Vashishtha, S. and Susan, S. (2018). Fuzzy logic based dynamic plotting of mood swings from tweets. In International Conference on Innovations in Bio-Inspired Computing and Applications, pages 129–139. Springer.

Vashishtha, S. and Susan, S. (2019). Fuzzy rule based unsupervised sentiment analysis from social media posts. Expert Systems with Applications, 138 :112834.

Verma, S., Vieweg, S., Corvey, W. J., Palen, L., Martin, J. H., Palmer, M., Schram, A., and Anderson, K. M. (2011). Natural language processing to the rescue ? extracting" situational awareness" tweets during mass emergency. In Fifth International AAAI Conference on Weblogs and Social Media.

Viard, T., Latapy, M., and Magnien, C. (2016). Computing maximal cliques in link streams. Theoretical Computer Science, 609 :245–252.

Wang, C., Satuluri, V., and Parthasarathy, S. (2007). Local probabilistic models for link prediction. In Seventh IEEE international conference on data mining (ICDM 2007), pages 322–331. IEEE.

Wang, H. and Castanon, J. A. (2015). Sentiment expression via emoticons on social media. In 2015 ieee international conference on big data (big data), pages 2404–2408. IEEE.

Wei, H., Sankaranarayanan, J., and Samet, H. (2017). Finding and tracking local twitter users for news detection. In Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, page 64. ACM.

Wolny, W. (2016a). Emotion analysis of twitter data that use emoticons and emoji ideograms.

Wolny, W. (2016b). Sentiment analysis of twitter data using emoticons and emoji ideograms. Studia Ekonomiczne, 296 :163–171.

Wu, F.-Y. (1982). The potts model. Reviews of modern physics, 54(1) :235.

Xi, J., Zhan, W., and Wang, Z. (2016). Hierarchical community detection algorithm based on node similarity. International Journal of Database Theory and Application, 9(6) :209–218.

Xie, W., Zhu, F., Jiang, J., Lim, E.-P., and Wang, K. (2016). Topicsketch : Real-time bursty topic detection from twitter. IEEE Transactions on Knowledge and Data Engineering, 28(8) :2216–2229.

Yadollahi, A., Shahraki, A. G., and Zaiane, O. R. (2017). Current state of text sentiment analysis from opinion to emotion mining. ACM Computing Surveys (CSUR), 50(2) :25.

Yakushiji, A., Tateisi, Y., Miyao, Y., and Tsujii, J.-i. (2000). Event extraction from biomedical papers using a full parser. In Biocomputing 2001, pages 408–419. World Scientific.

Yang, Z., Algesheimer, R., and Tessone, C. J. (2016). A comparative analysis of community detection algorithms on artificial networks. Scientific reports, 6 :30750.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. Journal of anthropological research, 33(4) :452–473.

Zadeh, L. A. (2015). Fuzzy logic a personal perspective. Fuzzy sets and systems, 281 :4–20.

Zhang, W., Kong, F., Yang, L., Chen, Y., and Zhang, M. (2018). Hierarchical community detection based on partial matrix convergence using random walks. Tsinghua Science and Technology, 23(1) :35–46.

Zhen-Qing, Y., Ke, Z., Song-Nian, H., and Jun, Y. (2012). A new definition

of modularity for community detection in complex networks. Chinese Physics Letters, 29(9) :098901.

Zhou, C., Feng, L., and Zhao, Q. (2018). A novel community detection method in bipartite networks. Physica A : Statistical Mechanics and its Applications, 492 :1679–1693.

Zhou, T., Lü, L., and Zhang, Y.-C. (2009). Predicting missing links via local information. The European Physical Journal B, 71(4) :623–630.

Zhou, X., Coiera, E., Tsafnat, G., Arachi, D., Ong, M.-S., Dunn, A. G., et al. (2015). Using social connection information to improve opinion mining : Identifying negative sentiment about hpv vaccines on twitter.

Zhu, F. and Zhang, X. (2006). The influence of online consumer reviews on the demand for experience goods : The case of video games. ICIS 2006 Proceedings, page 25.