Influencers characterization in a social network for viral marketing perspectives

Siwar Jendoubi

ANNÉE 2016

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

En Cotutelle Internationale avec
L'université de Tunis, Tunisie

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

Ecole doctorale Matisse

présentée par

Siwar Jendoubi

préparée à l'unité de recherche UMR 6074 IRISA
Institut de Recherche en Informatique et Système Aléatoire
Université de Rennes I

Influencers Characterization in a Social Network for Viral Marketing Perspectives

Thèse soutenue à l'Université de Rennes I le 16/12/2016

devant le jury composé de :

Hanene AZZAG
Maitre de conférences à l' Université de Paris 13 France / rapporteur

Hend BEN HADJI
Docteur-ingénieur à CERT, Tunisie / examinateur

Frédéric DAMBREVILLE
Expert senior HDR, DGA-MI, Bruz, France / examinateur

Eric LEFEVRE
Professeur à l'Université d'Artois, France / rapporteur

Francis ROUSSEAUX
Professeur à l'Université de Reims, France / examinateur

Boutheina BEN YAGHLANE
Professeur à l'Universités de Carthage, Tunisie / co-directrice de thèse

Ludovic LIETARD
Maitre de conférences HDR à l'Université de Rennes 1, France / co-directeur de thèse

Arnaud MARTIN
Professeur à l'Université de Rennes 1, France / directeur de thèse

MOBILISATION DE CHERCHEURS AU PROFIT DES ENTREPRISES (Travaux de Recherche Doctorale dans l'Entreprise) Session 2013

«Ces travaux de recherche et innovation sont effectués dans le cadre du dispositif MOBIDOC financé par l'Union Européenne dans le cadre du programme PASRI et administré par l'ANPR en Tunisie».

Ce travail a été aussi partiellement financé par le "Centre d'Etude et de Recherche des Télécommunications", CERT Tunisie.

Abstract

The Viral Marketing is a relatively new form of marketing that exploits social networks in order to promote a product, a brand, etc. It is based on the influence that exerts one user on another. The influence maximization is the scientific problem for the Viral Marketing. In fact, its main purpose is to select a set of influential users that could adopt the product and trigger a large cascade of influence and adoptions through the network. In this thesis, we propose two evidential influence maximization models for social networks. The proposed approach uses the theory of belief functions to estimate the user's influence. Furthermore, we introduce an influence measure that fuses many influence aspects, like the importance of the user in the network and the popularity of his messages. Next, we propose three Viral Marketing scenarios. For each scenario we introduce two influence measures. The first scenario is about influencers having a positive opinion about the product. The second scenario searches for influencers having a positive opinion and influence positive opinion users and the last scenario looks for influencers having a positive opinion and exert more influence on negative opinion users. On the other hand, we turned to another important problem which is about the prediction of the social message topic. Indeed, the topic is also an important parameter in the influence maximization problem. For this purpose, we introduce four classification algorithms that do not need the content of the message to classify it, they just need its propagation traces. In our experiments, we compare the proposed solutions to existing ones and we show the performance of the proposed influence maximization solutions and the proposed classifiers.

Résumé

Le marketing viral est une nouvelle forme de marketing qui exploite les réseaux sociaux afin de promouvoir un produit, une marque, etc. Il se fonde sur l'influence qu'exerce un utilisateur sur un autre. La maximisation de l'influence est le problème scientifique pour le marketing viral. En fait, son but principal est de sélectionner un ensemble d'utilisateurs d'influences qui pourraient adopter le produit et déclencher une large cascade d'influence et d'adoption à travers le réseau. Dans cette thèse, nous proposons deux modèles de maximisation de l'influence sur les réseaux sociaux. L'approche proposée utilise la théorie des fonctions de croyance pour estimer l'influence des utilisateurs. En outre, nous introduisons une mesure d'influence qui fusionne de nombreux aspects d'influence, comme l'importance de l'utilisateur sur le réseau et la popularité de ces messages. Ensuite, nous proposons trois scénarii de marketing viral. Pour chaque scénario, nous introduisons deux mesures d'influence. Le premier scénario cherche les influenceurs ayant une opinion positive sur le produit. Le second scénario concerne les influenceurs ayant une opinion positive et qui influencent des utilisateurs ayant une opinion positive et le dernier scénario cherche des influenceurs ayant une opinion positive et qui influencent des utilisateurs ayant une opinion négative. Dans un deuxième lieu, nous nous sommes tournés vers un autre problème important, qui est le problème de la prédiction du sujet du message social. En effet, le sujet est également un paramètre important dans le problème de la maximisation de l'influence. A cet effet, nous introduisons quatre algorithmes de classification qui ne nécessitent pas le contenu du message pour le classifier, nous avons juste besoin de ces traces de propagation. Dans nos expérimentations, nous comparons les solutions proposées aux solutions existantes et nous montrons la performance des modèles de la maximisation de l'influence et les classificateurs proposés.

Acknowledgements

My great gratitude goes first to my advisors Pr. Arnaud Martin, Pr. Boutheina Ben Yaghlane, Dr. Ludovic Liétard and Dr. Hend Ben Hadji, who expertly guided me through my graduate education and who shared the excitement of all these years of discovery. Besides, I would like to express my sincere thanks to them for introducing me to an interesting area of research and also for all valuable discussion we have had during these four years.

Second, I would like to thank the members of my thesis committee: Pr. Francis Rousseaux, Pr. Eric Lefèvre, Dr. Hanene Azzag, Dr. Frédéric Dambreville and Dr. Tanguy Urvoy for accepting to review my dissertation.

I would like to address my thanks to all colleagues in my laboratory LARODEC and my team DRUID for their help and kindness. A special thank for all colleages in IUT of Lannion and in CERT for their encouragement, kindness and support.

I would especially like to acknowledge the financial and/or technical support of LARODEC, ISG, University of Tunis, CERT, MOBIDOC and IUT of Lannion.

On a more private note, I thank my lovely parents and sisters for supporting me and for providing a wonderful environment in which I could make progress. I also thank all my friends for their support, help and advices in all difficult periods.

Caractérisation des influenceurs dans un réseau social pour des perspectives de Marketing Viral

1 Introduction

Récemment, l'attention des entreprises a été attirée par une nouvelle forme de marketing, couramment appelée *Marketing Viral*. Le Marketing Viral est le processus de cibler les utilisateurs les plus influents dans un réseau social de telle sorte que ces clients peuvent déclencher une réaction en cascade d'influence entrainée par le bouche-à-oreille. Ainsi avec un petit budget de marketing, une grande proportion d'un réseau social peut être atteinte ou influencée [1]. L'enjeu du Marketing Viral est donc de trouver un ensemble d'utilisateurs d'influence à cibler pour déclencher un processus de bouche-à-oreille.

Le problème du Marketing Viral se traduit, scientifiquement, par un problème de maximisation de l'influence des personnes. Le problème est de sélectionner parmi les socionautes, un ensemble de k utilisateurs qui sont capables de déclencher une large cascade de propagation et d'influence. Dans la littérature, plusieurs travaux de recherche cherchent à résoudre ce problème en proposant des modèles de maximisation de l'influence. Le problème de ces modèles est qu'ils n'utilisent que la structure du réseau pour détecter les influenceurs, alors que la position de l'utilisateur sur le réseau est généralement insuffisante pour confirmer son influence [41]. Il est ainsi important de chercher des indicateurs d'influence plus performants.

Parmi les indicateurs d'influence, nous trouvons la propagation de l'information sur le réseau, l'importance de l'utilisateur sur le réseau qui peut être mesurer en nombre de messages qui lui ont été envoyés directement, la position de l'utilisateur dans le réseau ainsi que son opinion. En fusionnant ces indicateurs d'influence, nous disposerions d'une mesure

d'influence puissante. Dans ce document, nous introduisons des mesures d'influence qui tiennent compte de ces indicateurs tout en utilisant la théorie des fonctions de croyance [79] pour les fusionner.

Un problème important auquel nous nous sommes, également, intéressés, est le problème de la classification des messages sociaux. En effet, les messages sociaux ont des caractéristiques particulières qui les différencient du texte ordinaire. Parmi ces caractéristiques, on trouve que le message social est de petite taille ce qui conduit au problème de la sparcification, *i.e.* le nombre des mots est insuffisant pour la prédiction de la classe du message. Par conséquent, il est important de trouver une nouvelle approche de classification qui résout ce problème. En fait, les sujets auxquels un utilisateur de réseau social est intéressé, est une information très utile pour de nombreuses applications parmi lesquelles le marketing viral.

Dans ce qui suit, nous présentons l'état de l'art de la maximisation de l'influence. Puis, nous introduisons les solutions que nous proposons pour améliorer la qualité des influenceurs sélectionnés. Ensuite, nous présentons les résultats de nos expérimentations. Par la suite, nous nous focalisons sur le problème de la classification des messages sociaux et finalement nous concluons ce résumé.

2 Modèles existants de maximisation de l'influence

Domingos et Richardson [30] ont été les premiers à modéliser le problème du Marketing Viral en un problème de fouille de données. Motivés par ce travail, *Kempe et al.* [55] ont abordé ce problème. Ils l'ont formulé en un problème de maximisation de l'influence. C'est un problème d'optimisation NP-Difficile. Comme solutions, les auteurs ont proposé d'utiliser le modèle en cascade indépendants [37] et le modèle à seuil linéaire [43] (Granovetter, 1978). Ces modèles sont utilisés pour l'approximation d'une fonction σ mesurant le gain que peut apporter un ensemble d'utilisateurs. Le problème est donc de maximiser la fonction gain σ. *Kempe et al.* [55] ont montré que le problème de maximisation de l'influence peut être résolu avec une bonne approximation en utilisant l'algorithme glouton. *Leskovec et al.* [60] ont proposé une amélioration du processus de la maximisation de l'influence, ils ont développé l'algorithme "Cost-Effective Lazy Forward" (CELF) qui a permis d'accélérer le processus jusqu'à 700 fois tout en garantissant la même approximation de l'algorithme glouton.

Les travaux présentés ci-dessus considèrent disposer d'un réseau social dont les probabilités d'influence sur les liens sont connues, mais l'apprentissage de ces probabilités n'est pas traité. *Goyal et al.* [40] ont étudié le problème d'apprentissage des probabilités d'influence. *Goyal et al.* [41] ont proposé une approche de maximisation de l'influence qui utilise les traces de propagation pour la prédiction directe de la propagation de l'influence. Ainsi, leur approche n'utilise pas un modèle explicite de diffusion. *Wei et al.* [94] et *Gao et al.* [35] ont introduit la théorie des fonctions de croyance [26, 79] dans le processus de l'identification des nœuds influencés.

L'opinion de l'utilisateur par rapport au produit est un paramètre important dans le problème de maximisation de l'influence. Le travail de *Chen et al.* [23] a été parmi les premiers travaux considéront l'opinion négative dans le modèle en cascade indépendant. Les auteurs affirment que l'avis négatif est plus contagieux que l'avis positif dans les décisions et les choix des gens. De même, l'étude de *Wen et al.* [95] a été l'une des premières tentatives qui se concentrent non seulement sur le type d'information, mais aussi la propagation simultanée de l'information négative (comme les rumeurs) et positive (comme les idées et les nouvelles). Cependant, ils ne considèrent pas l'avis de l'utilisateur par rapport au produit. Un autre travail intéressant est celui de *Zhang et al. [99].* Les auteurs ont proposé le modèle en cascade à base d'opinions qui prend en considération les avis positifs des utilisateurs. Ils ont utilisé leur modèle afin de maximiser l'influence positive en tenant compte de l'avis de l'utilisateur et le changement de son opinion.

Dans la section suivante, nous introduisons les solutions que nous proposons pour améliorer la qualité des utilisateurs d'influence sélectionnés par rapport aux solutions de maximisation de l'influence existantes.

3 Solutions proposées pour la maximisation de l'influence

La plupart des solutions de maximisation de l'influence existantes n'ont pas considéré de nombreux aspects d'influence. En fait, on constate que la plupart des solutions existantes utilisent uniquement la structure du réseau pour identifier les influenceurs [55, 23, 99, 57]. Toutefois, la structure du réseau ne suffit pas pour résoudre ce problème [41]. En effet, ces solutions peuvent souvent tomber sur des influenceurs bien positionnés sur le réseau mais qui sont inactifs. Par conséquent, il est devenu nécessaire de trouver des solutions de maximisation de l'influence dans lesquelles nous considérons plus d'aspects d'influence tels que l'activité de l'utilisateur dans le réseau.

L'opinion de l'utilisateur est un autre paramètre important. En fait, ce paramètre n'a pas été pris en compte dans la plupart des travaux existants. En outre, nous trouvons quelques travaux qui tiennent compte de l'opinion dans leur modèle. Cependant, ce n'est pas toujours une opinion sur le produit. Ainsi, vient la nécessité de nouvelles solutions qui tiennent compte de l'avis de l'utilisateur par rapport au produit.

Afin de remédier à ces problèmes, nous proposons des solutions de maximisation de l'influence en tenant compte de plusieurs aspects d'influence comme la position de l'utilisateur sur le réseau et son activité de propagation de l'information. Nous introduisons alors une amélioration qui considère l'opinion de l'utilisateur par rapport au produit.

3.1 Mesure incertaine d'influence

Tout d'abord, nous proposons une mesure d'influence pour Twitter qui combine de nombreux aspects d'influence. Cette mesure a fait l'objet du papier *Jendoubi et al.* [49]. En premier lieu, nous commençons par l'estimation d'un ensemble de poids définis sur les liens (u, v) du réseau. Pour ce faire, nous utilisons quelques statistiques sur les utilisateurs, $u, v \in V$ tel que V est l'ensemble des nœuds, du réseau à savoir: 1) le poids définissant l'ensemble de voisins en commun entre u et v: $w_f(u, v)$. 2) le poids définissant le nombre de fois que u a mentionné v dans ces tweets[1]: $w_m(u, v)$. Et 3) le poids définissant le nombre de fois que v a partagé les tweets de u: $w_r(u, v)$.

Dans un deuxième temps, nous passons au niveau du nœud et nous calculons les trois poids pour chaque nœud. Les poids des nœuds sont obtenus en additionnant les poids sur ses liens sortant: $w_x(u) = \sum_{v \in V} w_x(u, v)$, tel que $w_x(u) \in \{w_f(u), w_r(u), w_m(u)\}$ et $w_x(u, v) \in \{w_f(u, v), w_r(u, v), w_m(u, v)\}$. Dans un troisième temps, nous mettons à jour les poids sur les liens en considérant les poids sur leurs nœuds de destination et nous obtenons $w_x^{'}(u, v) \in \left\{w_f^{'}(u, v), w_r^{'}(u, v), w_m^{'}(u, v)\right\}$. L'objectif principal de cette étape est de considérer l'hypothèse qui dit "*je suis plus influenceur, si je suis connecté à des influenceurs*".

La prochaine étape du processus de l'estimation de l'influence consiste à estimer une distribution de masse de croyance pour chaque poids sur les liens, comme suit :

$$m^{\Omega}_{x_{(u,v)}}(I) = \frac{w_x^{'}(u, v) - L_{min_x}}{\pi_x} \quad (1)$$

$$m^{\Omega}_{x_{(u,v)}}(P) = \frac{L_{max_x} - w_x^{'}(u, v)}{\pi_x} \quad (2)$$

$$m^{\Omega}_{x_{(u,v)}}(\{I, P\}) = 1 - \left(m^{\Omega}_{x_{(u,v)}}(I) + m^{\Omega}_{x_{(u,v)}}(P)\right) \quad (3)$$

sachant que $\Omega = \{I, P\}$, I pour un influenceur et P pour un utilisateur passif, $L_{min_x} = \min_{(u,v) \in E} w_x^{'}(u, v)$, $L_{max_x} = \max_{(u,v) \in E} w_x^{'}(u, v)$ et $\pi_x = L_{max_x} - L_{min_x} + \varepsilon$. Comme résultats, nous avons trois distributions de masses $m^{\Omega}_{f_{(u,v)}}$, $m^{\Omega}_{m_{(u,v)}}$ et $m^{\Omega}_{r_{(u,v)}}$. Après, nous combinons les résultats des trois masses obtenues en utilisant la règle de combinaison de Dempster [26] et nous obtenons $m^{\Omega}_{(u,v)} = \left(m^{\Omega}_{f_{(u,v)}} \oplus m^{\Omega}_{m_{(u,v)}}\right) \oplus m^{\Omega}_{r_{(u,v)}}$. Finalement, l'influence de l'utilisateur u sur v est définie par l'équation suivante :

$$Inf(u, v) = m^{\Omega}_{(u,v)}(I) \quad (4)$$

Nous notons que la mesure proposée peut être adaptée à d'autres réseaux sociaux.

[1] un message de 140 caractères publié sur Twitter

3.2 Considération de l'opinion de l'utilisateur

Nous améliorons la mesure d'influence proposée par la prise en compte de l'opinion de l'utilisateur. En effet, nous introduisons trois scénarios de marketing viral qui peuvent se produire dans les cas réels. Le premier scénario cherche à détecter les influenceurs ayant une opinion positive. Une solution pour ce premier scénario a été publiée dans *Jendoubi et al.* [48]. Pour ce scénario, nous proposons les deux mesures d'influence suivantes qui tiennent compte de l'opinion positive de l'utisateur :

$$Inf_1^+(u,v) = \mathrm{Pr}_u^\Theta(Pos).m_{(u,v)}^\Omega(I) \quad (5)$$
$$Inf_2^+(u,v) = m_u^\Theta(Pos).m_{(u,v)}^\Omega(I) \quad (6)$$

sachant que Pr_u^Θ et m_u^Θ sont respectivement une distribution de probabilités et une distribution de masse de croyance qui définissent l'opnion de u sur le cadre de décernement $\Theta = \{Pos, Neg, Neut\}$, Pos pour une opinion positive, Neg pour une opinion négative et $Neut$ pour une opinion neutre.

Le deuxième scénario proposé, cherche les influenceurs ayant une opinion positive et qui exercent plus d'influence sur les utilisateurs ayant une opinion positive. Pour cet objectif nous proposons les mesures d'influence suivantes :

$$Inf_1^{++}(u,v) = Inf_1^+(u,v).\left(1-\mathrm{Pr}_v^\Theta(Neg)\right) \quad (7)$$
$$Inf_2^{++}(u,v) = Inf_2^+(u,v).\left(1-m_v^\Theta(Neg)\right) \quad (8)$$

Le troisième scénario concerne les influenceurs ayant une opinion positive et qui exercent plus d'influence sur les utilisateurs négatifs. Pour ce scénario, nous introduisons les deux mesures d'influence suivantes :

$$Inf_1^{+-}(u,v) = Inf_1^+(u,v).\left(1-\mathrm{Pr}_v^\Theta(Pos)\right) \quad (9)$$
$$Inf_2^{+-}(u,v) = Inf_2^+(u,v).\left(1-m_v^\Theta(Pos)\right) \quad (10)$$

3.3 Modèles de maximisation de l'influence

Nous nous tournons vers le problème de la maximisation de l'influence et nous introduisons deux modèles de maximisation d'influence. Ces modèles sont publiés dans le papier *Jendoubi et al.* [49]. Nous introduisons les deux formules suivantes pour estimer l'influence d'un ensemble d'utilisateurs S sur le nœud v :

$$\Phi_{M1}(S,v) = \begin{cases} 1 & \text{si } v \in S \\ \sum_{u \in S} Inf(u,v) & \text{Sinon} \end{cases} \quad (11)$$

$$\Phi_{M2}(S,v) = \begin{cases} 1 & \text{si } v \in S \\ \sum_{u \in S} \sum_{x \in D_{IN}(v) \cup \{v\}} Inf(u,x).Inf(x,v) & \text{Sinon} \end{cases} \quad (12)$$

de telle sorte que $Inf(v,v) = 1$ et $D_{IN}(v)$ est l'ensemble des voisins entrants de v. Le travail de *Chen et al.* [23] justifie les deux modèles proposés. En fait, ils affirment que lorsque le produit présente quelques problèmes de qualité, il est plus adaptable de choisir des influenceurs ayant beaucoup de voisins immédiats d'où l'intérêt du premier modèle $\Phi_{M1}(S,v)$. En outre, lorsque le produit a une haute qualité, nous pouvons choisir le deuxième modèle $\Phi_{M2}(S,v)$ qui vise la profondeur du réseau. Ensuite, nous définissons les fonctions objectives à maximiser comme suit :

$$\sigma^{Bel}_{M1}(S) = \sum_{v \in V} \Phi_{M1}(S,v) \quad (13)$$

$$\sigma^{Bel}_{M2}(S) = \sum_{v \in V} \Phi_{M2}(S,v) \quad (14)$$

Les fonctions $\sigma^{Bel}_{M1}(S)$ et $\sigma^{Bel}_{M2}(S)$ sont les fonctions à maximiser. Nous avons prouvé que ces fonctions sont monotones et sous-modulaires. De plus, nous avons montré que la maximisation de l'influence en utilisant ces modèle est un problème NP-Défficile. Ainsi, nous proposons une solution de maximisation fondée sur l'algorithme glouton. En effet, nous avons utilisé l'algorithme CELF proposé par *Leskovec et al.* [60].

4 Maximisation de l'influence : Étude expérimentale

La section précédente est principalement dédiée à l'introduction des mesures d'influence et des modèles de maximisation de l'influence que nous proposons. Dans cette section, nous nous concentrons sur l'expérimentation des solutions proposées. En effet, nous étudions la performance des solutions proposées sur des données réelles collectées à partir de Twitter. Nous étudions aussi la qualité des influenceurs sélectionnés sur des données générées.

Nous introduisons deux ensembles de données, le premier a été collecté à partir de Twitter. En effet, nous avons collecté un jeu de données contenant les utilisateurs, les

liens entre eux, les tweets, les mentions[2] et les retweets[3]. Le second jeux de données a été généré. En effet, nous avons sélectionné, aléatoirement, 1010 nœuds et leurs liens des données collectées à partir de Twitter et nous avons généré aléatoirement les valeurs de l'influence et de l'opinion des utilisateurs.

Dans cette thèse, nous proposons un ensemble de mesures d'influence qui tiennent compte de l'opinion de l'utilisateur. A cet effet, nous avons besoin d'estimer cette opinion. Nous avons utilisé des outils existant qui sont dédiés à cet objectif à savoir le *Stanford Log-linear Part-Of-Speech Tagger*[4], le *GATE Twitter part-of-speech tagger*[5] et le dictionnaire d'opinion *SentiWordNet 3.0*. Tout d'abord, nous commençons par estimer la polarité de l'opinion de chaque tweet dans les données, puis, nous définissons l'opinion de l'utilisateur comme étant l'opinion moyenne de ses tweets.

Dans une première expérimentation effectuée sur les données réelles, nous n'avons pas considéré l'opinion et nous avons comparé les modèles de maximisation de l'influence proposés à quelques modèles de la littérature, à savoir le modèle en cascade indépendant, le modèle à seuil linéaire [55] et le modèle de distribution des crédits [41]. La principale conclusion à partir de cette expérimentation est que les solutions de maximisation d'influence proposées améliorent la qualité des influenceurs sélectionnés par rapport aux solutions de maximisation d'influence existantes. En effet, nous avons comparé les modèles en termes de nombre de #Follow[6], #Mention, #Retweet et #Tweet des influenceurs détectés. Ainsi, les modèles de maximisation de l'influence proposés sont très utiles pour promouvoir une campagne de marketing viral donnée.

Dans une deuxième expérimentation effectuée sur les données réelles, nous avons considéré l'opinion et nous avons comparé les mesures d'influence proposées avec le modèle de distribution des crédits [41] et le modèle en cascade à base d'opinion [99]. Cette expérience montre l'importance de l'opinion et sa contribution à l'amélioration de la qualité des influenceurs choisis, non seulement en fonction des critères d'influence, à savoir #Follow, #Mention, #Retweet et #Tweet, mais aussi en termes de l'opinion des influenceurs sélectionnées. En effet, nous réussissons à détecter des graines ayant une opinion positive sur le produit.

Dans une troisième expérimentation effectuée sur les données générées, nous avons comparé les mesures d'influence proposées en termes de précision. En effet, on a généré les données de telles sorte qu'on puisse savoir les influenceurs, les influenceurs positifs, les influenceurs positifs qui influencent des utilisateurs positifs et les influenceurs positifs qui influencent des utilisateurs négatifs. Nous avons alors évalué les algorithmes de maximisation

[2]La mention permet à un utilisateur de Twitter d'envoyer un tweet directement à d'autre utilisateurs en mentionnant leurs nom d'utilisateur dans le tweet.

[3]Le retweet est une fonctionnalité de Twitter qui permet le partage des tweets.

[4]http://nlp.stanford.edu/software/tagger.shtml

[5]https://gate.ac.uk/wiki/twitter-postagger.html

[6]Le follow est une relation sur Twitter qui permet à un utilisateur donné de suivre les mises à jour des autres utilisateurs qu'il suit.

de l'influence en calculant le taux de bonne classification. Cette expérience montre que tous les algorithmes de maximisation ont réussi à avoir au moins 80% de taux de bonne classification pour la détection des influenceurs. Une deuxième remarque concerne la détection des influenceurs positifs. En effet, nous avons aussi des taux de bonne classification qui vont jusqu'à 90%. Une dernière remarque par rapport aux influenceurs positifs qui influencent des utilisateurs positifs et les influenceurs positifs qui influencent des utilisateurs négatifs, nous avons noté que lorsque le nombre détecté d'influenceurs positifs qui influenceent des utilisateurs positifs augmente, le nombre détecté des influenceurs positifs qui influencent des utilisateurs négatifs diminue. Cette observation est dûe principalement au processus de la génération des données.

5 Classification de la propagation des messages sociaux

Le thème auquel les utilisateurs de réseaux sociaux et les influenceurs sont intéressés, est un paramètre important pour le problème de maximisation de l'influence. Ainsi, nous avons besoin de prédire le thème (la classe) des messages sociaux. A cet effet, nous introduisons une nouvelle approche de classification des messages sociaux qui utilise les traces de propagation des messages au lieu de leur contenu afin de prédire sa catégorie.

Tout d'abord, nous proposons un modèle de propagation de l'information qui tient compte de la classe du message à propager. Ce modèle est utilisé pour simuler les traces de propagation d'un type de message donné et pour créer un jeu de données des réseaux de propagation afin de l'utiliser dans les expérimentations de l'approche de classification.

Ensuite, nous introduisons deux classificateurs pour les messages sociaux qui utilisent des modèles de classification. Le premier classificateur se fonde sur la théorie des probabilités alors que le deuxième utilise la théorie des fonctions de croyance. Les classificateurs proposés ne nécessitent pas le contenu du message. De plus, tout type de contenu peut être classé. Nous avons juste besoin de ses traces de propagation et les types de liens traversés par le message. Ces classificateurs ont fait l'objet de l'article *Jendoubi et al.* [51]. Cette solution est plus adaptée pour les types de liens distincts, par exemple un type de lien peut être «amitié».

Dans le but de résoudre le problème des types des liens distincts présentés par les classificateurs qu'on a proposé, nous introduisons deux autres classificateurs qui se fondent sur la distance Dynamic Time Warping qu'on a adapté pour mesurer la distance entre les réseaux de propagation et on a appelé cette distance PrNeT-DTW. La distance proposée présente deux avantages. Elle fonctionne avec tout type de liens et elle considère le fait que les chemins dans le réseau de propagation sont dépendants du temps. Ensuite, nous avons utilisé la distance PrNeT-DTW avec les algorithmes k plus proche voisin probabiliste et crédibiliste pour classifier les traces de propagation des messages sociaux. Ce travail a été publié dans *Jendoubi et al.* [50].

Nous avons effectué un ensemble d'expérimentations à partir duquel nous pouvons conclure que les classificateurs de message social proposés sont utiles pour caractériser un message social donné sans avoir à accéder à son contenu. En effet, nous avons juste besoin des traces de propagation du message et des types des liens parcourus afin de déterminer sa catégorie. De plus les classificateurs proposés peuvent être triés en fonction de leur performance par rapport au bruit de la classification comme suit : le meilleur classificateur est le k plus proche voisin crédibiliste qui utilise PrNeT-DTW, après nous avons le k plus proche voisin probabiliste qui utilise PrNeT-DTW. Le suivant est le classificateur crédibiliste et le dernier est le classificateur probabiliste sans la distance. Une dernière contribution importante des classificateurs proposés est qu'ils sont adaptés à tout type de contenu propagé sur les réseaux sociaux, *i.e.* image, vidéo, etc.

6 Conclusion

Dans cette thèse, nous nous concentrons sur la proposition de nouvelles solutions pour le problème de la maximisation de l'influence afin d'améliorer l'efficacité d'une campagne de marketing viral. En fait, le marketing viral est une stratégie qui exploite l'effet de bouche à oreille et utilise les réseaux sociaux pour promouvoir un produit, une marque, etc.

Nous proposons une mesure crédibiliste d'influence et un ensemble de mesure d'influence qui tiennent en compte de l'opinion de l'utilisateur. De plus, nous introduisons deux modèles de maximisation de l'influence qui sont adaptés aux mesures proposées. Nous effectuons un ensemble d'expérimentations sur des données réelles et des données générées qui montre l'efficacité des mesures proposées.

Nous introduisons également une approche de classification des messages sociaux qui n'a pas besoin du contenu du message. En effet, on utilise les traces de propagation et les types des liens parcourues par le message afin de le classifier. Nous proposons ensuite, un ensemble d'expérimentations qui montre l'efficacité de l'approche proposée.

Dans ce qui suit, nous présentons quelques perspectives :

1. La généralisation de l'approche de la classification du message social en considérant le contenu. L'approche de la classification sera utilisée après avec les modèles de maximisation de l'influence.
2. La maximisation de l'influence dans les communautés sociales.
3. L'adaptation des modèles de maximisation de l'influence à d'autres réseaux sociaux comme Facebook et LinkedIn.

Contents

List of Tables

List of Figures

List of Algorithms

List of abbreviations

In the following, we present a list of abbreviations used in this document that is as exhaustive as possible.

Models and algorithms

- LTM: Linear Threshold Model
- ICM: Independent Cascade Model
- UN ICM: Independent cascade model with uniform edge probabilities
- TV ICM: ICM with trivalency edge probabilities
- GTM: General Threshold Model
- GCM: General Cascade Model
- WC, WC ICM: Weighted Cascade
- CD: Credit Distribution
- OC: Opinion-based Cascading model
- SI: Suspected-Infected epidemic model
- SIR: Suspected-Infected-Recovered epidemic model
- SIS: Suspected-Infected-Suspected epidemic model
- SIRS: Suspected-Infected-Recovered-Suspected epidemic model
- SPM: Shortest-Path Model
- WIC: Weighted Independent Cascade

- CELF: Cost-Effective Lazy Forward algorithm
- 1 Level: the first evidential influence maximization model that uses the formula (3.34)
- 2 Levels: the second evidential influence maximization model that uses the formula (3.35)
- k-NN: k nearest neighbors

Social network

- ONS: Online Social Network
- WoM: Word-of-Mouth
- eWoM: electronic word of mouth
- G: a graph or a directed graph
- V: a set of vertices
- T: a subset of V
- E: a set of directed links
- W: the set of weights vectors associated with each link in E
- u, v: vertices or nodes
- (u, v): a directed link where u is its source and v is its destination
- D_u: the overall degree of the vertex u
- $D_{IN}(u)$: the number of in-neighbors of u
- D_{min} : the minimum degree in the network
- D_{max}: the maximum degree in the network
- ϖ_u: the strength of u
- ϖ_{min}: the minimum strength in the network
- ϖ_{max}: the maximum strength in the network
- μ: the difference between D_{max} and D_{min} plus an $\varepsilon_1 \in [0, 1]$
- ν: the difference between ϖ_{min} and ϖ_{max} plus an $\varepsilon_2 \in [0, 1]$
- $w(u)$: the weight of the node u

- $w\left(u,v\right)$: the weight of the link (u,v)
- $N_a\left(u\right)$: the set of active neighbors of u
- $w_f\left(u,v\right)$: the follow weight
- $w_m\left(u,v\right)$: the mention weight
- $w_r\left(u,v\right)$: the retweet weight
- $w_x\left(u,v\right) \in \{w_f\left(u,v\right),\, w_r\left(u,v\right),\, w_m\left(u,v\right)\}$
- $w_x^{'}\left(u,v\right) \in \left\{w_f^{'}\left(u,v\right),\, w_r^{'}\left(u,v\right),\, w_m^{'}\left(u,v\right)\right\}$: updated link weights
- $N_{min_x} = \min_{u \in V} w_x\left(u\right)$
- $N_{max_x} = \max_{u \in V} w_x\left(u\right)$
- $\vartheta_x = N_{max_x} - N_{min_x} + \alpha$, $\alpha \in [0,1]$ and $\vartheta_x \in \{\vartheta_f,\, \vartheta_m,\, \vartheta_r\}$
- $L_{min_x} = \min_{(a,b) \in E} w_x^{'}\left(a,b\right)$
- $L_{max_x} = \max_{(a,b) \in E} w_x^{'}\left(a,b\right)$
- $\pi_x = L_{max_x} - L_{min_x} + \varepsilon$, $\varepsilon \in [0,1]$ and $\pi_x \in \{\pi_f,\, \pi_m,\, \pi_r\}$
- Sc_u: the set of successors of u
- Sc_{max}: the biggest successors set in the network
- Pc_u: the set of predecessors of u
- Tc_u: the set of tweets of u
- Tc_{max}: the biggest tweets set in the network
- $Rt_u\left(v\right)$: the set of tweets of u that were retweeted by v
- $Mt_u\left(v\right)$: the set of tweets of u in which v was mentioned
- Mt_{max}: the biggest $Mt_u\left(v\right)$ in the network
- Mt_u: the set of tweets in which u mentions another user.
- $c_b\left(v\right)$: the betweeness centrality measure
- $c_c\left(v\right)$: the closeness centrality measure
- $g\left(u,w\right)$: the number of geodesics in G
- $g'\left(u,v,w\right)$: the number of (u,w) geodesics in G containing v.

Information propagation

- θ_u: the LTM threshold, it is a random uniform value from the range $[0, 1]$
- $\rho\,(u, v)$: the ICM success probability of u to activate v
- f_u: the threshold function used by GTM
- Sus: the fraction of suspected individuals
- Ift: the fraction of infected individuals
- R: the fraction of recovered individuals
- ξ: the probability with which a suspected individual becomes infected
- ι: the average rate at which an infected individual becomes recovered
- ϵ: the average rate at which the individual loses his immunity
- Ac_u: the number of actions performed by the user u
- $Ac_{u\&v}$: the number of actions performed by u and v
- $Ac_{u|v}$: the number of actions performed by u or v
- Ac_{u2v}: the number of actions propagated from u to v.
- L : an action log that is defined as the set of tuples ($User$, $Action$, $Time$)
- a: an action
- $N^a_{out}\,(v)$: the set of v's active out-neighbors
- $N^{ia}_{out}\,(v)$: the set of v's inactive out-neighbors
- $att \in [-1,\, 1]$: the user's attitude to the product
- $lp \in \{-1,\, 1\}$: a link parameter
- PrNet: Propagation Network
- η: the number of iterations of the proposed information propagation algorithm
- $Source$: the source of the message, it is the first node that will trigger the propagation process
- $ReadyNodes$: a list that contains nodes having received the message and that will try to propagate it
- $LinkType$: the type of the link

- e: the number of node neighbors that will receive the message from a given type of link
- ψ: a matrix structure where in its lines we have the types of links, and in its columns we have the propagation levels
- $BbaSet$: a set of BBA distributions to represent each propagation level
- $ProbaSet$: a set of probability distributions to represent each propagation level
- $ProbaDist$: the vector of distances between probabilities distributions
- $BbaDist$: the vector of distances between BBAs distributions

Influence measures

- S: the seed set or the set of influencer users, $S \subseteq V$
- k: an integer, it is used to define the size of S or to define the number of considered nearest neighbors of k-NN classifier
- $\sigma_M(S)$: the objective function of the ICM and LTM
- $\gamma_{v,u}(a)$: a direct influence credit given to v for propagating the action a to u
- $\Gamma_{v,u}(a)$: the total influence credit given to v for propagating the action a to u
- $\Gamma_{S,u}(a)$: the total influence credit given to S for propagating the action a to u
- $\Delta(v,u)$: the total influence credit given to v to u
- $\Delta(S,u)$: the total influence credit given to S to u
- $\sigma_{CD}(S)$: the objective function of the CD model
- $PMG(v)$: the potential marginal gain of v
- $Op(v)$: the opinion indicator of v
- $u.mg1$: the marginal gain of u under the current S
- $u.prevBest$: the node with the maximum marginal gain in the current iteration that was examined before u
- $u.mg2$: the marginal gain of $u.prevBest$
- $u.flag$: the iteration number when $u.mg1$ was last updated
- Inf: the evidential influence of the user u on his neighbor v

- Inf_1^+: the evidential influence measure that considers the user's probabilistic positive opinion about the product
- $Inf_2^+(u,v)$: the evidential influence measure that considers the user's evidential positive opinion about the product
- $Inf_1^{++}(u,v)$: the evidential influence measure that considers the user's probabilistic positive opinion and the neighbors having a probabilistic positive opinion
- $Inf_2^{++}(u,v)$: the evidential influence measure that considers the user's evidential positive opinion and the neighbors having an evidential positive opinion
- $Inf_1^{+-}(u,v)$: the evidential influence measure that considers the user's probabilistic positive opinion and the neighbors having a probabilistic negative opinion
- $Inf_2^{+-}(u,v)$: the evidential influence measure that considers the user's evidential positive opinion and the neighbors having an evidential negative opinion
- $M1$ and $M2$: are the proposed first and the second maximization models respectively
- $\Phi_{M1}(S,v)$: the influence of S on a user v using $M1$
- $\Phi_{M2}(S,v)$: the influence of S on a user v using $M2$
- $\sigma_{M1}^{Bel}(S)$: the spread function under $M1$
- $\sigma_{M2}^{Bel}(S)$: the spread function under $M2$
- Q: sorted list in decreasing order according to the marginal gain of nodes
- $nodeMax$: a node having a maximum marginal gain

Uncertainty theories

- Ω, Θ: frames of discernment
- 2^{Ω}, 2^{Θ}: power sets
- I: influencer user
- P: passive user
- Pos: positive opinion
- Neg: negative opinion
- $Neut$: neutral opinion
- $high$: high influence

- *low*: low influence
- m_{D_u}: the degree BBA
- m_{ϖ_u}: the strength BBA
- $m^{\Omega}_{f_u}$: the follow BBA defined on the node u
- $m^{\Omega}_{m_u}$: the mention BBA defined on the node u
- $m^{\Omega}_{r_u}$: the retweet BBA defined on the node u
- $m^{\Omega}_{x_u} \in \left\{ m^{\Omega}_{f_u}, m^{\Omega}_{m_u}, m^{\Omega}_{r_u} \right\}$
- m^{Ω}_{u}: the combination result of $m^{\Omega}_{f_u}$, $m^{\Omega}_{m_u}$ and $m^{\Omega}_{r_u}$
- $BetP^{\Omega}_{u}$: the result of the pignistic transformation of m^{Ω}_{u}
- $m^{\Omega}_{f_{(u,v)}}$: the follow BBA defined on the link (u, v)
- $m^{\Omega}_{m_{(u,v)}}$: the mention BBA defined on the link (u, v)
- $m^{\Omega}_{r_{(u,v)}}$: the retweet BBA defined on the link (u, v)
- $m^{\Omega}_{x_{(u,v)}} \in \left\{ m^{\Omega}_{f_{(u,v)}}, m^{\Omega}_{m_{(u,v)}}, m^{\Omega}_{r_{(u,v)}} \right\}$
- $m^{\Omega}_{(u,v)}$: the combination result of $m^{\Omega}_{f_{(u,v)}}$, $m^{\Omega}_{m_{(u,v)}}$ and $m^{\Omega}_{r_{(u,v)}}$
- m^{Θ}_{u}: the BBA distribution defined on Θ that express the opinion of the user $u \in V$ about the product
- Pr^{Θ}_{u}: the probability distribution defined on Θ that express the opinion of the user $u \in V$ about the product
- $\{C_1, C_2, ..., C_n\}$: a set of n classes
- A and B: focal elements
- χ, $\kappa_i\,(d_j)$: decreasing functions that take the distance d_j as input
- $\gamma_i > 0$ and $\beta \in \{1, 2, \ldots\}$: parameters of the evidential k-NN
- $m^{\Omega}\,(A)$: the mass value assigned to the subset $A \subseteq \Omega$
- $bel^{\Omega}\,(A)$: the belief function defined on Ω
- $pl^{\Omega}\,(A)$: the plausibility function defined on Ω
- Pr^{Ω}: a probability distribution defined of Ω
- $\oplus$: Dempster's rule of combination
- CRC, $\circledast$: Conjunctive rule of combination

Distances

- $d_C\left(X_1, X_2\right)$: the Chebyshev distance
- $d_M\left(X_1, X_2\right)$: the Manhattan distance
- $d_E\left(X_1, X_2\right)$: the Euclidean distance
- $d_J\left(X_1, X_2\right)$: the Jousselme distance
- X_1, X_2: two vectors
- $\underset{=}{\Lambda}$ is a $2^n \times 2^n$ matrix and $\Lambda\left(A, B\right) = \frac{|A \cap B|}{|A \cup B|}$
- DTW: the Dynamic Time Warping distance
- $\delta\left(b1_i, b2_j\right)$: a distance between $b1_i$ and $b2_j$
- d_j: the distance between the object to be classified and the j^{th} nearest neighbor
- PrNet-DTW: the Propagation Network DTW distance

Other notations

- α, ε, ε_1, ε_2: numbers between [0, 1]
- i, j: counters
- t, t_1, t_2: time instant
- q: quality factor
- l: an object that we do not know its class
- $TS1 = (b1_1, b1_2, \ldots, b1_{T1})$, $TS2 = (b2_1, b2_2, \ldots, b2_{T2})$: two time series
- Ξ: a $|T1| \times |T2|$ matrix, used to estimate the DTW distance between $TS1$ and $TS2$

1

Introduction

The electronic Word of Mouth (WoM), called Viral Marketing, is a relatively new form of marketing communication that exploits the internet and more specifically online social networks in order to promote a product, a brand, etc. The Viral Marketing is based on the social influence that exerts one user on another. In fact, the idea behind it, is to target a small set of influencers that are able to trigger a large cascade of propagation and adoption of the marketing message. Thus, with a small marketing budget a large proportion of a social network can be reached or influenced [1].

Nowadays, there are 7.2 billion people in the world, 2.1 billion among them uses online social networks (ONS). Today, 90% of young adults (ages 18 to 29) and 35% of those aged over 65 use ONS[1]. Furthermore, many companies have recourse to social networks to promote their products and brands. In fact, 96% of small companies use social networks for marketing, and 92% of those adopt the phrase, "Social media marketing is important for my business"[2]. From these statistics, we can conclude that OSN have successfully reached many users worldwide, which make them more powerful in propagating any information. Also, these statistics show the importance of OSN for business. These facts make them very suitable for marketing and more specifically for Viral Marketing. Scientifically, the Viral Marketing problem is translated into the influence maximization problem that searches to select a small set of social network users that are able to maximize the global influence through the network.

The purpose of the influence maximization problem is to find a set of social influencers called seeds, those users have to be able to influence a large proportion of the network. For this purpose, many research works were conducted trying to resolve this NP-Hard problem [55]. The first optimization solutions were proposed in 2003 by *Kempe et al.* [55]. Indeed, they introduced the Independent Cascade Model (ICM) and the Linear Threshold Model

[1] Pick, T., 47 Superb Social Media Marketing Stats and Facts, http://www.business2community.com/social-media/47-superb-social-media-marketing-stats-facts-01431126#mL1oK4xCld7sLb6S.97, Posted on 19/01/2016, Seen on 24/09/2016.

[2] Delzio, S., 12 Social Media Marketing Trends for Small Business, http://www.socialmediaexaminer.com/social-media-marketing-trends-for-small-business/, Posted on 09/06/2015, Seen on 24/09/2016.

(LTM). These two models require only the network structure and are maximized through the greedy algorithm. Later in 2007 an amelioration of the maximization solution were proposed, it is about 700 times faster than the basic greedy solution. It is the Cost-Effective Lazy Forward algorithm (CELF) [60]. This is true that CELF has ameliorated the time spent to find the set of influencers. However, the quality of selected seeds stills not yet as good as needed. Then, this problem stills always unclosed.

To improve the quality of selected seeds, *Goyal et al.* [41] propose a new influence maximization solution that has a new spirit in its principle. In fact, they used past propagation to estimate the user's influence in the network and they introduced the Credit Distribution solution. Besides, the authors show the adaptability of their model to a greedy based approximation. Then, CELF algorithm is very adaptable to select a set of seeds having the maximum amount of influence credit. According to the experiments of *Goyal et al.* [41], their solution ameliorates the quality of selected seeds and the time spent to find them when compared to ICM and LTM. However, it is possible to ameliorate more the quality of the selected seeds. In fact, many other parameters can be considered for this purpose. Among these parameters we find the user's opinion that is crucial in the influence universe.

Existing influence maximization approaches assume only positive opinions influence among users and availability of positive influence probabilities. Whereas, a key function of social networks, besides sharing, is that they enable users to express their personal opinions about a product or trend of news by means of posts, shared posts, likes/dislikes, or comments on friend's posts, etc. Such opinions are propagated to other users and might make a significant influence on them, either positive or negative. For example, if some friends have shown any positive (or negative) comments against a certain product or news, one will have a similar feeling regardless of their own personal opinion. Consequently, the users opinion is an interesting parameter if we are looking for influencers.

Opinion-based influence maximization attracted many researchers in these last years. Let take the work of *Zhang et al.* [99] as an example. In fact, they propose a new influence maximization model that looks for positive influencers. Their model is called Opinion-based Cascading model. It is an extension of the Independent Cascade Model that considers the user's opinion. However, in this work all model parameters are randomly generated. In fact, according to the work of [41], influence maximization models that use randomly generated parameters in their input "can end up selecting seed sets of poor quality". Therefore, it is important to find a way to estimate the user's influence and opinion from available real world data. Other works in the literature considered the opinion in their process but it is not always the user's opinion about the product, it may be just about positive messages like ideas and news or negative messages like rumors [95]. As a consequence, proposing a new influence maximization solution that takes into account the user's opinion is very important.

Another interesting social networks analysis research field that is very related to the Viral Marketing is the problem of social message classification. In fact, the main purpose of

this problem is to find the set of topics to which a given user is interested to. In fact, the topic is an interesting parameter for the Viral Marketing and it is considered in many works in the literature like the work of *Barbieri et al.* [10]. However, existing text classification approaches are not always adaptable for social messages. Indeed, the social message is characterized by its shortness which leads to the lack of sufficient word occurrence problem. Besides, the social message is not an ordinary text, it may contain URLs, special characters, etc. All these characters are not considered by ordinary text classification techniques. In the literature, many works were conducted to resolve this problem [9, 47]. However, it still not yet resolved.

A crucial problem that can arise with real social networks data is about imprecision and uncertainty that is caused by many factors. In fact, social interactions can not always be precise and certain, also, online social networks allow only a limited access to their data. These facts are the sources of the imprecision and uncertainty for social networks data. These data imperfections can generate some problems to social networks analysis. Indeed, we may be confronted to obtain erroneous analysis results. In such a situation, the theory of belief functions [26, 79] is widely applied. Furthermore, this theory is used many times to handle such problems for analyzing social networks [94, 35, 51, 50, 100].

In these last years, many researchers are focusing on proposing new solutions to ameliorate the quality of selected seeds or to improve the running time of existing approaches. Despite their efforts, many issues still not yet processed and many improvement can, always, be done. In this thesis, we are interested to the Viral Marketing and especially we want to ameliorate the quality of selected seeds.

When studying the state of the art of the influence maximization and social messages classification problems, we become more motivated for our choice. In fact, we found that most of existing works use only the structure of the network to select seeds and such a model can select well located seeds. However, the position of the user in the network is not sufficient to confirm his influence. For example, he may be a user that was active in a period of time, then, he collected many connections, and now he is no more active. Hence, the user's activity is an interesting parameter that must be considered while looking for influencers.

Besides to the user's activity in the network, many other important influence behaviors are not considered. Among these behaviors, we found the sharing and tagging activities of network users. These activities allow the propagation of social messages from one user to another. Also, the tagging activity is a good indicator of the user's importance in the network. In fact, more he is tagged in others' posts more he is important for them. Therefore, taking into account such influence behaviors will be very beneficial to improve the quality of selected seeds.

Another crucial parameter in the influence maximization universe, that practically, was not considered, is the opinion of the user about the object of the Viral Marketing campaign,

i.e. we mean by the object here the product or the brand that we want to make viral. In fact, when we do not consider the user's opinion we may be confronted to harmful results of the marketing campaign especially when we fall on influencers having a negative opinion. Consequently, we find that the user's opinion is an interesting parameter that must be took into account to find appropriate influencers for a given Viral Marketing campaign. Besides, we use the theory of belief functions to represent all these influence aspects, also to combine them and manage the conflict that can arise between them.

An important problem to which we are also interested to, is the problem of classifying social messages. As we explained above, social messages have some special characteristics that differentiates them from ordinary text. Among these characteristics, we find the shortness of the message that leads to the topic sparcification problem. Consequently, it is important to find a new classification approach that resolves this problem. In fact, knowing the topics to which a social network user is interested to, is a very useful information for many applications among them the Viral Marketing.

Motivated by all these points, we conducted our thesis and we achieved many interesting contributions. Figure 1.1 illustrates the main contributions of this thesis in terms of inputs, steps and results. In a first place, we focused on the problem of measuring the influence on a social network. For this step, we chosen Twitter as an example and we explained the proposed measure using Twitter vocabulary. In fact, the proposed evidential influence measure considers many influence aspects like the user's position and his popularity in the network. Besides, we use the theory of belief functions [79] to represent each parameter using a basic belief assignment distribution and to combine all pieces of information in order to manage the conflict that can arise between them.

In a second place, we incorporate the user's opinion in the influence measure. For this purpose, we introduce three new Viral Marketing scenarios. The first scenario is about influencers having a positive opinion about the product. The second one concerns influencers having a positive opinion about the product and that exert more influence on users having a positive opinion too. The last scenario is about influencers having a positive opinion about the product and that influences users having a negative opinion. For each scenario of those we define two influence measures.

After defining the set of influence measures, we define an influence maximization model that works with them. For this purpose, we introduced two new influence maximization models. The first one considers the influence of a given user on his direct neighbors. In fact, it is ideal to detect influencers having many neighbors. Furthermore, this model is very useful in the case where we have a product with some quality issues [23]. The second model considers more indepth influence. Indeed, it maximizes the influence that exerts the user on his neighbors and his neighbor's neighbors. This model is very adaptable to maximize the Viral Marketing campaign.

To prove the performance of the proposed influence maximization solutions, we present

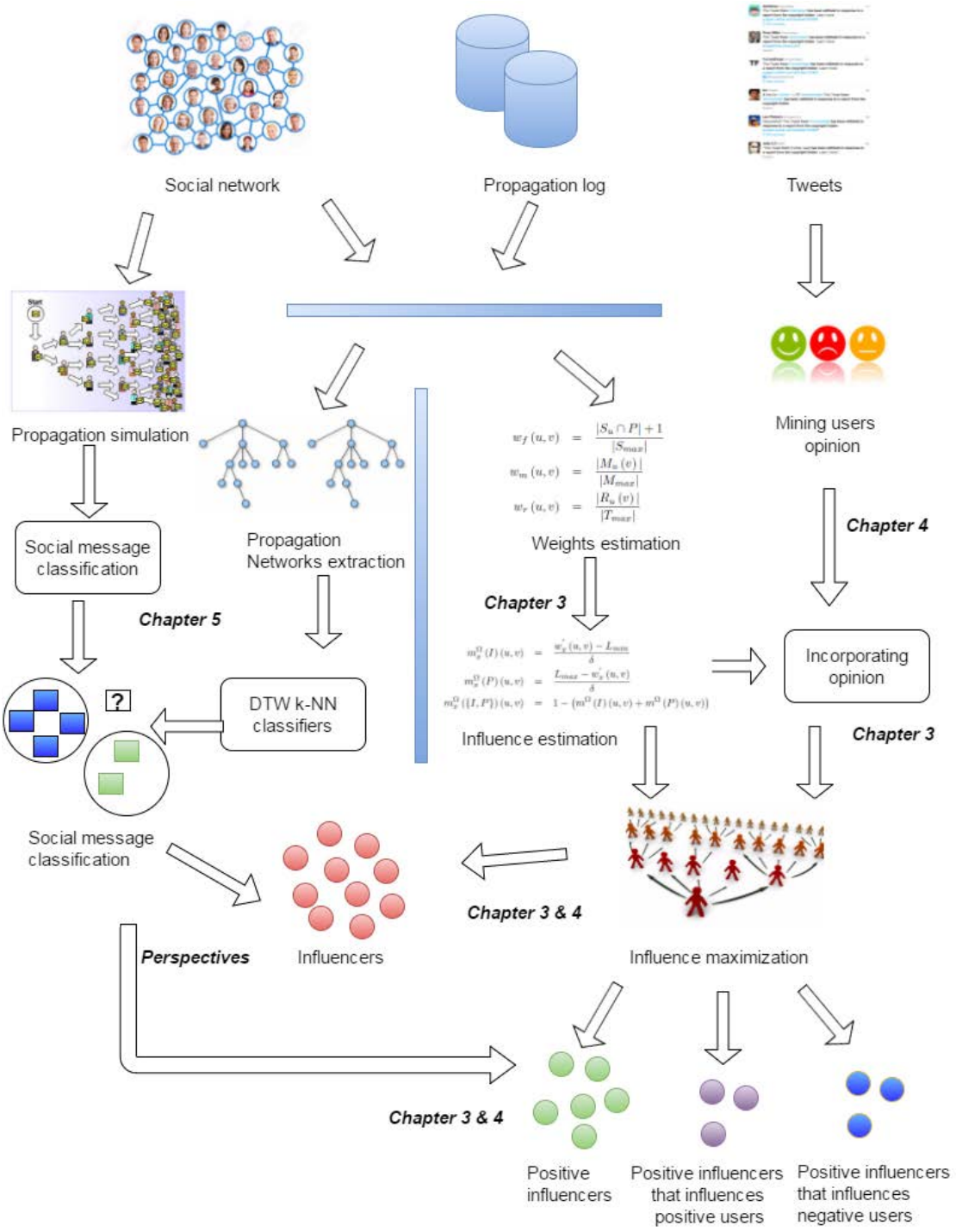


Figure 1.1: Characterization of social influencers

a case study on real world data. Its purpose is to maximize the influence for smartphones in Twitter. It is the first case study that details all the process of the influence maximization problem starting from data collect until getting seeds. Besides, we compare the quality of selected seeds using our models to detected ones using existing models.

The topic to which social network users and influencers are interested in, is also an important parameter for the influence maximization problem. Then we need to predict the topic (the class) of the social messages. For this purpose, we introduce a new classification approach that uses the propagation traces of the message in order to predict its class. Then we present four classifiers, two of them are model based and use a training set of previous propagation to learn their models parameters. The two others algorithms are distance based. In fact, we propose a new distance measure between two propagation networks that is based on the Dynamic Time Warping distance [75]. Then, we use the proposed distance with probabilistic and evidential k-Nearest Neighbors algorithms.

This thesis is organized in four chapters as follows:

- In Chapter 2, we review the state of the art of the information dissemination, the influence maximization and social message classification problems. In fact, these problems are among the popular research fields that are related to social network analysis. First, we discuss the basic information propagation models and their extensions. Besides, we introduce epidemic models that study the propagation of a given disease through a population. Next, we present existing influence measures and their properties. Then, we move to influence maximization approaches and we classify them into three main classes which are maximization models that uses an information diffusion model, data-based maximization models and opinion-based influence maximization models. Finally, we introduce some existing algorithms for social message classification.

- In Chapter 3, we present our contributions to the social influence universe. In fact, we present the process we propose to estimate the influence from social network data. Then, we introduce three opinion-based scenarios, the first one is about influencers having a positive opinion, the second one is about influencers having a positive opinion and influencing positive users and the last scenario searches for influencers having a positive opinion and influencing negative users. For each scenario we propose two influence measures. Next, we present two new influence maximization models that are useful with the proposed measures. Finally, we propound some examples to explain deeper the proposed approach.

- In Chapter 4, we conduct some experiments to prove the performance of the proposed influence maximization solutions. First, we introduce a case study in which we want to promote smartphones on Twitter. Then, we present the process we used to collect the dataset from Twitter and to estimate the user's opinion. Next, we compare our influence maximization models with some basic models like the Independent Cascade

Model and the Linear Threshold Model [55]. We compare, also, the proposed models to the Credit Distribution [41] that we consider the closest in its principle to our models. Furthermore, other experiments are done to study the impact of incorporating the user's opinion in the influence maximization process. On the other hand, we study the accuracy of the proposed solutions on generated data and we compare them to each other.

- In Chapter 5, we move to another important problem in the Viral Marketing universe, which is the problem of social message classification. First, we propose a new information propagation algorithm that considers the message class in its process, besides, this algorithm is useful to simulate the propagation of a given type of messages. Furthermore, we introduce a new social message classification approach that uses past propagation to learn a classification model. Next, the learned model is used to classify new coming messages. After that, we present a new distance metric useful to estimate the distance between two propagation networks. Then, we use the proposed distance with the probabilistic and the evidential k-Nearest Neighbors algorithms to classify propagation networks of social messages. Finally, a set of experiments is made to prove the performance of the proposed solutions and to compare them to each other.

Besides to these chapters, we have Chapter 6 that is dedicated to the conclusion and some perspectives of our work. Furthermore, we add the following three appendices:

- Appendix A introduces the theory of belief functions. Then we introduce some of its basic concepts like the basic belief assignment. Besides, we present an overview of the information fusion and decision making using the evidence theory.
- Appendix B defines some basic concepts from the graph theory that are very useful to understand this document. Next, it introduces some centrality measures.
- Appendix C presents the list of published articles.

2

Information diffusion and influence maximization

Contents

Summary

The information diffusion is the process with which the information propagates from one user to another through a social network, while the influence maximization process searches to select a small set of the social network users that are able to trigger an important diffusion cascade through the network. In this chapter, we review the state of the art of information diffusion models, influence measures and influence maximization in online social networks. Besides, we present some classification approaches for social messages that we find interesting and popular in this domain. Finally, we introduce the Viral Marketing which is an online marketing that uses influence maximization techniques to make products goes viral in a social network.

2.1 Introduction

In these last years, online social networks introduced some new functionalities that are appreciated and used by millions of people all over the world. Among these functionalities we find the sharing, sending messages, liking or commenting someone's post, etc. In return, online social networks collect every day a huge amount of data that may contain user's relationships, discussions, ideas, news, etc. Mining and analyzing such a data is always the challenge of many researchers, also, many interesting results were found. However, until today, many important problems are still not yet resolved. In this thesis, we are mainly interested in the simulation of the information propagation phenomenon, the influence maximization and the social message classification problems. In this chapter, we present the state of the art of these problems.

The information propagation is the phenomenon with which the information moves from one user to another through the network relationships. In the literature, there are many works that are interested in the study and/or the simulation of this phenomenon. Indeed, many works (like [25, 39, 38]) used to study the information propagation process in order to understand and explain this phenomenon, such a model is called explanatory model [44]. A second category of models called predictive models search to simulate the propagation traces in the network. In this category, we find, for instance, works of [43, 37, 55, 57]. In our work, we are interested in predictive models.

Another attractive research field, to which we are interested in, is the influence maximization problem. It is the problem of selecting a set of k users that are able to trigger a large propagation cascade through the network. This problem is shown to be NP-Hard [55]. In the literature, many solutions were designed for this problem. However, the quality of selected nodes, commonly called seeds, is not always assured. Indeed, most of the existent solutions use only the network structure, whereas, *Goyal et al.* [41] showed that these solutions are not efficient. In fact, the network structure is useful to detect well positioned users. However, those users may be inactive. Then comes the need for new influence maximization approaches that consider more data and more influence aspects like the user's behavior and past propagations. In this chapter, we present a state of the art overview of the information propagation simulation, the social message classification and the influence maximization problems.

A third important research field concerns the classification of social messages. We call social message all communications that can be done through an online social network, like sending a message, writing a comment, etc. The social message is generally characterized by its shortness, *e.g.* tweets are social messages of 140 characters. The short text is characterized by the lack of sufficient word occurrence. Such a characteristic, makes ordinary text classification techniques fail to classify short messages. In fact, these techniques use an existing word corpus to represent each possible text class, then, they try to mach the set of

words in the text to be classified with each class, and finally, they choose the class that fits more to the input text. However, these solutions are not efficient with short texts. Then, comes the need for new solutions.

The remainder of this chapter is organized as follows: Section 2.2, presents some existing models of information propagation and epidemic models that simulate diseases dissemination. Next, we present works that are related to the influence maximization problem. We divided this part into two sections: Section 2.3 that reviews influence measures and Section 2.4 that reviews maximization models. Finally, in Section 2.5, we move on to the social message classification problem and we introduce some works.

2.2 Information propagation in a social network

A diffusion model, also called propagation model, is a model that simulates and describes the entire propagation process and determines which node in the network will receive the propagated message [1]. In this section, we give an overview of information propagation models and how these models are used to simulate the information dissemination process.

2.2.1 Basic models and their extensions

The *Linear Threshold Model (LTM)* [43] and The *Independent Cascade Model (ICM)* [37] are among the first simulation models that were used to simulate the information propagation process. The *Linear Threshold Model (LTM)* was first proposed by *Granovetter* [43] to model collective behavior where one can trait two binary decisions, like the diffusion of rumors, diseases, innovations, etc. Then, LTM was used to model the information propagation process in social networks. The *Independent Cascade Model (ICM)* was introduced in the context of the marketing by *Goldenberg et al.* [37] drawing inspiration from works in interacting practical systems [64] and probability theory. LTM and ICM were adopted by *Kempe et al.* [55] to simulate the propagation of the information in social networks.

LTM and ICM are similar in that, in both of them we suppose having a social graph $G = (V, E)$ where its vertices can be either *active* or *inactive*. A vertex v is said to be active when it receives the information and accepts it. It is said to be inactive when it does not receive the information or rejects it. An inactive node becomes active if it receives and accepts the message. In the LTM, we associate a *weight* $\omega(u, v)$ to each edge (u, v) and a *threshold* θ_u to each vertex u. A vertex u will be activated if the total weight, between it and its activated neighbors, is at least θ_u:

$$\sum_{v} \omega(u, v) \geq \theta_u \tag{2.1}$$

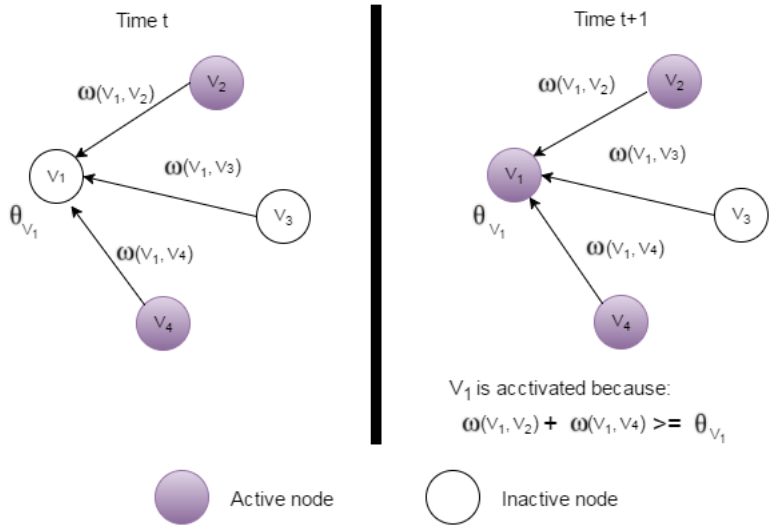


Figure 2.1: Linear Threshold Model example

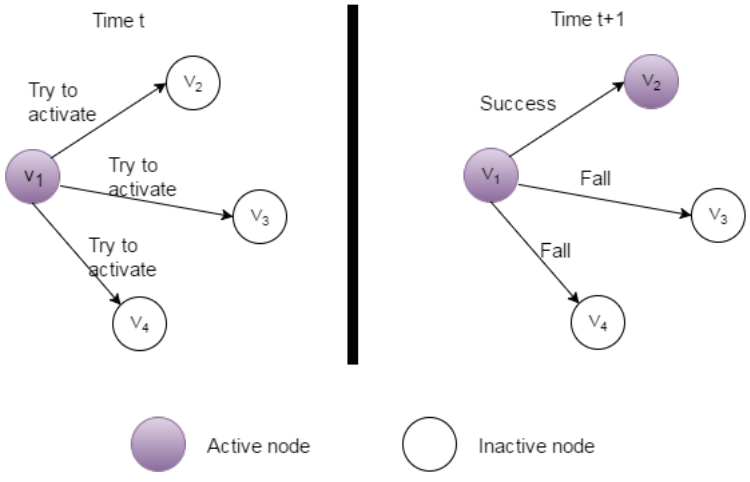


Figure 2.2: Independent Cascade Model example

The threshold θ_u is a random uniform variable chosen from $[0, 1]$, it "intuitively represents the different latent tendencies of nodes to adopt the innovation when their neighbors do" [55]. Next, we present a running example of LTM.

Example 1. Figure 2.1 presents a running example of the LTM. In this example, we have a social network with four nodes and three links between them, v_2 and v_4 are active at the time instant t. At time $t+1$, the node v_1 is activated because $\omega(v_1, v_2) + \omega(v_1, v_4) \geq \theta_{v_1}$.□

In the ICM each newly activated node is given only one chance to activate its inactive neighbors, *i.e.* an active node can send its message only one time for all its inactive neighbors. For instance, at the step t, a newly activated node u will try to activate its inactive neighbor v, the success probability of u to activate v is given by $\rho(u, v)$ (parameter of the system). A special case of ICM is *Weighted Cascade (WC)* where

$$\rho(u, v) = \frac{1}{D_u} \tag{2.2}$$

such that D_u is the overall degree of the vertex u (see the degree definition in the appendix B). Next, we present a running example of ICM.

Example 2. Figure 2.2 introduces a running example of the ICM. In this example, we have a social network with four nodes and three links. At the instant t, only the node v_1 is active and it has only one chance to try to activate its neighbors. At $t+1$, the node v_1 succeeds to activating its neighbor v_2. □

Kempe et al. [55, 57] introduced a broader framework that generalizes ICM and LTM. This "general framework has equivalent formulations in terms of thresholds and cascades" [55]. The *General Threshold Model (GTM)* generalizes LTM. As in LTM, it associates to each node u in the network a threshold θ_u. It differs from the LTM in that, it defines a monotone threshold function f_u for each node u that maps the set of active neighbors of u to the range [0, 1], such that $f_u(\emptyset) = 0$. The activation of u in the GTM depends on the following inequation:

$$f_u(N_a(u)) \geq \theta_u \tag{2.3}$$

that defines the activation condition, such that $N_a(u)$ is the set of active neighbors of u. The *General Cascade Model (GCM)* is a generalization of ICM. In GCM, the probability with which a user v succeeds in activating its neighbor u is defined by an incremental function $\rho_u(v, N_a(u))$ that maps the set of active neighbors already tried to activate u, to the range [0, 1]. Further propagation models are later proposed, like the *Decreasing Cascade Model* in which the activation probability is a decreasing function [56, 57]. All these models try to ameliorate the basic ICM and LTM and that with a more flexible ways for measuring the user's influence in the network. However, they still use random functions to estimate the influence.

2.2.2 Epidemic models

In the previous subsection we reviewed some basic information propagation models. This subsection is mainly dedicated for epidemic models. They are used to describe the transmission of infectious diseases, they model the diffusion process of a particular disease to understand the mechanism with which the disease spreads through the population to prevent and/or control its spreading. Recently, epidemic models have been also used to model the information propagation like rumors or news propagation.

The simplest version of epidemic models considers two states: *Suspected (S)* and *Infected (I)*, this model is called *SI model* (see Figure 2.3a for illustration). The suspected state means that the individual has not caught the disease yet but could catch it through a contact with another individual who did. An individual in the infected state if he has the disease and can transmit it to susceptible people who are in contact with him [73]. This is the simplest epidemic model, it was extended to be more appropriate to model a specific disease. In the SI model, once the individual catches the disease, he still infectious forever. However, someone can recover from the disease after a period of time and he may preserve its immunity to the disease and will not catch it again. To model such a case, we need a third state usually called *Recovered (R)*.

SIR model is an epidemic model that considers the three disease states *Suspected (S)*, *Infected (I)* and *Recovered (R)*, the reader can refer to Figure 2.3b for illustration. SIR model was first introduced by [58]. This epidemic model has two main steps, in the first step, the individual is suspected and may become infected if he has a contact with another infectious individual. This contact is assumed to happen according to a probability ξ. In the second step, the infected individual becomes recovered at an average rate ι. The following equations are defined for the SIR model:

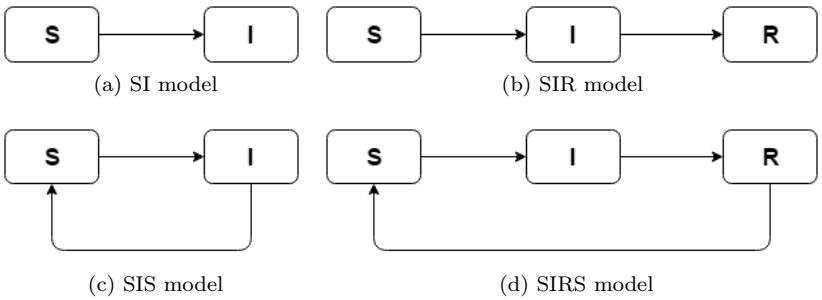
$$\frac{dSus}{dt} = -\xi.Sus.Ift \tag{2.4}$$

$$\frac{dIft}{dt} = \xi.Sus.Ift - \iota.Ift \tag{2.5}$$

$$\frac{dR}{dt} = \iota.If \tag{2.6}$$

where Sus is the fraction of suspected individuals, Ift is the fraction of infected individuals and R is the fraction of recovered individuals, such that $Sus + Ift + R = 1$.

Another extension of the SI model is defined for diseases where the individual can be infected many times. This case can arise with diseases that confer limited immunity or do not confer it at all. Such a model is called *SIS model* (see Figure 2.3c). It has two states as the SI model, however, infected individuals can return to the suspected state after recovery.



(a) SI model (b) SIR model

(c) SIS model (d) SIRS model

Figure 2.3: Epidemic models

Equations of the SIS model are defined as follows:

$$\frac{dSus}{dt} = \iota.Ift - \xi.Sus.Ift \tag{2.7}$$

$$\frac{dIif}{dt} = \xi.Sus.Ift - \iota.Ift \tag{2.8}$$

We present another epidemic model which is the SIRS model, the reader can refers to Figure 2.3d. In this model the individual recovers from the disease and confers immunity as in the SIR model. However, this immunity is not forever, after a period of time the individual loses it and returns suspected again. A third parameter is needed for this model which is ϵ, the average rate at which the individual loses his immunity. The equations of this model are:

$$\frac{dSus}{dt} = \epsilon.R - \xi.Sus.Ift \tag{2.9}$$

$$\frac{dIft}{dt} = \xi.Sus.Ift - \iota Ift \tag{2.10}$$

$$\frac{dR}{dt} = \iota.Ift - \epsilon.R \tag{2.11}$$

Epidemic models are very useful to study the spreading mechanism of a given disease through the population. In the case of the information propagation, the SIR model for example can corespond to the case where a set of initially infected users corresponds to a set of active users (recieved and accepted the information). Those infected (active) users try to infect (activate) their neighbors. An active user is recovered as he can not purchase again the product. According to *Leskovec et al.* [61] "The problem with these type of models is that they assume a known social network over which the diseases (the information) are spreading and usually a single parameter which specifies the infectiousness of the disease". In the context of the information propagation "this would mean that the whole population is equally susceptible" to receive the information, and this is not the case in the information propagation problem. The reader can refer to [2, 73] for further details.

2.3 Influence measures

In the previous section, we presented the state of the art of information propagation models. This section, reviews some of the existing influence measures in the literature that we find interesting and having a relation with the proposed measures in this document. We are mainly interested in influence measures that are adapted to Twitter and those that uses the theory of belief functions. Besides, we present other existing measures that we find interesting. The remainder of this section is organized as follows: first, we present influence measures that were proposed for Twitter, second, we introduce evidential influence measures. Finally, we focus on other influence measures.

2.3.1 Measuring influence on Twitter

Nowadays, there exists a lot of online social networks like Twitter, Facebook, LinkedIn, etc. They propose several services to their users like sending messages, sharing images, videos, etc. Each social network has its specific notions and characteristics. For example, on Twitter we have follow relationships, tweets, retweets, etc. On Facebook, we have friendship relations, status, share, etc. These specific characteristics require to be considered while measuring influence. In this work, we choose Twitter, as did many works in the literature. In fact, it is easier to get data from Twitter, because it provides a documented API with many toolboxes adapted to the most programming languages. In this section, we first introduce Twitter and its notions then we review some of the existing works that try either to study the influence on Twitter or to measure it.

2.3.1.1 What is Twitter?

Actually, Twitter is one of the most popular social networking *microblogging service*. In this section, we define some common concepts that one need to know about Twitter and that are used in this document.

Definition 1. *A Follow* is an explicit relation that allows a given user to follow updates from other users he follows. The follow relationship can either be reciprocated or one way.

Example 3. Let u and v be two Twitter users, then, if u is interested in updates from v, u can simply "follow" v and it will receive all the messages (called tweets) from v in its actuality timeline. □

Definition 2. *A Tweet* is a short message of 140-characters. Twitter allows its users to publish tweets in order to share their new, for example, with others.

Definition 3. *The Mention* functionality allows for a given Twitter user to send tweets directly to other users by mentioning their usernames prefixed with an "@" sign. Then, the tweet will appear in the timeline of all mentioned users.

Definition 4. *A Retweet* is a Twitter functionality that allows the sharing of tweets from each other. In fact, when someone retweets a tweet, it will appear to his followers. A retweet is always prefixed by an "RT @" plus the username of the user that wrote it.

2.3.1.2 Estimating influence on Twitter

In the literature, influence on Twitter was widely studied. However, most of the existing works either try to study Twitter users' behavior or use existing Twitter metrics like indegree and the number of mentions to identify influencers. In this section, we present some of the existing measures that we find relevant.

The work of *Cha et al.* [21] was among the first research works that studied the influence on Twitter. The authors present an empirical comparison of three basic influence indicators which are indegree (follow), retweets and mentions. They found that "indergree represents the popularity of a user, retweet represents the content value of one's tweets and mention represents the name value of a user". In a similar study, *Dubois and Gaffney* [32] compare six influence metrics (like indegree, eigenvector centrality and clustering coefficient) that are commonly used to identify influential users on Twitter. Besides, they identified the characteristics of top influencers that are selected by each measure.

Authors in [17] measure the user influence on Twitter using the k-Shell decomposition algorithm that determines the core and hierarchical structure of a given network. Then, the algorithm gives to each set of nodes a k value that is equal to their level. The authors used k-Shell decomposition algorithm for measuring influence. In fact, they interpret the k level of each node as its influence value. In the work of [17], authors modify the basic algorithm to assign to each user a logarithmic k-Shell influence value.

Ben Jabeur et al. [13] define Twitter influencers as "active actors who have the ability to spread information and inspire other people in the network". According to their definition, an influencer on Twitter is a user that is able to gain many reweets for his published tweets. Then, they propose InfRank algorithm to rank Twitter users by their opportunity to be retweeted. Similarly the work of *Sung et al.* [86] proposes an interaction ranking measure, called InterRank, that improves the PageRank measure by considering not only the follower relationship of the network but also the topical similarity between Twitter users.

The work of [76] study the influence of the "information value" of the tweet (content criteria) and the "agent awareness" (context criteria) on the retweeting decision. They found that both the content and the context criteria of the tweet leads to its retweeting. The work of *Azaza et al.* [6] introduces an influence measure for Twitter users. We will detail this work in the next section as they used the theory of belief functions to identify influencers.

2.3.2 Evidential influence measures

In the literature, there are some recent works that use the theory of belief functions to model the uncertainty while measuring the user's influence in online social networks. Appendix A is an overview of the theory of belief functions and its basic concepts. Also, Appendix B details the used graph theory concepts. In this section, we present a review of evidence theory-based works that we find close to our work.

The work of *Wei et al.* [94] was among the first works that use the theory of belief functions to estimate the user's influence in social networks. This work presents an evidential centrality (EVC) measure that combines "the degree and strength of every node in a weighted network" and tries to find a trade off between them. The method starts, first, by estimating two BBA distributions for each user, u, on the frame $\{high, low\}$ where $high$ for high influence and low for low influence. The first BBA to represent the degree, m_{D_u}, and the second BBA for the strength, m_{ϖ_u}, as follows:

$$m_{D_u}(high) = \frac{D_u - D_{min}}{\mu} \tag{2.12}$$

$$m_{D_u}(low) = \frac{D_{max} - D_u}{\mu} \tag{2.13}$$

$$m_{\varpi_u}(high) = \frac{\varpi_u - \varpi_{min}}{\nu} \tag{2.14}$$

$$m_{\varpi_u}(low) = \frac{\varpi_{max} - \varpi_u}{\nu} \tag{2.15}$$

where D_u, D_{min} and D_{max} are, respectively, the degree centrality of u, the minimum degree and the maximum degree in the network. Besides, ϖ_u, ϖ_{min} and ϖ_{max} are, respectively, the strength of u, the minimum strength and the maximum strength in the network. Finally, $\mu = D_{max} - D_{min} + \varepsilon_1$, $\varepsilon_1 \in [0, 1]$ and $\nu = \varpi_{max} - \varpi_{min} + \varepsilon_2$, $\varepsilon_2 \in [0, 1]$. In a second step, m_{D_u} and m_{ϖ_u} are combined using the Dempster's rule of combination:

$$m_u = m_{D_u} \oplus m_{\varpi_u} \tag{2.16}$$

Finally, the evidential centrality of u is defined by the BBA m_u (more details about centrality measures can be found in appendix B). An extension of this measure was, later proposed by *Gao et al.* [35]. In fact, they propose a centrality measure with a similar spirit as EVC. They modified the EVC measure according to the actual degree of the node instead of following the uniform distribution, also, they extended the semi-local centrality measure [22] to be used with weighted networks. Their centrality measure is the result of the combination of the modified EVC and the modified semi-local centrality measure. The work of [35] is similar to the work of [94] in that, they used the same frame of discernment, their approaches are structure based, and they choose the influential nodes to be top-1 ranked nodes according to the proposed centrality measure.

Azaza et al. [6] introduces an influence measure for Twitter. They consider three

different types of relationships on Twitter: Retweet, Mention and Reply. For each of these they estimated a BBA distribution defined on the frame Ω ={Very weak, Weak, Average Enough, Average, Strong Enough, Strong, Very Strong, Extremely Strong}. Also, they presented a modified version of the conjunctive combination rule [81] in which they take into account a subset of 2^{Ω}. Next, they used the defined combination rule to combine their BBAs on a preselected subset of 2^{Ω}. Finally, the user's influence class (Very weak, Weak, etc) is obtained from the prignistic probability that results from the pignistic transformation of the combined BBA. The purpose of the work of [6] was to classify Twitter users according to their influence class and not to quantify their influence.

2.3.3 Other influence measures

In the literature, there exist many ways to quantify the user's influence in a social network. We may find measures that need only the structure of the network. Those, generally, work with most social networks. Other measures estimate the influence using some characteristics of the online social network, we can take for example, the measures presented in section 2.3.1. Those measures are adaptable for similar social networks, *i.e.* we can find some analogy between them. A third category of measures tries to model the user's past propagations in the network. In this section, we give an overview of some interesting influence measures in the literature.

Among the first influence measures that were introduced in the literature, we find the measures that use a propagation model to simulate the propagation process of messages in the network. The influence is defined as the number of users that have received the message at the end of the propagation process. Such an influence measures were used, first, by *Kempe et al.* [55].

Goyal et al. [40] introduced many methods that can be used to learn influence probabilities from past propagation. In their paper they consider the static case, the continuous time case and the discrete time case. In this document, we consider the static case, for more details the reader can refer to [40]. Let Ac_u be the number of actions performed by the user u, $Ac_{u\&v}$ the number of actions performed by u and v, $Ac_{u|v}$ the number of actions performed by u or v and Ac_{u2v} the number of actions propagated from u to v. An action here may be posting a message for example. We choose to present two static models which are the Bernoulli distribution and the Jaccard index. In the Bernoulli distribution, the authors interpreted each propagation (successful attempt) as Bernoulli trial and they compute the maximum likelihood of success probability as:

$$\rho\left(u,v\right)=\frac{Ac_{u2v}}{Ac_u} \tag{2.17}$$

They also adapted the Jaccard index to estimate the influence probability as follow:

$$\rho(u, v) = \frac{Ac_{u2v}}{Ac_{u|v}} \tag{2.18}$$

In the next section, we move on to the problem of influence maximization, and we present some of the existing influence maximization models that we find interesting.

2.4 Influence maximization

Measuring user's influence is useful to identify influencer users in a social network. However, the influence maximization problem consists of choosing from influencer users those that are able to influence the maximum proportion of users in the network. To resolve this problem we have two main challenges, the first one is about the estimation of the user's influence in the network and the second challenge consists in finding the set of users that maximize the influence.

In previous sections, we reviewed some existing works on information dissemination and measuring influence in online social networks. In this section, we survey many influence maximization models. First, we introduce influence maximization models that use a diffusion model in their process. Second, we present data-based maximization models. Then, we focus on maximization models that consider the opinion in their process. Next, we introduce the used maximization algorithms in the literature. After that, we talk about an interesting application of the influence maximization problem which is the Viral Marketing. Finally, we present a discussion of the state of the art of this problem.

2.4.1 Diffusion models-based influence maximization

In this section, we present influence maximization models that use an information propagation model in their maximization process (more details about propagation models can be found in section 2.2). *Domingos and Richardson* [30] were the first to introduce the problem of identifying influencers for a marketing campaign as a learning problem. They modeled the customer's network value, *i.e.* "the expected profit from sales to other customers he may influence to buy, the customers those may influence, and so on recursively" [30]. Furthermore, they modeled the market as a social network of customers. Later in 2003, *Kempe et al.* [55] formulated the influence problem as an optimization problem. Also, they proved the NP-Hardness of their models. Besides, they assumed that they have the social network, and influence probabilities extent to which each individual influence one another. Their issue is to find/choose a set of influential individuals that maximizes the spread of the marketing message within the network.

Given a social network $G = (V, E)$, V is a set of vertices, E is a set of edges and a diffusion model M, the influence maximization (IM) problem is to select a set S of k influential users (called seed set) that maximizes the awareness of the "product" over the social network G [55]. In other words, it is the problem of choosing S seed nodes that maximize the expected number of influenced nodes, $\sigma_M(S)$. To estimate $\sigma_M(S)$, *Kempe et al.* [55] propose the use of the classical propagation models ICM and LTM (refer to section 2.2.1 for more details about ICM and LTM). Besides, they prove that maximizing σ_M is a NP-Hard problem, also, σ_M is monotone and submodular, the reader can refer to section 2.4.4 for more details about useful maximization models in such a case. To maximize σ_M and extract seed nodes, the authors used the greedy algorithm with the Monte Carlo simulation.

In the literature, many works were conducted to improve the running time when considering ICM and LTM. *Leskovec et al.* [60] introduced the *Cost Effective Lazy Forward* (CELF) algorithm. CELF is a greedy based solution that exploits the submodularity property of the function to be maximized. It is proved to be 700 times faster than the solution of [55]. More details about CELF algorithm can be found in section 2.4.4. *Kimura and Saito* [59] proposed the *Shortest-Path Model (SPM)* which is a special case of the ICM. In SPM, shortest paths are considered in the activation process. In fact, an inactive node u have the chance to be activated only through the shortest path from the seed set. *Bozorgi et al.* [16] considered the community structure, *i.e.* a community is a set of social network users that are connected more densely to each other than to other users from other communities [101], in the influence maximization problem. In fact, *Bozorgi et al.* [16] used the LTM to find the influencers within each community. Those are called local influencers. Next, they estimate the global influence, *i.e.* on the whole network, of those users using LTM. The influence of a given node is a combination between its local and global influence. Finally, they select a set of influencers that maximizes the influence in the network.

We also find other extensions of the basic models that search to improve the quality of the selected influencers or that consider some other important parameters. *Wang et al. [92]* introduced the Weighted Independent Cascade (WIC) model, which is an extension of ICM that considers attributes on the nodes of the network. The main purpose of the WIC model is to maximize the value of the influenced nodes. The values of the nodes are defined by their attributes. These attributes may model the ability of a given user to buy the product.

2.4.2 Data-based influence maximization

A data-based model is a model that uses social network data in addition to the network structure. The social network data may contain user profiles, messages, past propagation, etc. Works of [40] and [41] propose to use past propagation to learn their models. Besides the network structure, $G = (V, E)$, they used an action log L that is defined as the set of tuples $(User(u), Action(a), Time(t))$ such that $(u, a, t) \in L$ means that the user u

performed the action a at time t [41]. Past propagation is extracted from the action log L. Indeed, if we have $(u, a, t_1) \in L$, $(v, a, t_2) \in L$, $t_1 < t_2$ and $(u, v) \in E$, we say that the action a propagates from u to v. The work of [40] was detailed in section 3.3.

The *Credit Distribution* (CD) [41] is, also, a data-based model that investigates past propagation to detect influencers. It uses past propagation actions to associate an influence credit to each user in the network. The reader can refer to Figure 2.4 for illustration. Indeed, the figure shows that CD takes two main inputs which are the network structure and propagation log. It uses these inputs to estimate the influence. In fact, the influence spread is defined as the total influence credit given to a set of users S from the whole network. The idea behind this algorithm is when an action a propagates from a user v to a user u, a direct influence credit $\gamma_{v,u}(a)$ is given to v. Also, a credit amount is given to the predecessors of v in the propagation graph. The total credit of a user v is defined as follows:

$$\Gamma_{v,u}(a) = \sum_{u_1 \in D_{in}(u,a)} \Gamma_{v,u_1}(a)\, \gamma_{u_1,u}(a) \tag{2.19}$$

where $\Gamma_{v,v}(a) = 1$ and $D_{in}(u, a)$ is the set of in-neighbors of u that performed the action a. They also defined the credit given to a set of nodes as:

$$\Gamma_{S,u}(a) = \begin{cases} 1 & if\ u \in S \\ \sum_{u_1 \in D_{in}(u,a)} \Gamma_{S,u_1}(a)\, \gamma_{u_1,u}(a) & otherwise \end{cases} \tag{2.20}$$

The total influence credit given to v by u and the total influence credit given to S by u are defined, respectively, as follows:

$$\Delta(v, u) = \frac{1}{|A_u|} \sum_{a \in A_u} \Gamma_{v,u}(a) \tag{2.21}$$

$$\Delta(S, u) = \frac{1}{|A_u|} \sum_{a \in A_u} \Gamma_{S,u}(a) \tag{2.22}$$

such that A_u is the set of actions made by u. Finally, the total influence credit given to S is defined as:

$$\sigma_{CD}(S) = \sum_{u \in V} \Delta(S, u) \tag{2.23}$$

Authors [41] demonstrated that σ_{CD} is monotone and sub-modular, then, it can be maximized using the CELF algorithm. Also, they proved that the marginal gain of a given node u with respect to S is:

$$\sigma_{CD}(S + u) - \sigma_{CD}(S) = \sum_{a \in A} \left((1 - \Gamma_{S,u}(a)) \sum_{v \in V} \frac{1}{A_u} \Gamma_{v,u}^{V-S}(a) \right) \tag{2.24}$$

The first step of the credit distribution algorithm consists of scanning the action log L

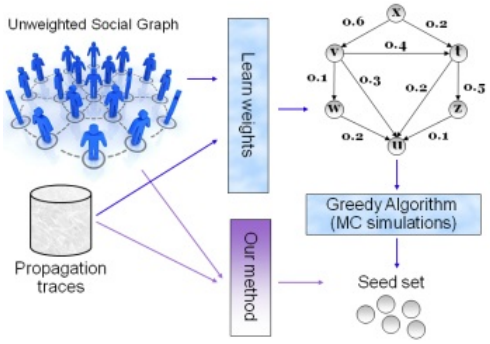


Figure 2.4: Credit distribution model [41]

to compute $\Gamma_{v,u}(a)$. The set of seed nodes, S, is initialized to $\emptyset$. In the second step, CD runs up the CELF algorithm to select a node with the maximum marginal gain (eq. (2.24)). Then, the algorithm updates $\Gamma_{S,u}(a)$ and $\Gamma_{v,u}^{V-S}(a)$. Next, it loops the second step until getting all needed seed nodes. For more details the reader can refer to [41].

The works presented above used the social network data to estimate the user's influence in the network. However, they omitted some characteristics that can be critical for the influence maximization problem. Among these characteristics the topic of the message and the trust or distrust between network users. *Barbieri et al.* [10] proposed topic-aware influence propagation models. First, they extended ICM and LTM to consider the topic of the message in the propagation process. Then, they ameliorated these models by considering the user's authoritativeness and interest in the topic instead of the user to user influence (such the case of IC and LT models). Later, *Aslay et al.* [5] introduced INFLEX: a query based system that gives a seed set in the social network for a given topic. Another characteristic was considered which is the trust between users. In an interesting work, [1] proposed a new diffusion model called Trust-General Threshold (TGT) model in which they considered the trust and the distrust probabilities that were defined on the relationships between network users. Besides, they introduced an influence maximization algorithm to select a seed set from the trust network. The work of [70] proposed an extension of ICM that considers trust and time factors in the maximization process.

2.4.3 Opinion-based influence maximization models

In sections 2.4.1 and 2.4.2, we reviewed two categories of influence maximization models, the first one, only uses the network structure to predict the user's influence while the second category of models uses past propagation for the same purpose. In this section, we are mainly interested in works that incorporate the user's opinion in the influence maximization process which is a relatively new idea. In fact, the user's opinion is a critical factor in marketing and social science. In the social psychology literature, the concept of positive-negative opinion asymmetry was largely studied [87, 11]. These works agreed on the fact that negativity

(negative events, ideas, news, etc) is always stronger than positivity (positive events, ideas, news, etc). This fact was, also, shown in marketing science like the work of *Cheung and lee* [24] that studied the impact of the negative electronic word of mouth (eWoM) on online shops and they found that "negative eWoM has a significantly larger impact on consumer trust and intention to the online shop". These works prove the importance of the opinion in the influence maximization process and especially the importance of selected seeds opinion.

The work of *Chen et al.* [23] was among the first works to incorporate the propagation of negative opinion in ICM [55]. They said that the negative opinion is more contagious than positive one, especially in people's decisions. Also, they defined an interesting parameter that models the quality of the product, called quality factor q, this parameter is used to detect when a seed node turns from having a negative opinion to a positive one or the inverse according to $(1-q)$. Similarly, the study of [95] was one of the first attempts to focus, not only, on one type of information, but also, on simultaneous spread of negative (like rumors) and positive (like ideas and news) information. However, they did not consider the user's opinion towards the propagated product. Furthermore, the work of [98] studied the problem of minimizing the influence of negative information (like rumors). Their main idea was to detect a set of influencers that maximizes the spread of negative information and to block them in order to minimize their influence.

On an interesting work, *Zhang et al.* [99] proposed the *Opinion-based Cascading (OC)* model that takes positive opinions of users into consideration. They used the OC model to maximize the positive influence by taking into account the user's opinion and the change of the opinion. They showed that the objective function of the OC model is no longer submodular. Besides, they proved the NP-Hardness of their model. Then, they proposed an approximation of the maximization results in a polynomial time. In a first step, OC ignores all users that have a small potential marginal gain that is defined as:

$$\begin{aligned} PMG(v) &= Op(v) + \sum_{u \in N^{a}_{out}(v)} \left(Op(u) + Op(v).w(v,u)\right) \\ &+ \sum_{u \in N^{ia}_{out}(v)} \frac{w(v,u)}{\theta_u} \left(Op(u) + Op(v).w(v,u)\right) \end{aligned} \tag{2.25}$$

where $Op(v)$ defines the opinion indicator of v such that:

- $Op(v) = 0$ means that v has a neutral opinion,
- $Op(v) > 0$ indicates that the opinion is positive,
- $Op(v) < 0$ the opinion of v is negative.

The sets $N^{a}_{out}(v)$ and $N^{ia}_{out}(v)$ are respectively, the sets of v's active and inactive out-neighbors. The parameter θ_u defines the activation probability of u. Finally, $w(v,u)$ is the weight associated to the edge (v,u). In the next step, OC iterates until getting k

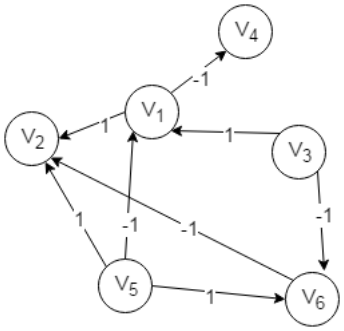


Figure 2.5: Signed social network example

seed nodes. In each iteration, the algorithm updates the activation status according to the following condition:

$$\sum_{u \in N_{in}^{a}(v)} w\left(u,v\right) \geq \theta_{v} \tag{2.26}$$

where $N_{in}^{a}\left(v\right)$ is the set of active in-neighbors of v. Also, it updates the opinion value of each user according to his previously activated neighbors as:

$$Op\left(v\right) = Op\left(v\right) + \sum_{u \in N_{in}^{a}(v)} \left(Op\left(u\right).w\left(u,v\right)\right) \tag{2.27}$$

Then, it chooses the user that stills in the top of the potential list. *Li et al.* [63] considered not only the friendship relations, but also foe relations in the influence maximization problem. They extended the IC model of *Kempe et al.* [55] and proposed a *Polarity-related Independent Cascade (P-IC)* model. P-IC is useful to select seeds with maximum positive influence or maximum negative influence. P-IC works with a signed social network in which we find positive relations (1) to model friendship or trust and negative relations (-1) to model foe or distrust.

Example 4. Figure 2.5 shows an example of a signed social network. In this figure, we have v_3 trusts v_1 and distrusts v_6. □

Similarly, *Wang et al.* [91] mined positive influencer form signed social networks. They proposed an extension of the Linear Threshold model of *Kempe et al.* [55] by incorporating two parameters which are the user's attitude to the product, $att \in [-1, 1]$, and a link parameter, $lp \in \{-1, 1\}$, that indicates if it is a positive or negative relationship.

All of these recent works assumed that positive and negative influence probabilities are known and given to the influence maximization algorithm as input. This is, obviously, not the case in real-world social networks. Therfore, some preprocessing is needed to close the gap between the model and the real data [41].

2.4.4 Maximization algorithms

In the influence maximization process, we have two main issues. The first one is: how to estimate the user influence in the network? Some interesting solutions for this problem are presented in previous sections. The second issue is: how to select the set of users that maximizes the influence? In the literature, this last problem is shown to be NP-Hard. The reader can refer to [55] and [41] for some examples of NP-Hardness proofs. In the state of the art, we often found a common solution for this problem. In fact, this solution consists of proving some specific characteristics of the objective function to be maximized and then they use an adaptable optimization algorithm. In this section, we present the popular greedy-based optimization solutions that performs a $(1 - 1/e)$-approximation to the optimal solution [72].

To use a greedy-based solution, the objective function σ has to be submodular and monotone set function that is defined from the power set 2^V to $\mathbb{R}$ where V is the set of network nodes and 2^V is the set of all subsets of V. The function σ is said to be submodular if it satisfies a *"diminishing returns"* property which means that the marginal gain of adding an element x to an input set S is at least as high as adding the same element to a superset T of S as follows:

$$\sigma(S \cup \{x\}) - \sigma(S) \geq \sigma(T \cup \{x\}) - \sigma(T) \tag{2.28}$$

whenever $S \subseteq T \subseteq V$ and $x \in V$. Besides, σ is said to be monotonic increasing function if

$$\sigma(S) \leq \sigma(T) \tag{2.29}$$

whenever $S \subseteq T \subseteq V$. A greedy-based solution can be adapted to maximize any monotone submodular set function σ that has $\sigma(\emptyset) = 0$. To maximize the influence in a social network, the purpose is to select a set S of k influencer users that are able to trigger a large cascade of adoption through the network. *Kempe et al.* [55] were the first to use the greedy algorithm for influence maximization. The greedy algorithm (Algorithm 1) is very simple in its principle. At each step, it estimates the marginal gain of each node, $x \in V$, with respect to S. The marginal gain is defined as the gain in influence of a given node with respect to the current S. Then, it chooses the node that has the maximum marginal gain until having k nodes in S. The marginal gain of nodes is estimated using ICM or LTM in the work of [55].

Another interesting greedy-based maximization solution was, later, proposed by *Leskovec et al.* [60]. They introduced the *Cost-Effective Lazy Forward* algorithm (CELF). CELF (algorithm 2) exploits the submodularity property of the objective function to minimize the number of calls of the marginal gain function. In fact, submodularity guarantees that the marginal gain decreases with the solution size. Then, instead of estimating it for each expected node at each iteration as the basic greedy algorithm do, CELF computes the marginal gain, for all nodes, in the first iteration and keeps an ordered list of them according to their

Algorithm 1: Greedy algorithm

```
begin
    S = ∅;
    // S is the set of seed nodes
    while | S |≤ k do
        u ← argmax_{x∈V/S} marginalGain(x);
        S ← S ∪ {u};
```

marginal gain for next iterations. In the next iteration, the algorithm pulls off the top node in the list (that has the current maximum marginal gain) and re-estimates its marginal gain, next, if the top node maintains its position in the list (still in the top), then it will be chosen and added to S, otherwise CELF re-evaluates the marginal benefit for the new top node and so on. This algorithm is up to 700 time faster than the basic algorithm [60] and it gives the same approximation guarantee.

An amelioration of CELF was, later, proposed by *Goyal et al.* [42], called CELF++. This extension reduces again the number of calls of the marginal gain function. Then, CELF++ improves the efficiency of the CELF algorithm by about 35%-55%. The idea behind this algorithm is that it maintains for each node a tuple of the form $(u.mg1; u.prevBest; u.mg2; u.flag)$ where $u.mg1$ is the marginal gain of u under the current S, $u.prevBest$ is the node with the maximum marginal gain in the current iteration that was examined before u, $u.mg2$ is the marginal gain of $u.prevBest$ and $u.flag$ is the iteration number when $u.mg1$ was last updated. At each iteration of the algorithm, if the node $u.prevBest$ is chosen as a seed node in the current iteration then there is no need for estimating the marginal gain of u in the next iteration.

Algorithm 2: CELF algorithm

```
begin
    S = ∅;
    // S: the set of seed nodes
    Q = ∅;
    // Q: sorted list of nodes in decreasing order according to their
       marginal gain
    foreach u ∈ V do
        marginalGain(u);
        // a function that estimates the marginal gain of u with respect to S
        Q.add(u);
    while | S |≤ k do
        v ← Q.pop();
        marginalGain(v);
        if v.MG ≥ Q.getFirst().MG then S.add(v);
        else Q.add(v);
```

2.4.5 Influence maximization application: Viral Marketing

With the ever-increase of social networks, marketers have turned to alternate strategies, including *Viral Marketing*. It is one of the most popular application of the influence maximization problem. In fact, the main purpose of the influence maximization problem is to detect a set of k influencers in a social network that are able to trigger a large cascade of influence. This set of influencers is the set of the first users (called seeds) that will receive the Viral Marketing message in order to propagate it through the network. In fact, in a Viral Marketing campaign, the marketer needs to know some influencers for his campaign. Next, he tries to convince them to send Viral Marketing messages, related to the company, the brand or the product, triggering, thus, a cascade of influence by which friends will recommend intentionally or unintentionally the product to other friends and many individuals will, ultimately, adopt it.

The Viral Marketing takes its name from the spread of viruses or computer viruses as it uses a viral propagation to promote its messages. A classic example of this phenomenon is the example of *Hotmail email service* [54]. In fact, Hotmail had very fast adoption of a Viral Marketing strategy and reported a significant rise of its business from influence propagation. It gains 18 million users in 12 months, spending only $50,000 on traditional marketing which is a miniscule advertising budget. The strategy adopted by the Hotmail company was very simple, they, just included into each sent email a simple promotional message like *'PS: We love you! Get your free email at Hotmail,'*[1]

In this document we are interested in the influence maximization problem. In fact, our main purpose is to propose new solutions for this problem in order to improve the quality of selected seeds. We choose the Viral Marketing to be an application for the proposed solutions.

2.4.6 Discussion

In this section, we focused on the reviewing of influence maximization models. We classified them into three classes, in the first one, we presented basic models that use an information propagation model to estimate the user's influence in the network, like ICM and LTM [55]. The second category of models uses past propagation in the network to predict the user's influence, like the CD model [41]. Finally, in the third category we reviewed influence maximization models that considered the opinion in their process.

In Table 2.1, we give a summery of some relevant influence maximization models and their limitations. We found that the most common limitations are related to the parameters of the model that are generally supposed to be given as explicit parameters. For instance,

[1]Matt Janaway, Want to Hack Serious Business Growth? Do a Lean Start Up!, published on April 4, 2016, Seen on June, 13, 2016

the works of [55], [23] and [91] supposed to have the user influence values. A second common limitation is related to the user's opinion about the product. This parameter was omitted in [55], [23], [1] and [41]. On the other hand, we found that the opinion parameter is considered only in the work of *Zhang et al.* [99]. However, they assumed to have the user's opinion as input. We find the opinion in some other works, but they use a signed network that models generally the trust and distrust between users, like the work of [91], or they consider only the negative opinion like [23]. Other existing works, assume implicitly the positive opinion of all network users as they do not consider the opinion in their process. In Chapter 3 of this thesis, we propose some solutions to handle these problems and to ameliorate the quality of selected seeds.

Table 2.1: Limitations of existing influence maximization models

Work	**Description**	**Limitation**
Kempe et al. [55]	They proposed a greedy algorithm for the spread of influence through a social network.	• Overlooked the user's opinion in the influence process. • The probability of influence is defined randomly.
Goyal et al. [41]	They used past propagation to estimate the influence spread of users, and they used a greedy based solution to maximize the influence.	• Overlooked the user's opinion in the influence process.
Chen et al. [23]	They incorporated negative opinions and their propagation in influence maximization.	• The probability of negative opinion is known and given as an explicit parameter. • Overlooked the learning of influence and opinion probabilities.

Work	Description	Limitation
Ahmed et al.[1]	They considered the trust and distrust relationships in their influence maximization model.	• Overlooked the user's opinion in the influence process.
Zhang et al. [99]	They incorporated the spread of positive opinions in the influence maximization problem.	• Only positive opinions are considered in the influence maximization problem. • The probability of positive opinions is known and given as an explicit parameter.
Wang et al. [91]	They proposed an extension of the Linear Threshold model to mine positive influencer form signed social networks.	• Influence probabilities of nodes and nodes thresholds were randomly generated. • Used a signed network to model the opinion.

2.5 Social message classification

In this section, we consider another important problem in the context of social networks analysis fields which is the problem of social message classification, *i.e.* we call a social message, a message that is sent or published through an online social network. In fact, we propose new solutions for this problem in which there are no need for the content of the message neither for external sources of information. In this section, we review some social message classification approaches that we find useful and interesting.

Sriram et al. [85] classified tweets to a set of generic classes which are "News", "Events", Opinions", "Deals" and "Private Messages". For this purpose they defined a set of features

extracted from the tweet and the author's profile, then for each tweet they have a value for each feature. In the classification step, they used the classical Naive-Bayes classifier and they got better results than traditional methods. Similarly, *Zubiaga et al.* [102] classified tweets into the following classes: news, ongoing events, memes (funny or attractive tweets), and commemoratives (congratulating a birthday for example). They defined a set of 15 features to characterize each tweet and they used the Support Vector Machines classifier [52] to identify the class of each tweet. The advantage of this solution is that it is indpendent to the language of the tweet.

Another solution was proposed for short text clustering that uses not only the short text content but also an additional set of items that is extracted from an external source of information like Wikipedia and WorldNet[2]. *Banarjee et al.* [9] proposed a clustering approach that uses Wikipedia to enrich the content of a given social message before classifying it. The idea behind this solution is that for a given message they retrieve the top matching Wikipedia articles to that message. Then, they used the titles of these articles as an extra feature in the clustering step. Similarly, *Hu et al.* [47] introduced a solution for short text clustering that uses internal information from the short text and external information from Wikipedia and WorldNet. First, they process the short text to extract phrases, *i.e.* a set of words that produce a grammatical unit like the noun phrase "the year" for example, that are called seed phrases. These seed phrases are next used to extract external features from Wikipedia. When the phrase does not contain sufficient non-stopwords[3], authors use the WordNet to enrich that phrase. Finally, all extracted features are used together in the clustering step.

Social messages are also classified for sentiment analysis and opinion mining purposes. In this case, the task is to identify the dominant opinion about a product or a brand using text mining techniques. *Mostafa* [71] used 3516 tweets to identify costumer's sentiment about some well known brands. *He et al.* [46] used text published on Twitter and Facebook to analyze the opinion about three chain of pizza. *Lo and Potdar* [67] and *Othman et al.* [74] presented a survey of existing opinion mining and sentiment analysis approaches.

Social message classification approaches presented in the literature are generally based on the content of the information and text mining techniques. In fact, they usually define a set of features that are extracted from the social message and sometimes enriched with some external features from Wikipedia for example. With these solutions, one always needs at least the content of the message to classify it. However, in online social networks the content of the message is not always available. Besides, it is very frequent to find very short messages that may contain one or two words or only special characters, *e.g.* tweets on Twitter are short messages. In such cases, existing classification approaches always fail to find the class for the social message. As a consequence, using a traditional text classification technique

[2]https://wordnet.princeton.edu/

[3]Stop words are words that do not contain important significance to be used as a feature like "a", "about", "after", etc.

to classify tweets, like the "Bag-Of-Words" method, fail to achieve good classification rates due to the message shortness. In chapter 5, we propose a classification approach for social messages that does not need the content of the message to classify it. The proposed approach may be used together with existing ones in order to predict, as accurate as possible, the class of the social message.

2.6 Conclusion

In this chapter, we give an overview of the state of the art of some social network analysis axes to which we contribute with new solutions. In the first place, we review information propagation models. We classify them into two main categories, in the first one we review basic models like ICM and LTM and in the second one we give a summary about epidemic models.

In a second axis, we presented an overview of existing influence measures. In fact, we mainly reviewed influence measures that are designed for Twitter and those that use the theory of belief functions. We found that most of Twitter influence measures use an existing Twitter metric, like indegree, number of mentions, etc, to identify influencers. Whereas, the use of only one metric is not sufficient for influencers identification. On the other hand, we need an influence measure for Twitter users that consider many influence aspects on Twitter. In the Chapter 3, we present a measure that takes all these points into account. Besides, we introduce three extensions of the proposed evidential measure of influence in which we consider the user's positive and negative opinion.

The next axis that we review in this chapter, is about influence maximization models. In fact, we present three categories of models: first, we review basic models that use an information propagation model in their process. Second, we present the influence maximization models that use past propagation in the network to estimate user's influence. Finally, in the third category, we review the models that consider the opinion in their process. In the Chapter 3, we introduce two new influence maximization models that work well with the proposed influence measures. Moreover, in the Chapter 4, we present some cases study that show the performance of the proposed influence measures and maximization models.

Finally, the last research axis we discussed in this chapter is about social message classification. Indeed, we presented some of the existing approaches for this purpose. The problem of social message is its shortness which makes a new challenge to text classification techniques. In the Chapter 5 of this document, we introduce new classification algorithms that use the propagation traces of the message instead of its content in the classification process.

3

Proposed solutions for influence maximization

Contents

Summary

The previous chapter is, mainly, dedicated to the state of the art of the information diffusion, the influence maximization and the social message classification. Indeed, we presented some of the relevant research works that treat these problems. In this chapter, we consider an important problem in the context of social networks which is the influence maximization. It is the problem of selecting a set of influential users in the social network. Those users could adopt the product and trigger a large cascade of adoptions through the "word of mouth" effect. To resolve this problem, we propose four influence measures and two influence maximization models that we detail in this chapter. These solutions will be evaluated in the next chapter.

3.1 Introduction

Viral Marketing exploits existing social networks and sends marketing messages, related to a company, brand or product, triggering, thus, a cascade of influence by which friends will recommend intentionally or unintentionally the product to other friends and many individuals will ultimately adopt it. Hotmail and Yahoo had very fast adoption of a Viral Marketing strategy and reported a significant rise of their business from influence propagation through social networks. Hotmail gains 18 million users in 12 months, spending only $50,000 on traditional marketing [54], while Gmail rapidly gains users although referrals are the only way to sign up.

The influence maximization in online social networks (OSN) presents two main challenges: the first challenge is about data imprecision[1] and uncertainty[2]. In fact, OSNs allow only a limited access for their data, *e.g.* Twitter API[3] allows a limited number of requests per hour, which generates more imprecision and uncertainty for the social network analysis research fields. Then, if we ignore this imperfection of the data, we may be confronted to obtain erroneous analysis results. The second challenge is about the diversity of influence markers and parameters. Indeed, it is important to combine all of them to obtain a global influence measure that considers all these parameters and takes into account the data imperfection and the conflict that may exists between influence markers. In such a situation, the theory of belief functions [26, 79] has been widely applied. We find it used, for example, in some related research fields like pattern clustering [28, 66] and classification [65]. Furthermore, this theory was used for analyzing social networks [94, 35, 51, 50, 100]. More details about the evidence theory can be found in Appendix A.

Existing influence maximization solutions, generally, ignore many interesting influence aspects. In fact, we notice that most existing solutions use only the network structure to identify the influencers [55, 23, 99, 57]. However, the network structure is not sufficient for this task [41]. In fact, we can commonly fall on inactive users that are well positioned in the network. Then, there is a need for influence maximization solutions in which we consider more influence aspects like the user's activity in the network. Another important parameter of this problem is the user's opinion. In fact, this parameter is not considered in most existing works. In fact, we find some works that consider the opinion in their model. However, it is not always an opinion about the Viral Marketing campaign. Hence, comes the need for new solutions that consider the user's opinion.

The main contributions presented in this chapter are the following: first, we propose a new influence measure for Twitter that combines many influence aspects. We note that the proposed measure can be adapted to other social networks. This measure is a first part of

[1] The imprecision of the information is characterized by its content. In fact, it is related to the information or to the source. It measures a quality issue of the knowledge.

[2] The uncertainty of the information characterizes the degree of its conformity to the reality. Therefore, an uncertain information describes a partial knowledge of the reality.

[3] https://dev.twitter.com/

the subject of the paper *Jendoubi et al. [49].* The second contribution is an amelioration of the proposed influence measure. Then, we propose a set of measures that take into account the user's opinion about the product. Furthermore, we introduced three viral marketing scenarios that are adapted to the proposed measure. The first scenario looks for influencers having positive opinion. A solution for this first scenario has been published in *Jendoubi et al.* [48]. The second scenario looks for influencers having a positive opinion and that exert more influence on positive users and the third scenario is about influencers having a positive opinion and that exert more influence on negative users. A third contribution is about influence maximization. In fact, we introduced two greedy-based influence maximization models. These models are a second part of the subject of the paper *Jendoubi et al.* [49].

This chapter is organized as follows: Section 3.2 introduces a new evidential measure of influence. Section 3.3 presents three Viral Marketing scenarios and proposed influence measures for them. Section 3.4 explains the influence maximization models and the greedy based algorithm which we use to get a seed set of influencers. Finally, section 3.5 is a set of running examples that explains more and compares the behavior of the proposed influence maximization solutions.

3.2 Evidential measure of influence for Twitter

In the literature, influence in Twitter was widely studied. However, most of existing works either try to study Twitter users' behavior or use existing Twitter metrics like indegree and the number of mentions to identify influencers. In this section, we introduce a new influence measure for Twitter. The influence estimation process is as follows:

- In a first step, we assign to each influence aspect a weight and for each link (u, v) in the network, we attribute a vector of weights that has the form (w_f, w_m, w_r). This step is detailed in section 3.2.1.
- In a second step, for each link weight, $w_x(u,v) \in \{w_f(u,v), w_r(u,v), w_m(u,v)\}$, $(u,v) \in E$, we use the theory of belief functions (see Appendix A) to estimate a BBA distribution, defined on the frame of discernment $\Omega = \{I, P\}$, I for an influencer and P for a passive user. This step is detailed in section 3.2.2.

Consequently, we obtain, for each link, three influence BBA distributions that represent follow, mention and retweet beliefs respectively. To get a measure that contracts these BBAs, we combine them using the Dempster's rule of combination (equation (A.19) in Appendix A). Then, we obtain a BBA, $m^{\Omega}(u,v)$, that models the influence of u on v. The proposed influence measure will be, after, used with an influence maximization model for Viral Marketing perspective [49].

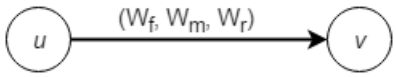


Figure 3.1: Weight vector between u and v.

3.2.1 Link weights estimation

Twitter social network is a multi-relational network that allows an explicit and many implicit relationships between users. In this work, we are mainly interested in the follow relation which is explicit and two implicit relations which are the mention and the retweet. We consider these relations as influence indicators and we use them to estimate the amount of influence that exerts u on its neighbor v. In Twitter, two users u and v can have a follow, a mention and/or a retweet relation between them. To model this property, we assign to each of those a weight and we attribute to each link (u, v) a vector of weights that has the form (w_f, w_m, w_r) as shown in Figure 3.1. The defined weights can be explained as follows:

- The follow weight w_f measures the strength of the followership between u and v, *i.e.* w_f measures the fact that u still receives v's tweets even if the direct followership relation is broken, *via* intermediary users between them.

- The mention weight w_m weights information exchange between users u and v. Indeed, when u mentions v in a tweet then this second (v) will receive directly the message. This behavior emphasizes direct communication between Twitter users.

- The retweet weight w_r represents the information diffusion and influence weight between users, in fact, more v retweets from u more it is influenced by u [13].

Let $G = (V, E)$ be the social network where V is the set of nodes and E is the set of links. *Ben Jabeur et al.* [13] proposes a measure to estimate each of these weights. Let $Sc_u \subseteq V$ be the set of immediate successor of $u \in V$, $Pc_u \subseteq V$ the set of immediate predecessors of u, Tc_u the set of tweets of u, $Rt_u(v)$ the set of tweets of u that were retweeted by $v \in V$, $Mt_u(v)$ the set of tweets of u in which v was mentioned and Mt_u the set of tweets in which u mentions any user in the network except himself. According to *Ben Jabeur et al.* [13], the weights w_f, w_m and w_r of the link $(u, v) \in E$ are estimated using the following measures:

$$w_f(u, v) = \frac{|Sc_u \cap Pc_v| + 1}{|Sc_u|} \tag{3.1}$$

$$w_m(u, v) = \frac{|Mt_u(v)|}{|Mt_u|} \tag{3.2}$$

$$w_r(u, v) = \frac{|Rt_u(v)|}{|Tc_u|} \tag{3.3}$$

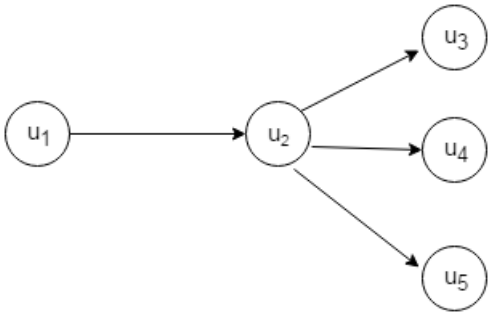


Figure 3.2: Follow weight example

These measures propose to estimate the link weights at the level of the source of the link, *i.e.* the divisor is always related to the source. The measures defined by *Ben Jabeur et al.* [13] are not suitable for our case. Indeed, in the case where the source of the link, u, has few successors, *i.e.* small Sc_u, then its out links will get high follow weights and the same goes for mention and retweet weights. This fact causes erroneous results in the next steps. In fact, we may be confronted to obtain users that have high influence value, but they are not influencers, *i.e.* with small values of $|Sc_u|$, of $|Mt_u|$ and of $|Tc_u|$.

Example 5. Let's take the example in Figure 3.2. If we use the equation (3.1) to estimate $w_f(u_1, u_2)$, we will obtain $w_f(u_1, u_2) = 1$. □

To remedy this problem, we modify these definitions to estimate the links weights with respect to the whole network as follows:

$$w_f(u,v) = \frac{|Sc_u \cap Pc_v| + 1}{|Sc_{max}|} \tag{3.4}$$

$$w_m(u,v) = \frac{|Mt_u(v)|}{|Mt_{max}|} \tag{3.5}$$

$$w_r(u,v) = \frac{|Rt_u(v)|}{|Tc_{max}|} \tag{3.6}$$

such that:

$$|Sc_{max}| = \max_{u \in V} |Sc_u| \tag{3.7}$$

$$|Mt_{max}| = \max_{u \in V} |Mt_u| \tag{3.8}$$

$$|Tc_{max}| = \max_{u \in V} |Tc_u| \tag{3.9}$$

After computing the three weights for each link in the network, we move on to the node level and we compute the three weights for each node, *i.e.* for each node in the network we compute a follow weight, a retweet weight and a mention weight. Node weights are obtained by summing its out links weights as:

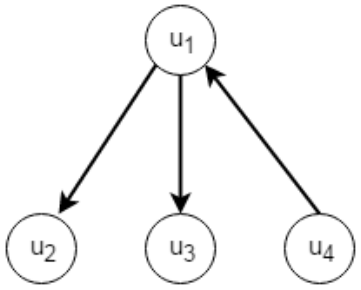


Figure 3.3: Network example

Link	W_f	W_m	W_r
(u_1, u_2)	0.3	0.4	0.2
(u_1, u_3)	0.4	0.3	0.1
(u_4, u_1)	0.5	0.4	0.3

(a) Links weights

Node	w_f	w_m	w_r
u_1	0.7	0.7	0.3
u_2	0	0	0
u_3	0	0	0
u_4	0.5	0.4	0.3

(b) Nodes weights

Table 3.1: Links and nodes weights

$$w_x(u) = \sum_{v \in V} w_x(u, v) \tag{3.10}$$

where $w_x(u) \in \{w_f(u), w_r(u), w_m(u)\}$ and $w_x(u, v) \in \{w_f(u, v), w_r(u, v), w_m(u, v)\}$. In the formula (3.10), we use the sum function to aggregate user's weights for its simplicity, but it is possible to use another aggregation function like the mean [49].

Example 6. Let's take the network example given in Figure 3.3, in this example, we have a social network of four users related to each other by three links. Suppose that after applying the process of link weights estimation described above for each link, we obtain weights given in Table 3.1a. To compute each node weights, we sum up its outlinks weights, then the follow weight of the node u_1 is $w_f(u_1) = w_f(u_1, u_2) + w_f(u_1, u_3) = 0.3 + 0.4 = 0.7$. Nodes weights are given in Table 3.1b.

In this section, we introduced a set of equations to estimate links and nodes weights from Twitter data. These weights summarize the information that we have about the network users, the links between them and their activity. In the next section, we use these weights to estimate the influence in Twitter.

3.2.2 Evidential influence measure

In this section, we present the estimation process of the proposed influence measure that is based on the defined weights in the previous section. We note that the proposed estimation process can be adapted to any directed weighted social network, this point will be detailed at the end of this section.

Let $\Omega = \{I, P\}$ be the influence frame of discernment: I models the user's influence and P the user's passivity, a user cannot be influencer and passive at the same time. We mean by passivity the antonym of influence, *i.e.* when the user is not influencer at all. Besides, let $G = (V, E, W)$ be a directed graph where $v \in V$, $u \in V$ are nodes in G, $(u, v) \in E$ is the edge having u as a source and v as a destination and W is the set of weights vectors, such that $(w_f(u, v), w_m(u, v), w_r(u, v)) \in W$ is the weight vector associated to (u, v). More details about basic concepts of the graph theory can be found in Appendix B. The influence estimation process contains three basic steps:

- Estimate a BBA distribution for each node in the network, this BBA summarizes many influence aspects that are related to the node.
- For each node, use its estimated BBA (the result of step one) to update its in-links weights.
- Use the updated weights to estimate a BBA distribution that contracts many influence aspects.

Next, we detail these steps, their importance in the influence estimation process and their impact on the resulting measure.

3.2.2.1 Step 1: Node level

In the node level step, the main purpose is to estimate a BBA distribution for each node in the network using its weights values. We used the theory of belief functions to define the BBA. The use of the evidence theory in this step is justified by the following properties:

1. The Dempster-Shafer theory allows its user to model various kinds of information through a rich modeling framework.
2. The theory of belief functions provides powerful combination rules for information fusion. In this step, we used the Dempster's rule of combination in order to combine the information that comes from the three influence parameters we defined.
3. The third property that justifies our choice concerns the conflict management. Indeed, when we combine many pieces of information from different sources, we can face some conflict between them, for example, one source may say that the user is an influencer but another source says that he is passive. In such a case, the Dempster's rule computes the conflict and redistribute it on the focal elements.

Let $N_{min_x} = \min_{u \in V} w_x(u)$ and $N_{max_x} = \max_{u \in V} w_x(u)$. For each node in the network, we estimate a mass distribution for each variable, *i.e.* Follow, Mention and Retweet, using its

weight. For each $u \in V$, and for each weight $w_x(u) \in \{w_f(u), w_r(u), w_m(u)\}$, we estimate a mass distribution as follows [94, 35]:

$$m^{\Omega}_{x_u}(I) = \frac{w_x(u) - N_{min_x}}{\vartheta_x} \tag{3.11}$$

$$m^{\Omega}_{x_u}(P) = \frac{N_{max_x} - w_x(u)}{\vartheta_x} \tag{3.12}$$

$$m^{\Omega}_{x_u}(\{I,P\}) = 1 - \left(m^{\Omega}_{x_u}(I) + m^{\Omega}_{x_u}(P)\right) \tag{3.13}$$

where $\vartheta_x = N_{max_x} - N_{min_x} + \alpha$, $\alpha \in [0,1]$ and $\vartheta_x \in \{\vartheta_f, \vartheta_m, \vartheta_r\}$. The mass value given to the set $\Omega = \{I, P\}$ is the mass that can not be given to its singletons and it is called total ignorance. At the end of this step, we have three BBA distributions defined on Ω, *i.e.* follow BBA $m^{\Omega}_{f_u}$, mention BBA $m^{\Omega}_{m_u}$ and retweet BBA $m^{\Omega}_{r_u}$, for each node in the network. Then, we combine all these BBAs using the Dempster's rule of combination (equation (A.19)):

$$m^{\Omega}_{u} = \left(m^{\Omega}_{f_u} \oplus m^{\Omega}_{r_u}\right) \oplus m^{\Omega}_{m_u} \tag{3.14}$$

Once $m^{\Omega}_{u_u}$ is computed, we apply the pignistic transformation on it. Then, we obtain a pignistic probability distribution $BetP^{\Omega}_{u}$ (the reader can refer to section A.4 for more details). In this stage, we have a probability distribution for each node that reflects the following influence aspects:

1. The importance of the user in the network structure. Indeed, the number of user's followers on Twitter network reflects his structural importance. In fact, more he has followers, more he is important and his tweets interests more users.
2. The popularity of user's tweets that is measured using the number of times where user's tweets are retweeted. In fact, the more the tweet is retweeted, more it propagates through the network and more users read it.
3. The popularity of the user that is measured by the number of times the user was mentioned in other users tweets. Indeed, we assume that more the user is mentioned more he is popular in the network.

In the next step, we use the pignistic probability distribution for each node u, $BetP^{\Omega}_{u}$, to update u in-links weights.

3.2.2.2 Step 2: Updating links weights

The main purpose of this second step is to consider the following assumption: *"I am more influencer if I am connected to influencer users"*. This assumption means that when a given user is connected to other influencer users, his personal influence increases. To take into

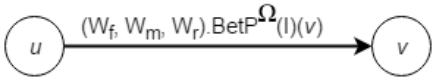


Figure 3.4: Updating link weights

account this assumption, we update weights vector of each link in the network using the estimated pignistic probability distributions defined on the link destination node:

$$w_x^{'}(u,v) = w_x(u,v).BetP_v^{\Omega}(I) \tag{3.15}$$

where $w_x(u,v) \in \{w_f(u,v),\ w_r(u,v),\ w_m(u,v)\}$ and $w_x^{'}(u,v) \in \left\{w_f^{'}(u,v),\ w_r^{'}(u,v),\ w_m^{'}(u,v)\right\}$ is the vector of updated link weights. In this equation, we ponder the weight value given to the influence link between u and v by the influence pignistic probability of the destination node v, $BetP_v^{\Omega}(I)$. Using the equation (3.15), the influence of the node v will propagate to its in-neighbors as shown in Figure 3.4. Then, if the influence of v is high, the weights of its in-links will maintain a high value from their original amount and if the influence of v is low, the weights of its in-links will maintain only a low value from their amount before the updating. Therefore, if a user u is connected to many influencer users, then, his own influence will be consolidated using the proposed equation.

In the next step, we move on to the link level and we estimate the influence that exerts a user u on his neighbor v *via* the link (u,v).

3.2.2.3 Step 3: Link level

In this third step, we estimate the influence that exerts each user u on his neighbors in the network. The theory of belief functions is used in the link level step for the same reasons as the first step. In fact, this theory provides a powerful framework for information modelling. Besides, it we have the choice between many combination rules while information fusion, which allows the use of an adapted combination rule according to the information properties. First of all, for each link $(u,v) \in E$ and for each weight value, $w_x^{'}(u) \in \left\{w_f^{'}(u), w_r^{'}(u), w_m^{'}(u)\right\}$, we estimate a mass distribution, $m^{\Omega}_{x_{(u,v)}}$, on the frame $\Omega = \{I,\ P\}$, as follow:

$$m^{\Omega}_{x_{(u,v)}}(I) = \frac{w_x^{'}(u,v) - L_{min_x}}{\pi_x} \tag{3.16}$$

$$m^{\Omega}_{x_{(u,v)}}(P) = \frac{L_{max_x} - w_x^{'}(u,v)}{\pi_x} \tag{3.17}$$

$$m^{\Omega}_{x_{(u,v)}}(\{I,P\}) = 1 - \left(m^{\Omega}_{x_{(u,v)}}(I) + m^{\Omega}_{x_{(u,v)}}(P)\right) \tag{3.18}$$

where:

$$L_{min_x} = \min_{(u,v)\in E} w_x^{'}(u,v) \tag{3.19}$$

$$L_{max_x} = \max_{(u,v)\in E} w_x^{'}(u,v) \tag{3.20}$$

$$\pi_x = L_{max_x} - L_{min_x} + \varepsilon \tag{3.21}$$

such that $\varepsilon \in [0,1]$ is used to model an imprecise knowledge by adding an amount of belief on ignorance, *i.e.* the ignorance value is the mass on the set $\{I,P\}$, to model our uncertainty. As a result, we have got three BBA distributions for each link $(u,v) \in E$, *i.e.* follow BBA $m^{\Omega}_{f_{(u,v)}}$, mention BBA $m^{\Omega}_{m(u,v)}$ and retweet BBA $m^{\Omega}_{r_{(u,v)}}$. In the next stage, we need to combine these three BBAs into one. For that purpose, we use the Dempster's rule of combination (equation (A.19)):

$$m^{\Omega}_{(u,v)} = \left(m^{\Omega}_{f_{(u,v)}} \oplus m^{\Omega}_{m_{(u,v)}}\right) \oplus m^{\Omega}_{r_{(u,v)}} \tag{3.22}$$

Therefore, for each link (u,v), we obtain a mass distribution, $m^{\Omega}_{(u,v)}$, that consider the following influence aspects:

1. The strength of the link between u and v in the network structure that is measured by the mean of the follow weight.
2. Information exchange and propagation activities between users that is considered through mention and retweet weights respectively.
3. The fact of being more influencer if you are connected to influencer users.

Finally, we define the influence of the user u on his neighbor v as the amount of mass given to the influence $\{I\}$ as:

$$Inf(u,v) = m^{\Omega}_{(u,v)}(I) \tag{3.23}$$

In this section, we introduced a new process for estimating the user's influence in an online social network for Viral Marketing perspectives [49]. We used Twitter as an example to detail more the proposed process. However, the proposed influence measure can be adapted for many other social networks, we just need to define link weights. In such a case, it is possible to define only one weight for each link, then, we use the same process to estimate the influence mass function. The only difference is that there is no need to combine. Also, it is possible to define as weights as influence aspects we have, and the proposed process remains always applicable. As a result, we get an influence measure that summarizes and

combines all defined influence aspects. In the next section, we introduce new extensions of the proposed influence measure in which we consider the user's opinion about the product. These extensions will be evaluated in the next chapter.

3.3 Opinion-based influence measures

In the previous section, we introduced a new evidential measure of influence for social network users. The novelty of the proposed measure is that it is based on the theory of belief functions to combine many influence aspects into one measure. In this section, we incorporate a new important parameter in the proposed influence measure which is the user's opinion about the product. This parameter plays a crucial role in a Viral Marketing campaign (more details about Viral Marketing can be found in section 2.4.5). In fact, if a user u shares his negative opinion about a product, then all users that will receive the opinion of u will have, at least, some doubt about the product and that in the case where u is not influencer for them. In the case where u is an influencer user, then, his negative opinion will be harmful for the product. This fact encouraged us to propose new influence measures for online social networks that consider the user's opinion. Details about the process we used to estimate the user's opinion from real messages can be found in section 4.3. Besides, we introduce the following three scenarios of opinion based Viral Marketing:

1. First scenario, *Positive influencers*: in this scenario we look for influencers having a positive opinion about the product. It is useful for marketers who are looking for positive influencer spreaders. In this case, the marketer may want to avoid influencers that have a negative opinion, and to target only influencer spreaders that have a positive opinion. A solution for this scenario was published in *Jendoubi et al.* [48].

2. Second scenario, *Positive influencers influencing positive users*: the purpose in this scenario is to find positive influencers that exert more influence on users having a positive opinion about the product. It is destined to marketers who are interested by influencer spreaders that have a positive opinion about the product and that are connected to users having a positive opinion too. In such a case, the marketer may want to boost the probability of success of his Viral Marketing campaign.

3. Third scenario, *Positive influencers influencing negative users*: the main goal of this scenario is to detect positive influencers that exert more influence on users having a negative opinion about the product. It is useful for marketers who are looking for influencer spreaders that have a positive opinion about the product and that are connected to users having a negative opinion. The marketing strategy here may be, for example, to try to gain more customers by changing the opinion of users that have a negative opinion.

In this section, we detail the solutions we propose for each defined scenario. In fact, we present two influence measures for each one. Then, we introduce the influence spread function. First of all, let's define our framework. Let $G = (V, E, W)$ be a directed graph (the social network) where V is a set of vertices, E is a set of directed links and W is the set of weights vectors associated with each link. Let $\Theta = \{Pos, Neg, Neut\}$ be a frame of discernment expressing opinion, Pos for positive, Neg for negative and $Neut$ for neutral, such that Pos, Neg and $Neut$ are exclusive. Let $\Omega = \{I, P\}$ a frame of discernment expressing influence and passivity, I for influencer and P for passive user, a given user can not be influencer and passive at the same time. Besides, we define a probability distribution $\Pr_u^\Theta$ on Θ to express the opinion of the user $u \in V$ about the product. In fact, we estimate the user's opinion through his published messages (see section 4.3 for some details about the estimation process), then, we can not be sure about the positivity, the negativity or the neutrality of his opinion which justifies the use of a probability distribution to model the opinion. Finally, we define a basic belief assignment (BBA) function $m_{(u,v)}^\Omega$ on Ω to model the influence that exerts the user u on v.

We transform the opinion probability distribution $\Pr_u^\Theta$ to a mass distribution m_u^Θ in order to consider the uncertainty that may exist in the user's opinion (see Appendix A for more details about the used uncertainty theory). Indeed, to estimate the user's opinion, generally, we use machine learning tools that give a good approximation of the opinion but not a certain one. For this reason, we propose to use the theory of belief functions to adjust slightly the approximation errors. For this purpose, we create two simple BBA distributions for $\Pr_u^\Theta(Pos)$ and $\Pr_u^\Theta(Neg)$. In fact, we take α value, the simple BBA parameter defined in equation (A.7), equals to $\Pr_u^\Theta(Pos)$ for the first BBA and to $\Pr_u^\Theta(Neg)$ for the second one. After this step we obtain two BBAs expressing the user's positive and negative opinions respectively. In the next step, we combine the resulting BBAs to obtain m_u^Θ that expresses the opinion of u. We will justify the choice of this transformation process in the next chapter while presenting the opinion learning process (Section 4.3.1). Indeed, we show that $\Pr_u^\Theta(Pos)$ and $\Pr_u^\Theta(Neg)$ come from independent sources. Next, we detail our solution for each Viral Marketing scenario.

Positive influencers. The goal in the first scenario is to detect social influencers that have a positive opinion about the product. In fact, we search to avoid negative influencers, because targeting these users may have a harmful effect on the Viral Marketing campaign. For example, the marketer wants to promote his product in an online social network. First, he starts by identifying a set of influencers in the network that maximizes the total influence. Second, he contacts them and tries to convince them to do some advertising for his product. He may give the influencers a free product or a discounting in order to encourage them more to do the advertising. If by chance he falls on some influencers that do not like his product, what would be their reaction in such a case? Then we propose to avoid negative influencers by detecting and targeting positive influencers. As defined in section 3.2, the mass value

$m^{\Omega}_{(u,v)}(I)$ measures the influence of u on v but without considering the opinion of u about the product. We define the positive opinion influence of u on v as the *positive proportion* of $m^{\Omega}_{(u,v)}(I)$ and we propose two measures to estimate this proportion as:

$$Inf_1^+(u,v) = \Pr_u^{\Theta}(Pos).m^{\Omega}_{(u,v)}(I) \quad (3.24)$$

$$Inf_2^+(u,v) = m_u^{\Theta}(Pos).m^{\Omega}_{(u,v)}(I) \quad (3.25)$$

In equation (3.24), we weight $m^{\Omega}_{(u,v)}(I)$ using $\Pr_u^{\Theta}(Pos)$ to estimate the positive influence [48], $Inf_1^+(u,v)$, while in equation (3.25), we consider the uncertainty of the user's opinion and we use $m_u^{\Theta}(Pos)$ to weight $m^{\Omega}_{(u,v)}(I)$ to estimate $Inf_2^+(u,v)$.

Positive influencers influencing positive users. In this second scenario, the goal is to select among positive opinion influencers, those that are connected to and exert more influence on positive users. We emphasize such a scenario is very useful in the Viral Marketing world, especially when the marketer wants to make his viral marketing campaign safer by targeting users having a positive opinion. To detect positive influencers that influence positive users, we define two influence measures by weighting $Inf_1^+(u,v)$ and $Inf_2^+(u,v)$ using $\left(1-\Pr_v^{\Theta}(Neg)\right)$ and $\left(1-m_v^{\Theta}(Neg)\right)$ respectively as follows:

$$Inf_1^{++}(u,v) = \Pr_u^{\Theta}(Pos).m^{\Omega}_{(u,v)}(I).\left(1-\Pr_v^{\Theta}(Neg)\right) \quad (3.26)$$

$$= Inf_1^+(u,v).\left(1-\Pr_v^{\Theta}(Neg)\right) \quad (3.27)$$

$$Inf_2^{++}(u,v) = m_u^{\Theta}(Pos).m^{\Omega}_{(u,v)}(I).\left(1-m_v^{\Theta}(Neg)\right) \quad (3.28)$$

$$= Inf_2^+(u,v).\left(1-m_v^{\Theta}(Neg)\right) \quad (3.29)$$

The proposed measures, Inf_1^{++} and Inf_2^{++}, give more importance to the positive connection. Indeed, the values of $\left(1-\Pr_v^{\Theta}(Neg)\right)$ and $\left(1-m_v^{\Theta}(Neg)\right)$ emphasize the positive opinion of u's neighbor.

Positive influencers influencing negative users. In the third scenario, we emphasize influencer users having a positive opinion about the product and that exert more influence on negative users. For example, the goal in this case may be to gain more customers by convincing negative users and make them change their opinion. Then, we define two influence measures for this scenario that are based on those defined in the first scenario. In fact, we multiply $Inf_1^+(u,v)$ and $Inf_2^+(u,v)$ with the non-positive proportion of v opinion using $\left(1-\Pr_v^{\Theta}(Pos)\right)$ and $\left(1-m_v^{\Theta}(Pos)\right)$ respectively as:

$$Inf_1^{+-}(u,v) = \Pr_u^{\Theta}(Pos).m_{(u,v)}^{\Omega}(I).\left(1-\Pr_v^{\Theta}(Pos)\right) \quad (3.30)$$

$$= Inf_1^{+}(u,v).\left(1-\Pr_v^{\Theta}(Pos)\right) \quad (3.31)$$

$$Inf_2^{+-}(u,v) = m_u^{\Theta}(Pos).m_{(u,v)}^{\Omega}(I).\left(1-m_v^{\Theta}(Pos)\right) \quad (3.32)$$

$$= Inf_2^{+}(u,v).\left(1-m_v^{\Theta}(Pos)\right) \quad (3.33)$$

The proposed measures, Inf_1^{+-} and Inf_2^{+-}, emphasize negative connections. In fact, the values of $\left(1-\Pr_v^{\Theta}(Pos)\right)$ and $\left(1-m_v^{\Theta}(Pos)\right)$ give more importance to neighbors having a negative opinion about the product.

To sum up this section, we introduce three new scenarios of viral marketing that can arise. The first scenario is intended for marketers who are looking for influencer users that have a positive opinion about the product (or the brand, etc) object of the viral marketing campaign. The second scenario searches to detect influencer spreaders that have a positive opinion about the product and exert more influence on positive users. In the third scenario we target positive influencers influencing negative users. For each defined scenario, we propose two influence measures in order to detect the targeted users. The proposed scenarios are adaptable for many social networks. Indeed, we just need to choose an appropriate opinion estimation process. In this thesis, we choose Twitter to be a validation example of the proposed solutions. Then, we used an existing opinion estimation process adapted to tweets. This process is introduced in secion 4.3.

3.4 Two influence maximization models

To maximize the influence in a social network, we need an influence maximization model. In this section, we define the amount of influence given to a set of users, $S \subseteq V$, for influencing a given user, $v \in V$. To estimate this amount we define two models. In fact, we introduce two new influence maximization models, the first one is suggested to be used in the case where we have a product with some quality issue, and the second influence model can be used in other cases. The proposed models can be used with each influence measure introduced in this chapter. Finally, we study the properties of the proposed models to choose an appropriate approximation algorithm.

3.4.1 Measuring the influence of a set of users

We start this section by defining the amount of influence given to a set of nodes, $S \subseteq V$, for influencing a user $v \in V$. We present two estimation models. The first model considers

the influence links directly connected to nodes in S. The second estimation model considers directed influence links, like in the first method, and the intermediate nodes having a direct influence link from S and a second influence link towards v. In other words, the second method models the fact that says *"my friend's influencer is my influencer"*.

Example 7. Let's take the network given in Figure 3.5 as an example. In this network, we have $S = \{u, w, x, w\}$ and we want to calculate the influence of S on v. If we use the first method, we consider the influence on the links (u, v) and (y, v). However, if we use the second method, we consider not only the influence on the links (u, v) and (y, v), but also, the influence on the links (x, z) and (z, v). □

Let $M1$ and $M2$ be the first and the second models respectively. We estimate the influence of S on a user v as follows:

$$\Phi_{M1}(S, v) = \begin{cases} 1 & \text{if } v \in S \\ \sum_{u \in S} Inf(u, v) & \text{Otherwise} \end{cases} \tag{3.34}$$

$$\Phi_{M2}(S, v) = \begin{cases} 1 & \text{if } v \in S \\ \sum_{u \in S} \sum_{x \in D_{IN}(v) \cup \{v\}} Inf(u, x).Inf(x, v) & \text{Otherwise} \end{cases} \tag{3.35}$$

such that $Inf(v, v) = 1$ and $D_{IN}(v)$ is the set of in-neighbors of v. The work of *Chen et al.* [23] justifies the two proposed models. In fact, they affirm that when the product have some quality issues, it is more adaptable to choose influencers having many immediate neighbors. In fact, when the influence propagates in many hops in the network, it may fall on a user that dislikes the product. Besides, when the product has a high quality, we can choose users that have a large reachable set.

Finally, we define the influence spread functions, $\sigma^{Bel}_{M1}(S)$, $\sigma^{Bel}_{M2}(S)$, under the two proposed models, respectively, as the total influence given to $S \subseteq V$ from all nodes in the social network as follows:

$$\sigma^{Bel}_{M1}(S) = \sum_{v \in V} \Phi_{M1}(S, v) \tag{3.36}$$

$$\sigma^{Bel}_{M2}(S) = \sum_{v \in V} \Phi_{M2}(S, v) \tag{3.37}$$

In the spirit of the influence maximization problem, as defined by [55], $\sigma^{Bel}_{M1}(S)$ and $\sigma^{Bel}_{M2}(S)$ are the objective functions to be maximized.

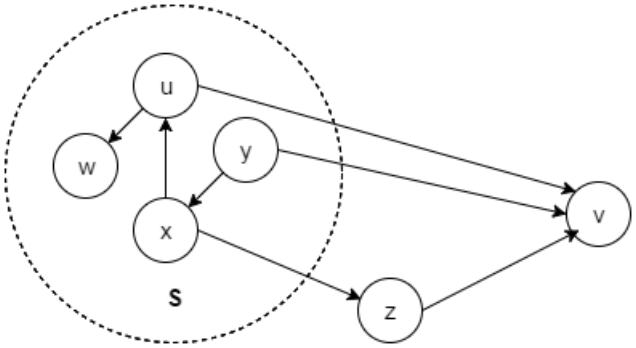


Figure 3.5: Measuring influence example

3.4.2 Objective functions properties

In this section, we study the properties of the proposed objective functions in order to choose the appropriate maximization algorithm. We demonstrate that the proposed functions are monotone and submodular [49]. Before that, we define these two properties.

Definition 5. *A monotone set function*, σ, is a function between ordered sets that either preserves or reverses the given order, *i.e.* $\sigma(S) \leq \sigma(T)$ if $S \subseteq T$ when x preserves the order and $\sigma(S) \geq \sigma(T)$ if $S \subseteq T$ when σ reverses the order. In our case, we are interested in increasing monotone set functions that preserves the given order.

Definition 6. *A submodular set function*, σ, is a set function that has a natural diminishing returns property which means that the gain of σ when adding an element x to a superset T is no more than the gain of σ when adding the same element to a subset S, $S \subseteq T$:

$$\sigma(S \cup \{x\}) - \sigma(S) \geq \sigma(T \cup \{x\}) - \sigma(T), \; S \subseteq T \tag{3.38}$$

The use of submodular monotone objective functions in the influence maximization field is very common in the literature. For example, the following works used such a function in their model [55, 56, 57, 41]. Next, we show that the defined objective functions are monotone and submodular.

Theorem 1. *$\sigma^{Bel}_{M1}(S)$ and $\sigma^{Bel}_{M2}(S)$ are monotone and submodular.*

Proof. $\sigma^{Bel}_{M1}(S)$ and $\sigma^{Bel}_{M2}(S)$ are monotone

$$\sigma^{Bel}_{M1}(S) \leq \sigma^{Bel}_{M1}(T), \; S \subseteq T \tag{3.39}$$

$$\sigma^{Bel}_{M2}(S) \leq \sigma^{Bel}_{M2}(T), \; S \subseteq T \tag{3.40}$$

In fact, $\sum_{v \in V} \Phi_{M1}(S, v) \leq \sum_{v \in V} \Phi_{M1}(T, v)$ and $\sum_{v \in V} \Phi_{M2}(S, v) \leq \sum_{v \in V} \Phi_{M2}(T, v)$.

$\sigma_{M1}^{Bel}(S)$ and $\sigma_{M2}^{Bel}(S)$ are submodular if and only if

$$\sigma_{M1}^{Bel}(S\cup\{x\})-\sigma_{M1}^{Bel}(S) \geq \sigma_{M1}^{Bel}(T\cup\{x\})-\sigma_{M1}^{Bel}(T),\ S\subseteq T \tag{3.41}$$
$$\sigma_{M2}^{Bel}(S\cup\{x\})-\sigma_{M2}^{Bel}(S) \geq \sigma_{M2}^{Bel}(T\cup\{x\})-\sigma_{M2}^{Bel}(T),\ S\subseteq T \tag{3.42}$$

i.e. the marginal gain of x with respect to T is no more than the marginal gain of x with respect to S. To demonstrate the submodularity of the proposed objective functions, we distinguish two cases. First, the case were $x\in S$, we have

$$\sigma_{M1}^{Bel}(S\cup\{x\})-\sigma_{M1}^{Bel}(S) = \sigma_{M1}^{Bel}(T\cup\{x\})-\sigma_{M1}^{Bel}(T)=0,\ S\subseteq T \tag{3.43}$$
$$\sigma_{M2}^{Bel}(S\cup\{x\})-\sigma_{M2}^{Bel}(S) = \sigma_{M2}^{Bel}(T\cup\{x\})-\sigma_{M2}^{Bel}(T)=0,\ S\subseteq T \tag{3.44}$$

In this case, the marginal gain of x in zero in all cases. Second, the case were $x\notin S$, we have

$$\sigma_{M1}^{Bel}(S\cup\{x\})-\sigma_{M1}^{Bel}(S) = 1+\sum_{v\in V\setminus S} Inf(x,v) \tag{3.45}$$
$$\sigma_{M1}^{Bel}(T\cup\{x\})-\sigma_{M1}^{Bel}(T) = 1+\sum_{v\in V\setminus T} Inf(x,v) \tag{3.46}$$
$$\sigma_{M2}^{Bel}(S\cup\{x\})-\sigma_{M2}^{Bel}(S) = 1+\sum_{v\in V\setminus S}\sum_{a\in D_{IN}(v)\cup\{v\}} Inf(x,a).Inf(a,v) \tag{3.47}$$
$$\sigma_{M2}^{Bel}(T\cup\{x\})-\sigma_{M2}^{Bel}(T) = 1+\sum_{v\in V\setminus T}\sum_{a\in D_{IN}(v)\cup\{v\}} Inf(x,a).Inf(a,v) \tag{3.48}$$

In the two models, we have $S\subseteq T$ which means that $|S|\leq|T|$, then $|V\setminus S|\geq|V\setminus T|$ which proves the sub-modularity of $\sigma_{M1}^{Bel}(S)$ and $\sigma_{M2}^{Bel}(S)$. □

We demonstrated the monotonicity and the submodularity of $\sigma_{M1}^{Bel}(S)$ and $\sigma_{M2}^{Bel}(S)$. In the next stage, we demonstrate that the maximization of $\sigma_{M1}^{Bel}(S)$ and $\sigma_{M2}^{Bel}(S)$ in an NP-Hard problem [49].

Theorem 2. *The influence maximization using the first of the second proposed models is NP-Hard.*

Proof. To demonstrate the NP-Hardness of our approach, we show that it can be seen as a particular case of the CD approach [41] that was shown to be NP-Hard. If we assume that we have one action a then

$$\gamma(u,v)(a)=\Gamma(u,v)(a)=Inf(u,v) \tag{3.49}$$

then we can write $\Gamma(S, v)$ as:

$$\Gamma(S,v) = \begin{cases} 1 & \text{if } v \in S \\ \sum_{x \in D_{IN}(v)} \Phi_{M2}(S,x).Inf(x,v) & \text{Otherwise} \end{cases} \tag{3.50}$$

$\Phi_{M2}(S, v)$ can be seen as $\Gamma(S, v)$ of the CD model by considering only two hops between neighbors while estimating influence. CD model is proved to be NP-Hard [41]. Consequently, we prove that the maximization of the second model, *i.e.* in which we use the objective function $\sigma^{Bel}_{M2}(S)$, is NP-Hard. The first model, *i.e.* in which we use the objective function $\sigma^{Bel}_{M1}(S)$, can be viewed as a generalization of the second model of the proposed approach. □

In this section, we demonstrated that the proposed objective functions are monotone and submodular. Besides, we shown the NP-Hardness of the proposed evidential models. In the next section, we select an appropriate optimization algorithm that goes with the properties of our models.

3.4.3 Maximization algorithm

The main purpose in this chapter is to find a set of k influencer users that maximizes the total influence in a social network. First, we define a set of data-based influence measures. These measures are based on the theory of belief functions to combine many influence aspects and to manage the conflict that may appear between them. Second, we move to the influence model and we propose two models. Next, we study their properties and we prove their NP-Hardness. Now, we need to choose an appropriate optimization algorithm to maximize the defined objective functions. In section 2.4.4, we discuss the maximization algorithms that are used to maximize the influence in the literature.

We showed that the influence maximization under the evidential model is NP-Hard, besides, the influence spread function is monotone and sub-modular. Therefore, these properties allow to choose the greedy algorithm because it performs good approximation for the optimal solution. We choose the cost effective lazy-forward algorithm (CELF) [60] which is a two pass modified greedy algorithm that is proved to be about 700 times faster than the basic greedy algorithm. CELF exploits the submodularity property of the function to be maximized, in fact, submodularity guarantees that marginal benefits decrease with the solution size. Hence, instead of computing the marginal benefit of each expected node at each iteration, it computes it in the first iteration and keeps an ordered list of nodes according to their marginal benefits value for the next iteration. In the next iteration, it re-evaluate the marginal benefit for the top node, then it resorts the node list. If the top node maintains its position, it will be chosen, elsewhere the algorithm re-evaluates the marginal benefit for the new top node and so on.

Algorithm 3 shows the steps of the CELF based evidential influence maximization algorithm. First, it starts by initializing the seed nodes set S and the node list Q to the empty set $\emptyset$. Second, it estimates the marginal gain of all nodes in the network and sorts them. Third, the algorithm adds the node with maximum marginal gain to S. Then, it loops on the following steps until getting k seed nodes in S:

1. Select the node, $nodeMax$, having a maximum marginal gain, *i.e.* the head of the list Q,
2. Update the marginal gain of the selected node, $nodeMax$,
3. If $nodeMax$ preserves its position in the list Q, it will be added to S, else, it will be returned to Q and the algorithm returns back to the first step.

Algorithm 3: CELF based evidential influence maximization algorithm

```
begin
    S = ∅;
    // S: the set of seed nodes
    Q = ∅;
    // Q: sorted list in decreasing order according to the marginal gain of
       nodes
    foreach node ∈ V do
        marginalGain(node);
        // marginalGain() estimate the marginal gain of the node
        Q.add(node);
    nodeMax ← Q.pop();
    S.add(nodeMax);
    while | S |≤ k do
        nodeMax ← Q.pop();
        updateMarginalGain(nodeMax);
        // We use formula 3.45 or 3.47 to update the marginal gain
        if nodeMax.MG ≥ Q.getFirst().MG then S.add(nodeMax) ;
        else Q.add(nodeMax) ;
```

In the next section, we present some running examples to explain more the proposed influence maximization solutions, and show the differences between them.

3.5 Running examples

In this section, we present some running examples to illustrate the proposed approach. In the first two running examples, we compare the behavior of each proposed model using the proposed evidential influence measure (equation (3.23)). Next, we present three examples to compare the three opinion scenarios using the second influence model.

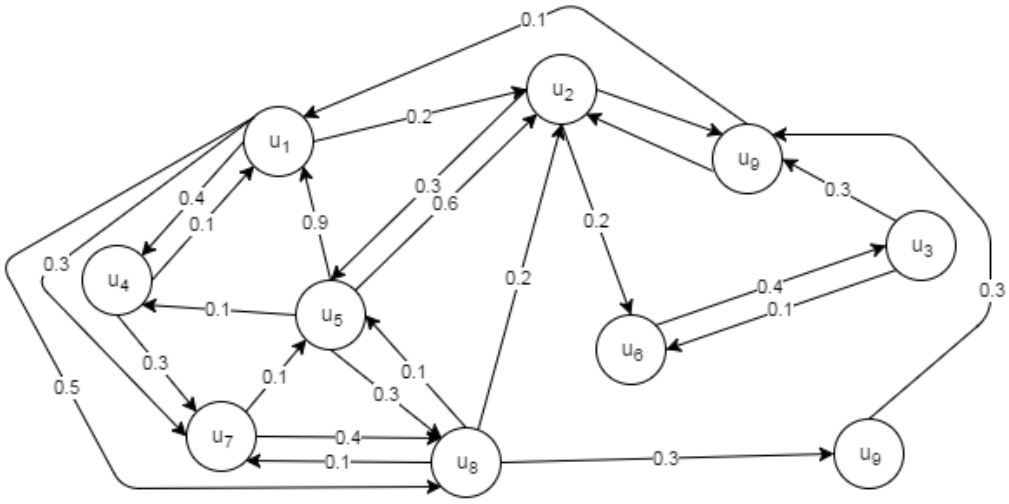


Figure 3.6: Influencers detection example

3.5.1 Influencers detection

In this section, we present two running examples to compare the behavior of the two proposed influence maximization models. In these examples, we use the influence network in Figure 3.6. It is a directed network with ten nodes that are related to each other by twenty five weighted links. Each link in the network is weighted by the influence that exerts its source on its destination. For example, the link (u_1, u_2) is weighted by 0.2, we say that the influence of u_1 on u_2 equals to 0.2.

Example 8. Let's consider the weighted network in the Figure3.6. We fix the parameter k to 3 and we run the Algorithm 3 with the first maximization model, *i.e.* using $\sigma^{Bel}_{M1}(S)$. First, the algorithm initializes $S_1 = \emptyset$. Second, it estimates the marginal gain (MG) of all users and sorts them. Table 3.2 contains the sorted list of nodes according to their marginal gain. Then, we select the node with a maximum MG and we add it to S_1, $S_1 = \{u_5\}$. In the next step, the algorithm loops until getting k nodes in S_1. In fact, the algorithm chooses the node that keeps its position as top of the list to be added to S. Table 3.3 shows the updates that have occurred on nodes marginal gains and orders when the algorithm is looking for the second node to be added to S_1. The node u_4 is the second node to be added to S_1 as $S_1 = \{u_5, u_4\}$. Table 3.4 shows the updates until finding the last node to be added to S_1 which is u_3. Finally, the set of seeds according to the first maximization model is $S_1 = \{u_5, u_4, u_3\}$. □

Example 9. Consider the weighted network in the Figure 3.6. Let $k = 3$, we turn the Algorithm 3 with the second maximization model, *i.e.* using $\sigma^{Bel}_{M2}(S)$. Similarly to the previous example, Table 3.5 presents the sorted list of nodes according to their marginal gain. We add the node u_5 to S_2 as it has the maximum marginal gain, $S_2 = \{u_5\}$. Next, we update the marginal of the other nodes until getting a node that does not change its

Table 3.2: First model: sorted users according to their marginal gain

MG(u_5)	**2.9**	MG(u_7)	1.7
MG(u_2)	2	MG(u_3)	1.5
MG(u_1)	1.8	MG(u_6)	1.4
MG(u_8)	1.7	MG(u_9)	1.3
MG(u_4)	1.7	MG(u_{10})	1.3

Table 3.3: First model: updated marginal gains after selecting $\{u_5\}$

~~MG(u_5)~~	~~2.9~~	MG(u_8)	1.3
MG(u_4)	**1.6**	MG(u_9)	1.3
MG(u_3)	1.5	MG(u_{10})	1.3
MG(u_7)	1.4	MG(u_2)	1.1
MG(u_6)	1.4	MG(u_1)	0.9

position. According to Table 3.6, the node u_7 is the second node to be added to S_2. Table 3.7 shows the updates after adding u_5 and u_7 to S_2. Finally, the last node to be added to S_2 is u_3. Then according to the second model $S_2 = \{u_5, u_7, u_3\}$. □

We present two running examples to explain more the proposed maximization models and the idea behind the CELF algorithm. In the first example, we obtain the seed set $S_1 = \{u_5, u_4, u_3\}$ and in the second example we have got $S_2 = \{u_5, u_7, u_3\}$. We notice that $S_1 \neq S_2$ and this is due to the use of two different maximization models. In fact, the first model chooses the node u_4 to be the second node in S. However, the second model prefers the node u_7 because it influences more its neighbors' neighbors than u_4. In the next section, we present other examples to explain the difference between the three opinion-based scenarios that are presented in section 3.3.

3.5.2 Opinion-based influencers detection

In this section, we present three examples using the second influence maximization model. In the first example, our purpose is to detect positive opinion influencers. In the second example, we are looking for positive influencers that influence positive users, and in the last example, we detect positive influencers influencing negative users. The main purpose behind these examples is to test the efficiency of the proposed opinion-based influence measures. Then, we define the input of the examples in such a way one knows the influencers, their

Table 3.4: First model: updated marginal gains after selecting $\{u_5, u_4\}$

~~MG(u_5)~~	~~2.9~~	MG(u_8)	1.3
~~MG(u_4)~~	~~1.6~~	MG(u_9)	1.3
MG(u_3)	**1.5**	MG(u_{10})	1.3
MG(u_7)	1.4	MG(u_2)	1.1
MG(u_6)	1.4	MG(u_1)	0.9

Table 3.5: Second model: sorted users according to their marginal gain

MG(u_5)	**4.29**	MG(u_8)	2.18
MG(u_7)	2.51	MG(u_3)	1.69
MG(u_2)	2.47	MG(u_6)	1.56
MG(u_1)	2.38	MG(u_{10})	1.45
MG(u_4)	2.19	MG(u_9)	1.39

Table 3.6: Second model: updated marginal gains after selecting $\{u_5\}$

~~MG(u_5)~~	~~4.29~~	MG(u_{10})	1.45
MG(u_7)	**2.11**	MG(u_9)	1.39
MG(u_4)	1.91	MG(u_2)	1.33
MG(u_3)	1.69	MG(u_1)	1.32
MG(u_6)	1.56	MG(u_8)	1.24

opinions and their neighbors' opinions. Finally, the task in each example is to detect the good influencers.

We created the network in Figure 3.7, this network is composed of two trees of thirteen nodes and twelve links each one. We gave the same influence probabilities in the two trees for links that have the same position. Besides, we defined these probabilities to obtain two influencers which are the root nodes, u_1 and u_{14}. Table 3.8 presents the users' opinions. We define a positive opinion for the root nodes and we define u_1 to be a positive influencer that influences negative users and u_{14} to be a positive influencer that influences positive users.

Example 10. Consider the network in Figure 3.7 and the Table 3.8 that contains the users' opinions. The purpose of this example is to detect positive opinion influencers. We fix $k = 2$ and we estimate the marginal gain of each node in the network using the second influence maximization model with the equation (3.24), we obtain the marginal gains in Table 3.9. After applying the CELF algorithm, we get $S = \{u_1, u_{14}\}$. The mean positive opinion of the seed nodes equals to 0.8, the mean positive opinion of seeds' neighbors is 0.4 and the mean negative opinion of seeds' neighbors is 0.43. These results mean that we achieve our goal in detecting positive opinion influencers. □

Example 11. In this example, we are interested in positive opinion influencers that influence positive users. Then, let's consider the network in Figure 3.7 and the Table 3.8 that contains the users' opinions. We fix $k = 1$ and we run the CELF algorithm with the second influence

Table 3.7: Second model: updated marginal gains after selecting $\{u_5, u_7\}$

~~MG(u_5)~~	~~4.29~~	MG(u_9)	1.39
~~MG(u_7)~~	~~2.11~~	MG(u_2)	1.33
MG(u_3)	**1.66**	MG(u_1)	1.32
MG(u_6)	1.56	MG(u_4)	1.29
MG(u_{10})	1.45	MG(u_8)	1.24

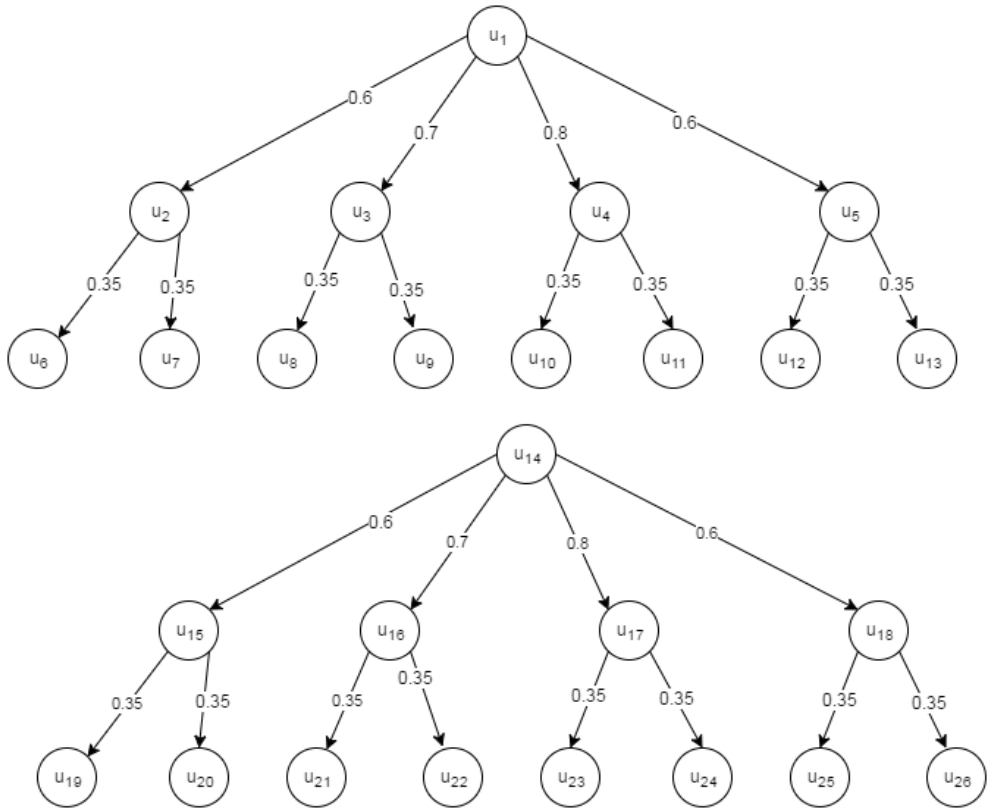


Figure 3.7: Opinion-based influencers detection example

Table 3.8: Users opinions

Id	**Pos**	**Neg**	**Obj**	**Id**	**Pos**	**Neg**	**Obj**
u_1	0.8	0.1	0.1	u_{14}	0.8	0.1	0.1
u_2	0.3	0.6	0.1	u_{15}	0.6	0.3	0.1
u_3	0.1	0.7	0.2	u_{16}	0.7	0.1	0.2
u_4	0.2	0.8	0	u_{17}	0.8	0.2	0
u_5	0.2	0.6	0.2	u_{18}	0.6	0.2	0.2
u_6	0.33	0.33	0.34	u_{19}	0.33	0.33	0.34
u_7	0.33	0.33	0.34	u_{20}	0.33	0.33	0.34
u_8	0.33	0.33	0.34	u_{21}	0.33	0.33	0.34
u_9	0.33	0.33	0.34	u_{22}	0.33	0.33	0.34
u_{10}	0.33	0.33	0.34	u_{23}	0.33	0.33	0.34
u_{11}	0.33	0.33	0.34	u_{24}	0.33	0.33	0.34
u_{12}	0.33	0.33	0.34	u_{25}	0.33	0.33	0.34
u_{13}	0.33	0.33	0.34	u_{26}	0.33	0.33	0.34

Table 3.9: Marginal gain table of the first scenario example

MG(u_1)	**2.89**	MG(u_{10})	1	MG(u_{19})	1
MG(u_2)	1.14	MG(u_{11})	1	MG(u_{20})	1
MG(u_3)	1.04	MG(u_{12})	1	MG(u_{21})	1
MG(u_4)	1.09	MG(u_{13})	1	MG(u_{22})	1
MG(u_5)	1.09	**MG(u_{14})**	**2.06**	MG(u_{23})	1
MG(u_6)	1	MG(u_{15})	1.28	MG(u_{24})	1
MG(u_7)	1	MG(u_{16})	1.18	MG(u_{25})	1
MG(u_8)	1	MG(u_{17})	1.37	MG(u_{26})	1
MG(u_9)	1	MG(u_{18})	1.28		

model and using the equation (3.26) to estimate the influence. At the end, we get $S = \{u_{14}\}$. The positive opinion of u_{14} is 0.8, the mean positive opinion of u_{14} neighbors is 0.6 and the mean negative opinion of u_{14} neighbors is 0.2. In these results, we notice that the second influence model detects the good influencer which is in our example the node u_{14}. □

Example 12. In this last example, we are mainly looking for positive opinion influencers that influence users having negative opinion. We consider the same network and opinion table as in the previous example. Also, we fix $k = 1$. Then, we run the CELF algorithm with the third influence model and using the equation (3.30) to estimate the influence. As a result, we have $S = \{u_1\}$. The positive opinion of u_1 is 0.8, the mean positive opinion of u_1 neighbors is 0.2 and the mean negative opinion of u_1 neighbors is 0.67. In this example, we are looking for an influencer having a positive opinion and that exerts more influence on negative users and we succeed in detecting it. □

In this section, we present some running examples to illustrate the proposed solution for the problem of influence maximization in a social network. The presented examples explain more the running process of the proposed CELF algorithm and show the differences between the proposed influence models and measures.

Table 3.10: Marginal gain table of the second scenario example

MG(u_1)	**1.74**	MG(u_{10})	1	MG(u_{19})	1
MG(u_2)	1.14	MG(u_{11})	1	MG(u_{20})	1
MG(u_3)	1.04	MG(u_{12})	1	MG(u_{21})	1
MG(u_4)	1.09	MG(u_{13})	1	MG(u_{22})	1
MG(u_5)	1.09	**MG(u_{14})**	**3.22**	MG(u_{23})	1
MG(u_6)	1	MG(u_{15})	1.28	MG(u_{24})	1
MG(u_7)	1	MG(u_{16})	1.18	MG(u_{25})	1
MG(u_8)	1	MG(u_{17})	1.37	MG(u_{26})	1
MG(u_9)	1	MG(u_{18})	1.28		

Table 3.11: Marginal gain table of the third scenario example

MG(u_1)	**2.89**	MG(u_{10})	1	MG(u_{19})	1
MG(u_2)	1.14	MG(u_{11})	1	MG(u_{20})	1
MG(u_3)	1.04	MG(u_{12})	1	MG(u_{21})	1
MG(u_4)	1.09	MG(u_{13})	1	MG(u_{22})	1
MG(u_5)	1.09	**MG(u_{14})**	**1.89**	MG(u_{23})	1
MG(u_6)	1	MG(u_{15})	1.28	MG(u_{24})	1
MG(u_7)	1	MG(u_{16})	1.32	MG(u_{25})	1
MG(u_8)	1	MG(u_{17})	1.37	MG(u_{26})	1
MG(u_9)	1	MG(u_{18})	1.28		

3.6 Conclusion

In this chapter, we mainly focus on the problem of social influence maximization. It is the problem of detecting a set of k influencers that are able to trigger a large cascade of adoptions through the social network. In fact, while studying the existing solutions for this problem, we find that many important influence aspects and parameters were not considered. Among these aspects, we find the user's opinion about the product that we consider as crucial parameter in the influence maximization problem. To remedy the drawbacks of existing models, we introduce new measures of influence and two maximization models that work with them.

First, we propose an evidential influence measure that contracts many influence aspects like the user's position in the network, the popularity of user's tweets, etc. Second, we incorporate the user's opinion on the proposed evidential measure and we introduce three Viral Marketing scenarios that can arise in the real world. The first scenario is about the detection of social influencers that have a positive opinion about the product. In the second scenario, the purpose is to find positive influencers that exert more influence on users having a positive opinion about the product. In the third scenario, we are looking for positive influencers that influence more users having a negative opinion about the product. In the second place, we define two influence maximization models and we use the CELF algorithm to maximize the influence using the two proposed models. Finally, to illustrate the proposed influence maximization solutions, we propose some running examples in which

we detail the process of each model.

In the next chapter, we present a set of experiments to compare the proposed solutions to existing ones and to study the quality of the detected seeds on real world data. Indeed, we propose two case studies that are done on two different datasets. The first dataset was collected from Twitter and the second one was randomly generated.

4

Influence maximization: Experimental study

Contents

Summary

In the previous chapter, we propose many influence measures for users of an online social network. The proposed measures consider many influence aspects. Also, we introduce two evidential influence maximization models. In this chapter, we focus on two case studies in order to compare the proposed Viral Marketing solutions to existing ones and to prove the performance of our solutions. Some parts of the results presented in this chapter are published in Jendoubi et al. [48].

4.1 Introduction

In the past, Word-of-Mouth (WoM) was seen as a powerful factor in sharing information about a product, promotion, etc., between a customer and a friend, colleague, or other acquaintance. With the appearance of online social networks, the WoM was developed more and more. Furthermore, it is actually used for Viral Marketing perspectives. It is the process of targeting the most influential users in the social network so that these customers can start a chain reaction of influence driven by WoM, thus with a small marketing budget a large proportion of a social network can be reached or influenced [1]. Scientifically, the Viral Marketing problem is known by the influence maximization problem. Its main goal is to find a set of k users that are able to make an information goes viral through the social network.

The previous chapter is mainly dedicated to introducing and explaining the propose Viral Marketing solutions. In fact, we present seven measures of influence that consider many influence aspects. Next, we propose two influence maximization models and we use the Cost-Effective Lazy Forward algorithm to find a set of k seeds that maximizes the proposed models. In this chapter, we focus on the experimentation of the proposed solutions. Indeed, we present a real world Viral Marketing task which is the maximization of the promotion of smartphones on Twitter and we compare the results given by our solutions to the results given by existing models.

This chapter is organized as follows: Section 4.2 introduces the used datasets and the process we used to get them. Section 4.3 explains the method and the tools we used to estimate the user's opinion from his tweets. Section 4.4 presents a set of experiments made on the Twitter dataset to study the performance of the proposed solutions in detecting influencers for smartphones on Twitter. Also, we compared the quality of the selected influencers to the quality of those selected by existing models. Finally, in Section 4.5, we study the accuracy of the proposed solutions using generated data.

4.2 Data gathering and processing

In this section, we present the datasets we used in our experiments. Also, we detail the process we followed and used tools to obtain our data. Next, we propose two datasets, the first one was collected from Twitter and the second one was randomly generated.

4.2.1 Twitter dataset

In our experiments, we define a Viral Marketing task which is about the promotion of smartphones on Twitter. For this purpose, we crawled Twitter data for the period between

Table 4.1: Statistics of the data set

#User	**#Tweet**	**#Follow**	**#Retweet**	**#Mention**
36274	251329	71027	9789	20300

08/09/2014 and 03/11/2014. We used the Twitter API through the Twitter4j java library[1]. It is an open-sourced java implementation of the Twitter API, created by Yusuke Yamamoto. Twitter API provides many kinds of data with some limitations, *i.e.* a limited number of queries per hour or limited response size. In our case, we are interested in collecting tweets written in English, users, who mentions whom and who retweets from whom. Next, we filtered the obtained data by keeping only tweets that talk about smartphones and users having at least one tweet in the data base. In a last step, we used the process explained in the section 4.3 to estimate the opinion of each user in the data set about smartphones.

Table 4.1 presents some statistics about the content of the collected data. Besides, Figure 4.1 displays data distributions over users based on the number of followers, mentions, retweets and tweets across our data. The follow relationship is an explicit relation between Twitter user. In fact, when a user u follows another user v, u will receive all the actuality of v. The mention and the retweet are implicit relations in Twitter. Besides, these relations allow the information propagation on the network. Finally, a tweet is 140 characters message. More details about Twitter can be found in section 2.3.1.1.

4.2.2 Generated dataset

The generated data is used in this thesis to study the performance of the proposed influence measures. In fact, we generated data in such a way one can know the influencers, the positive influencers, the positive influencers influencing positive users and the positive influencers influencing negative users. Then, we obtain a useful dataset to study the accuracy of the proposed influence maximization solutions. Next, we detail the process we used to obtain this data. The proposed process is parameterizable and allows the study of the accuracy variation in terms of each parameter.

Social network structure has some special characteristics that differentiate them from ordinary graphs like the small world assumption [73]. For this reason, we chose to use a real world structure. Then, we selected a random sampling of the collected network from Twitter, *i.e.* Twitter dataset introduced in section 4.2.1. The sampled network contains 1010 nodes and 6906 directed links between them. In a second step, we selected a set of users that have at least 15 outlinks. As a result, we have got a set of 108 users.

Next, we define, randomly, the influence on each link in the network and the selected 108 users are defined as influencers by setting maximum influence values in their outlinks.

[1] http://twitter4j.org/en/index.html

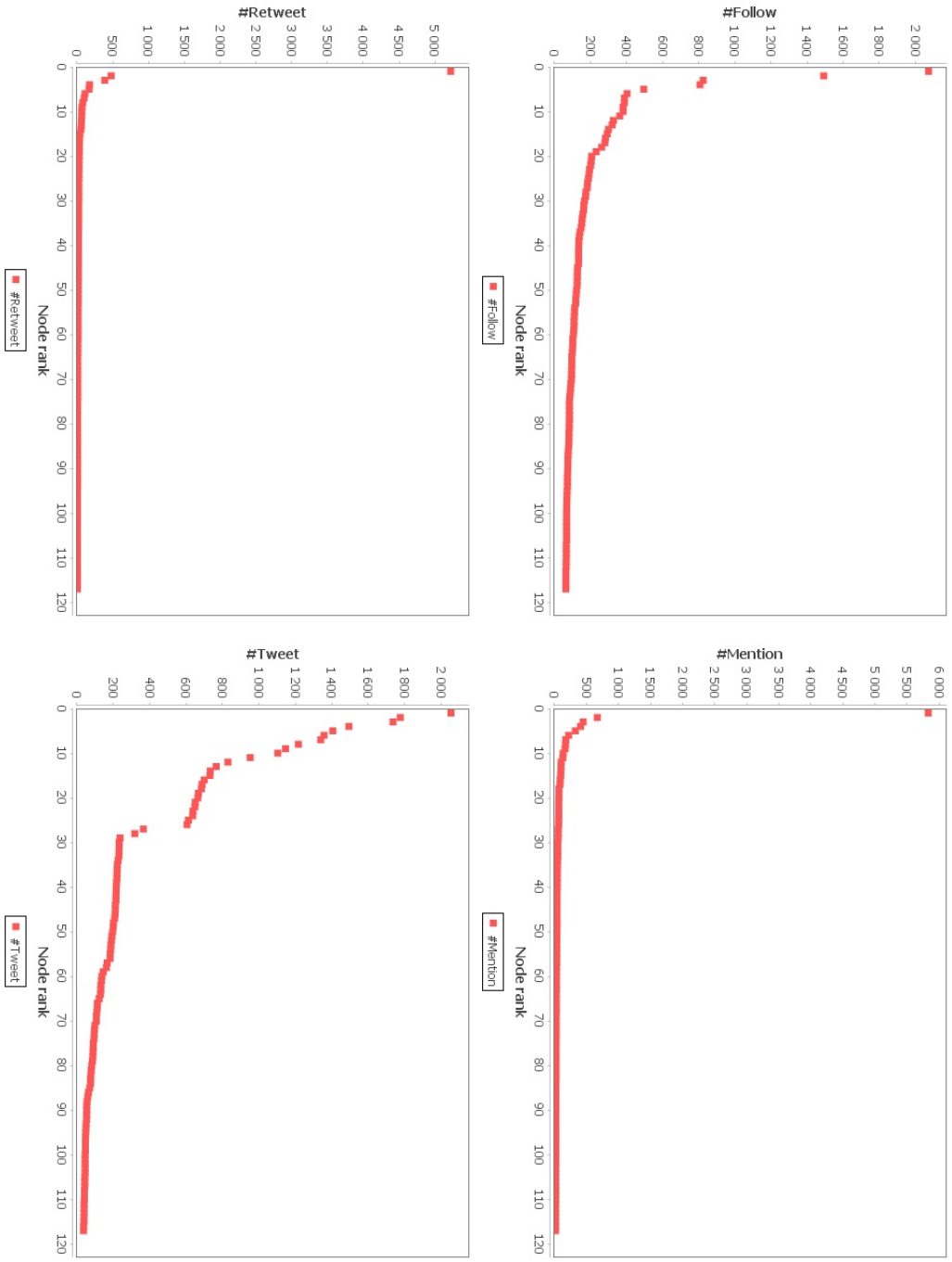


Figure 4.1: Data distributions

The minimum value of influence given to an influencer is a parameter to the random process. In a third step, we define positive influencers among the defined influencer users. Then, we select a random set of influencers and we give them a random value of positive opinion. The chosen positive opinion value equals at least a fixed minimum value given as a parameter to the random process. In a last step, we define among positive influencers those that influences positive and negative users. For this purpose, we divide the set of positive influencers into two random subsets. The first subset is for positive influencers influencing positive users, then, we set the opinion of the influencers neighbors to positive. The second subset is for positive influencers influencing negative users and we set the opinion of their neighbors to negative. We notice that we have two more parameters of the random process which are the minimum positive and negative opinion of positive influencers neighbors.

4.3 User's opinion estimation

In this thesis, we propose a set of influence measures that considers the user's opinion about the product for Twitter dataset. For this purpose, we need to estimate the user's opinion about the product. First, we start by estimating the opinion polarity of each tweet in our dataset, then, we take the user's opinion as the mean opinion of his tweets. Next, we present the tools and the process we used to estimate the user's opinion.

4.3.1 Text mining tools

To estimate the user's opinion about the product, we used some existing text mining tools that are designed for this purpose. Next, we introduce the used tools.

- *Stanford Log-linear Part-Of-Speech Tagger*[2] is a software implemented in java. It takes a text in its input and attributes, to each part of speech on it, a tag. A part-of-speech tag can be a verb, an adjective, an adverb or a noun. The tag of a given part-of-speech depends on many parameters like its position in the text, the context, etc. Stanford part-of-speech tagger is an implementation of the Log-linear part-of-speech tagger [88]. It can be used for any language we just need to choose an appropriate trained model. In our case we chose the Gate Twitter part-of-speech tagger.
- *GATE Twitter part-of-speech tagger*[3] is a trained model designed to be used with Stanford part-of-speech tagger. This model is customized for English tweets [29].
- *SentiWordNet 3.0* is "*a lexical resource explicitly devised for supporting sentiment classification and opinions mining applications*" for English language [7]. It was, automatically, generated from *WordNet*[4] dictionary that is an English lexical database.

[2]http://nlp.stanford.edu/software/tagger.shtml
[3]https://gate.ac.uk/wiki/twitter-postagger.html
[4]http://wordnet.princeton.edu/

Each row in SentiWordnet 3.0 contains six different attributes as follows:

```
# Part-Of-Speech ID PosScore NegScore SynsetTerms Gloss
a 00001740 0.125 0 able#1 (usually followed by 'to')...
a 00002098 0 0.75 unable#1 (usually followed by 'to')...
a 00002312 0 0 dorsal#2 abaxial#1 facing away from...
a 00002527 0 0 ventral#2 adaxial#1 nearest to or...
a 00002730 0 0 acroscopic#1 facing or on the side...
...
```

The first attribute is "*Part-Of-Speech*", it indicates the position of the word in a sentence. For example, "*a*" means adjective. The second attribute is "*ID*" which is an identifier. The couple (Part-Of-Speech, ID) is a unique identifier of the row. Next, we have "*PosScore*" and "*NegScore*" that are opinion-related values expressing positive and negative opinion respectively. Having the positive and the negative opinion, we can obtain the objective or the neural one as:

$$Neut(v) = 1 - (Pos(v) + Neg(v)) \tag{4.1}$$

where $Pos(v)$, $Neg(v)$ and $Neut(v)$ are respectively positive, negative and objective opinion of the user v. The next attribute is "*SynsetTermes*" and it contains a group of data elements or words that are considered semantically equivalent. The last attribute is "*Gloss*" and it may contain a definition or some examples or both of them. We studied the process used to estimate the opinion and we found that the positive and negative values are estimated separately. Next the process tests if $Pos\,(v) + Neg\,(v) \leq 1$, then, $Neut\,(v)$ is computed using equation (4.1), else the sum $Pos(s) + Neg(s)$ is normalized to 1 and $Neut\,(v)$ is null. As a conclusion, the probabilities $Pos(v)$ and $Neg(v)$ are independent and it is possible to consider that they have two different sources which justify the process explained in section 3.3 to transform the user's opinion probability to a basic belief assignment distribution.

4.3.2 Opinion estimation

In this section, we explain the process that we applied to estimate the opinion polarity of each user in our dataset. For this purpose, we used the text mining tools presented in the previous section. First, for each user, we select the set of tweets that he emitted. Next we estimate the opinion polarity of each tweet of those as follows:

1. In the first step, we delete URLs and special characters that are not considered by the used POS tagger. We note that URLs and special characters may be informative for the opinion. However, we choose to not consider them in order to simplify the task.

2. The second step is the part-of-speech tagging. Its goal is to attribute a label (noun, adjective, verb, etc) to each word in the tweet. Then, we use the java library "Stanford POS Tagger" with the model "GATE Twitter part-of-speech tagger" that were designed for tweets.

3. In the third step, we use the SentiWordNet 3.0 dictionary to get the polarity of each word (positive, negative and objective polarity) in the tweet according to its tag (result of the step two). The result of this step is a probability distribution defined on $\Theta = \{Pos, Neg, Neut\}$ for each word in the tweet.

4. The last step is the computation of the polarity of the tweet. Then, we take the mean probability distribution of words' probability distributions that compose the tweet.

Next, after estimating the opinion polarity of all user's tweets, we calculate the user's opinion about the product. As we did for the tweet, we take the mean probability distribution of the user's tweets probability distributions. Finally, for each user in the Twitter dataset we have got a probability distribution defined on the frame $\Theta = \{Pos, Neg, Neut\}$.

4.4 Detecting influencers for smartphones on Twitter

In this section, we present a set of experiments to show the performance of the proposed solutions against existing ones. The task we propose is about the influence maximization for smartphones on Twitter. The main purpose of this task is to find a set of k influencer users that are able to maximize the global influence through the network and to promote the adoption of smartphones. In this set of experiments, we use the dataset collected from Twitter presented in section 4.2.1. Next, we present the set of algorithms with which we compare our approach. After, we divide our experiments into two main parts, in the first part, we show the performance of the proposed influence maximization models with the evidential influence measure. In the second part, we show the interest of incorporating the user's opinion in the influence maximization process.

4.4.1 Experiments configuration

This section is dedicated for algorithms and configurations we used in our experiments. To prove the performance of our evidential approach, we compare it to existing influence maximization solutions. In the literature, there exist many influence maximization models. Then, we chosen to compare the proposed models to some basic models like ICM and LTM and to closest models like CD and OC as follows:

- Basic models detailed in section 2.2.1:

  - Independent cascade model with uniform edge probabilities (UN ICM), *i.e.* all edges have the same influence probability that is equal to 1%.
  - ICM with trivalency edge probabilities (TV ICM), *i.e.* we choose the influence probabilities, randomly, from {10%, 1%, 0.1%}.
  - Weighted cascade (WC ICM) *i.e.* is a special case of ICM with edge probability of an edge (u, v) equals to $\frac{1}{D_u}$.
  - Linear threshold model (LTM) with uniform edge weights $\omega(u, v) = 1\%$ and random threshold θ_u for each node.

- Closest models:

  - Credit distribution (CD) model is detailed in section 2.4.2. We consider the CD model closest in its principle to our models in that it uses real word propagation to estimate the user's influence.
  - Opinion-based cascading (OC) model is detailed in section 2.4.3. It is a modified version of ICM that considers the user's opinion. It sounds like our models in that it considers the user's opinion about the product.

To fix ICM edge probabilities and LTM weights we followed the experiments of previous works [55, 41]. We run each basic model 10000 times with the Monte-Carlo simulation. In fact, basic models used a random process to estimate the influence of a user or a set of users, the idea behind this process is that they use the random propagation simulation algorithm to estimate the influence many times, then, they take the global influence as the mean of all obtained values.

To examine the quality of the selected seeds, we need to fix/choose some comparison criteria. For this purpose, we choose the accumulated number of followers, *#Follow*, the accumulated number of tweets, *#Tweet*, the accumulated number of times the user was mentioned, *#Mention*, and the accumulated number of times the user was retweeted, *#Retweet*. In fact, if a given user is an influencer on Twitter, he is necessarily: very active then he has a lot of tweets, he is followed by many users in the network that are interested in his news, he is frequently mentioned in other tweets and his tweets are retweeted several times. These assumptions justify the chosen comparison criteria.

4.4.2 Quality of detected influencers

In this section, we propose an influence maximization task. It is about detecting influencers for smartphones on Twitter. For this purpose, we use many influence maximization models, listed in section 4.4.1, to select k seeds from the network users. Then, we study the quality

of selected seeds by each model. We denote by "1 Level" the first evidential influence maximization model that uses the formula (3.34) and by "2 Levels" the second evidential model with the formula (3.35). We note that we do not consider the user's opinion in this section.

First, we compare the proposed two maximization models to basic models (ICM and LTM) and CD model. It was very hard to turn basic models on the whole dataset. In fact, as we explaine in the previous section, basic models use the Mont Carlo simulation which is time and memory consuming, besides, this fact was shown by previous works like [41]. To remedy this problem, we used a sampling of 1010 nodes from the original data for this experiment.

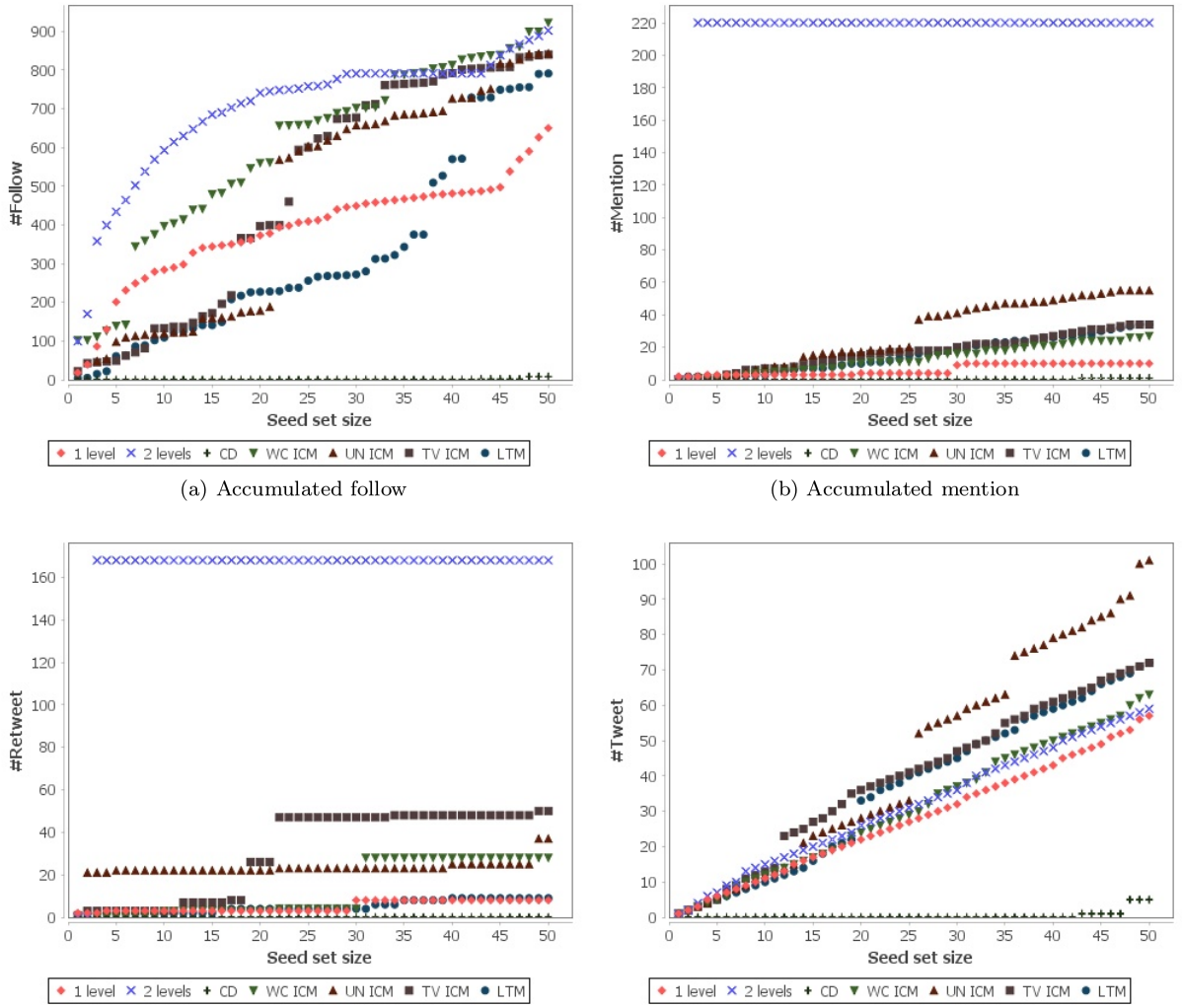
In the first experiment, our purpose is to compare the proposed models with basic models and CD in terms of quality of selected seeds. As mentioned above, the quality of the seed set is measured through four criteria, *i.e.* #Follow, #Mention, #Retweet and #Tweet. Then, we run all the experimented models on the sampled dataset with $k = 50$ and we obtained the results in Figure 4.2.

In Figure 4.2a, we observe that LTM, UN ICM, TV ICM and CD detect weakly connected users at first. However, we observe that the "2 Levels" model starts by detecting strongly connected users. Figure 4.2b shows that most of the scatter plots are close to each other except that of "2 Levels" that detected highly mentioned users at first. In Figure 4.2c, the "2 Levels" model has successfully detected highly retweeted users, also, UN ICM performs well by detecting users that are retweeted by others and we notice that "1 Level", WC ICM and LTM have almost close scatter plots. Finally, Figure 4.2d shows that "1 Level", "2 Levels", WC ICM, UN ICM, TV ICM and LTM detect active users. However, the CD selects inactive users that have few tweets.

From Figure 4.2, we conclude that "1 Level" and "2 Levels" models of the proposed approach detects influencer users. Detected users are active and have a good position in the network that allows them to propagate their messages in a short time. Also, we conclude that "2 Levels" model is the best model in selecting influencer users. In fact, it chooses users having a good compromise between the four criteria, *i.e.* #Follow, #Mention, #Retweet and #Tweet.

In a second experiment, we compare the experimented models in terms of running time. Table 4.2 presents the running time in milliseconds of all models used in experiments of Figure 4.2. In fact, all these models are proven to be NP-Hard [55, 41]. As shown in Table 4.2 the proposed models, "1 Level" and "2 Levels", are faster than the other models. In fact, the "1 Level" model gave its results in 32 milliseconds and the "2 Levels" in 536 milliseconds.

Two other experiments are made using the whole dataset. The purpose of these experiments is to compare the "1 Level" and "2 Levels" models with CD model according to the accumulated #Follow (Figures 4.3a and 4.4a), the accumulated #Mention (Figures 4.3b and 4.4b), the accumulated #Retweet (Figures 4.3c and 4.4c) and the accumulated #Tweet



(a) Accumulated follow

(b) Accumulated mention

(c) Accumulated retweet

(d) Accumulated tweet

Figure 4.2: Comparison between the proposed approach, ICM and LTM

Model	Time (ms)	Model	Time (ms)
1 Level	**32**	**TV ICM**	7267904
2 Level	**536**	**UN ICM**	4844867
CD	4654	**WC ICM**	4295455
LTM	65963285		

Table 4.2: Running time in milliseconds

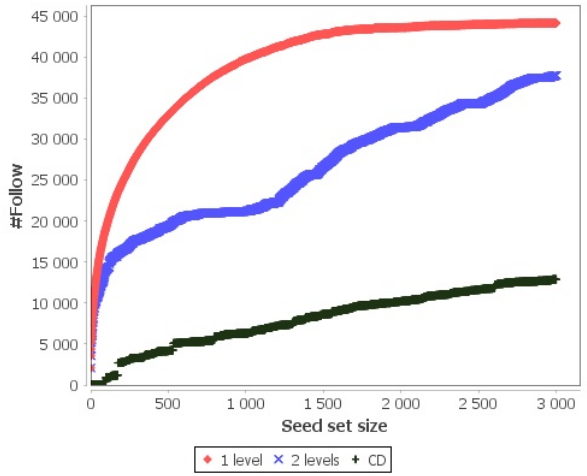
(Figures 4.3d and 4.4d) of seed set nodes. In fact, we consider the Credit Distribution model to be the closest in its principle to "1 Level" and "2 Levels" models.

In the first experiment, we fix k to 3000 (Figure 4.3). The goal here is to see the impact of the size of the seed set on the quality of selected seeds. In the second experiment, we fix k to 100 (Figure 4.4) to study the quality of the first selected seeds by each experimented model.

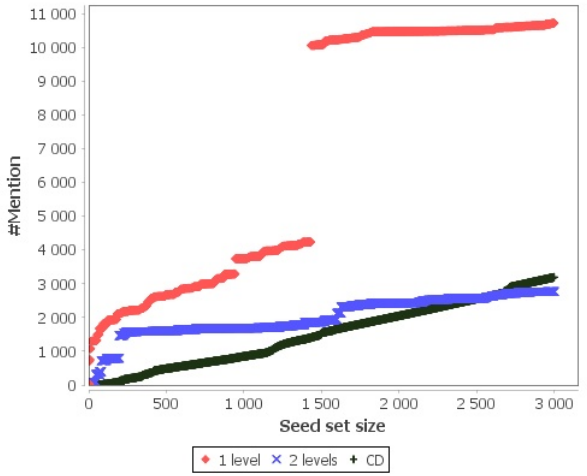
Figures 4.3 and 4.4 show the performance of the proposed models ("1 Level" and "2 Levels") against the credit distribution model. In fact, the evidential influence maximization approach detects influencer spreaders that have a good compromise between #Follow, #Mention, #Retweet and #Tweet. Detected seeds are followed by many users. In Figure 4.4a, the first 10 seeds detected by "1 Level" and "2 Levels" are followed by over 6000 users while there are no followers for the first 10 seeds detected by CD model. According to Figures 4.3b and 4.4b, detected influencer users with "1 Level" and "2 Levels" models are mentioned many times whereas the CD model starts to detect mentioned users after selecting over 93 seed nodes.

In Figure 4.3c, the CD model selected seeds that were retweeted a lot. However, in Figure 4.4c we notice another behavior of this model. In fact, it starts to detect retweeted users after about 70 influencer nodes detected while evidential influence maximization models start detecting them from the second seed. Finally, we move on to the fourth criterion of comparison which is shown in Figures 4.3d and 4.4d. It is about the accumulated activity size of the detected seeds that is measured by their number of tweets. As a first comment, CD model has the same behavior as in the retweet scatter plot. In fact, it starts to detect active users after about 50 selected influencers. Second, the proposed evidential approach demonstrates its performance in detecting active users from the first detected user.

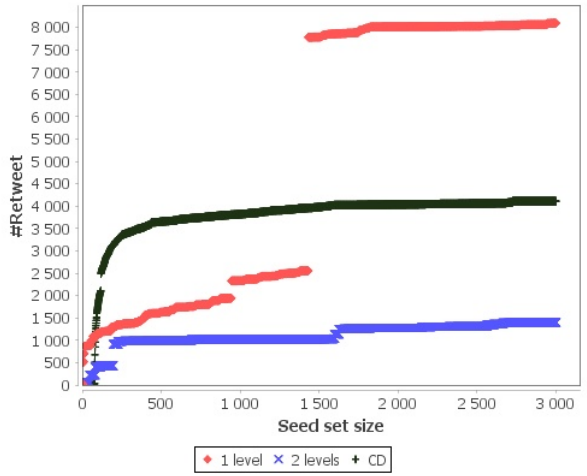
Based on Figures 4.3 and 4.4, the proposed evidential models, "1 Level" and "2 Levels", are better than the CD model in that the evidential models provide a good compromise between the four influence criteria, *i.e.* #Follow, #Mention, #Retweet and #Tweet. Also, the selected influencer spreaders have a good position in the network. Besides, they are active and highly mentioned in other tweets. Furthermore, their tweets are highly retweeted. These observations prove the performance of "1 Level" and "2 Levels" models in selecting seeds in our task. Then, if we want to promote smartphones or another product on Twitter, the use of "1 Level" and "2 Levels" models to detect seeds is recommended.



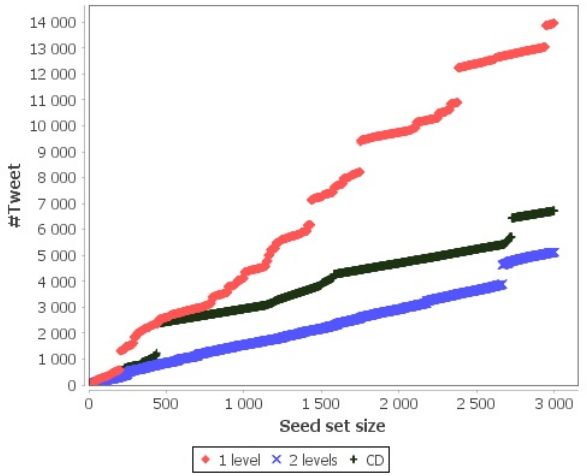
(a) Accumulated follow Comparison



(b) Accumulated mention comparison

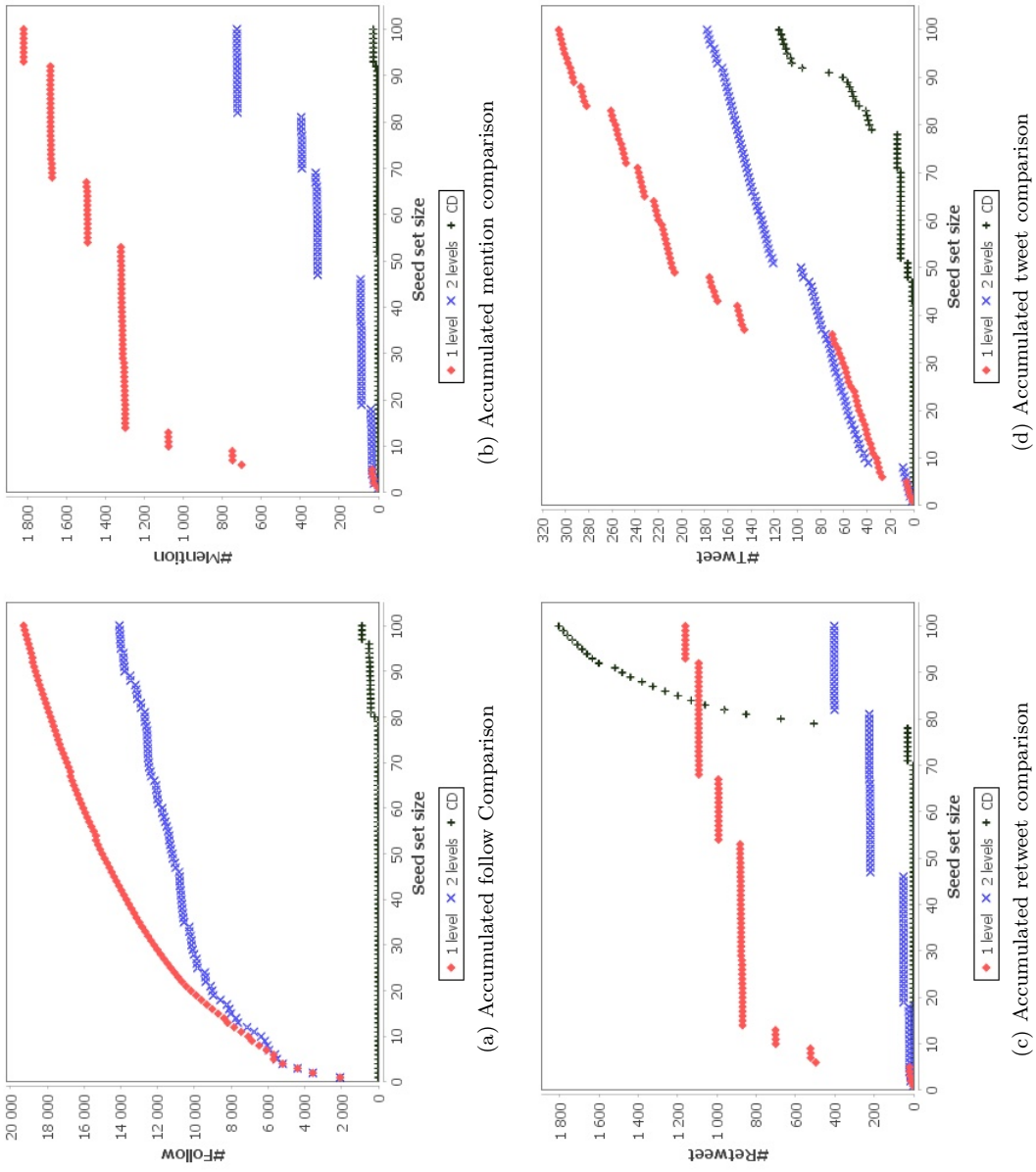


(c) Accumulated retweet comparison



(d) Accumulated tweet comparison

Figure 4.3: Comparison between "1 level", "2 levels" and credit distribution (CD) models with S size = 3000



(a) Accumulated follow Comparison

(b) Accumulated mention comparison

(c) Accumulated retweet comparison

(d) Accumulated tweet comparison

Figure 4.4: Comparison between "1 level", "2 levels" and credit distribution (CD) models with S size = 100

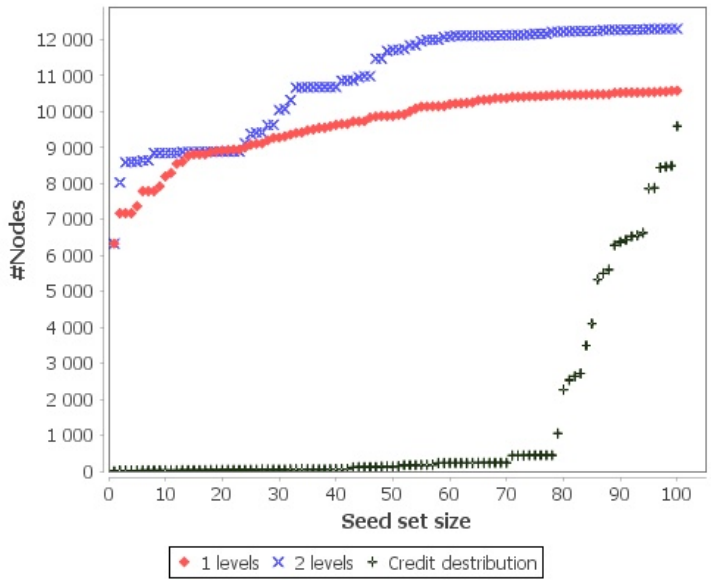


Figure 4.5: The dependance of the number of affected nodes to the size of S

Another interesting experiment is given by the number of distinct affected nodes connected to the influencers and to their neighbors. Through this experiment, we want to see the number of affected nodes by each detected seed set. The results of this experiment are shown in Figure 4.5.

In Figure 4.5, we observe that the CD model detects about 40 isolated users at first and from the seed node 80 it started to detect users that are followed by many other users. In the other hand, we notice a different behavior for the scatter plots of "1 Level" and "2 Levels". In fact, "1 Level" scatter plot is upper than the scatter plot of "2 Levels" in Figure 4.3a. However, in Figure 4.5, "2 Levels" scatter plot is upper than or equal to the scatter plot of "1 Level". From these observations, we conclude that "2 Levels" model detects influencer spreaders that are connected to highly followed users and "1 Level" model detects highly followed influencer spreaders. Also, we conclude that our models (either "1 Level" or "2 Levels") are better than the CD model in detecting highly connected seeds at first.

In a last experiment, we study the impact of considering the assumption of *"being more influencer if you are connected to influencer users"* on the influence maximization results. This fact is defined in the second step, *i.e.* the "*Updating step*", of our influence estimation process. The reader can refer to section 3.2.2.2 for more details. In Figures 4.6 and 4.7, we compare the "1 Level" (Figure 4.6) and the "2 Levels" (Figure 4.7) models with and without the updating step.

In Figure 4.6 the difference between "1 Level" with and without updating step is not significant. In fact, we notice that with the updating step we have, sometimes, slightly better results. However, in Figure 4.7 the impact of the updating step is clear. Indeed, we

observe that the updating step ameliorates the influence maximization results. We explain this observation by the fact that the "2 Levels" model consider the user's neighbors and their neighbors in its principle, then, a given user has more chance to have one or more influencer connected to him and then to reinforce more his influence.

In this section, we study the behavior of the two defined influence maximization models on a real world Viral Marketing task. Besides, we compare the two proposed models to some basic models and to CD model that we consider the closest to our models in its principle. The experiments we made, show the performance of the proposed influence maximization models in detecting reliable influencers according to the defined criteria. In the next section, we study the impact of the incorporation of the user's opinion about the product in the influence measure.

4.4.3 Impact of the opinion incorporation

In this section, we perform some experiments to study the impact of the user's opinion about the product on the detected seeds. We use the dataset collected from Twitter, the reader can refer to section 4.2.1 for more details. Next, we use the second influence maximization model that uses the equation (3.35) with the proposed influence measures as follows:

- The evidential influence measure, Inf, (equation (3.23)), called "2 Levels" or "2 Levels with Inf".
- The first measure of the first scenario, Inf_1^+, (equation (3.24)), called "2 Levels with Inf_1^+" or "First scenario with probability opinion".
- The second measure of the first scenario, Inf_2^+, (equation (3.25)), called "2 Levels with Inf_2^+" or "First scenario with belief opinion".
- The first measure of the second scenario, Inf_1^{++}, (equation (3.26)), called "2 Levels with Inf_1^{++}" or "Second scenario with probability opinion".
- The second measure of the second scenario, Inf_2^{++}, (equation (3.28)), called "2 Levels with Inf_2^{++}" or "Second scenario with belief opinion".
- The first measure of the third scenario, Inf_1^{+-}, (equation (3.30)), called "2 Levels with Inf_1^{+-}" or "Third scenario with probability opinion".
- The second measure of the third scenario, Inf_2^{+-}, (equation (3.32)), called "2 Levels with Inf_2^{+-}" or "Third scenario with belief opinion".

The reader can refer to the previous chapter for more details about the used "2 Levels" influence model and influence measures. We note that all the proposed influence measures can also be used with the "1 Level" model. We compared the proposed solutions to the

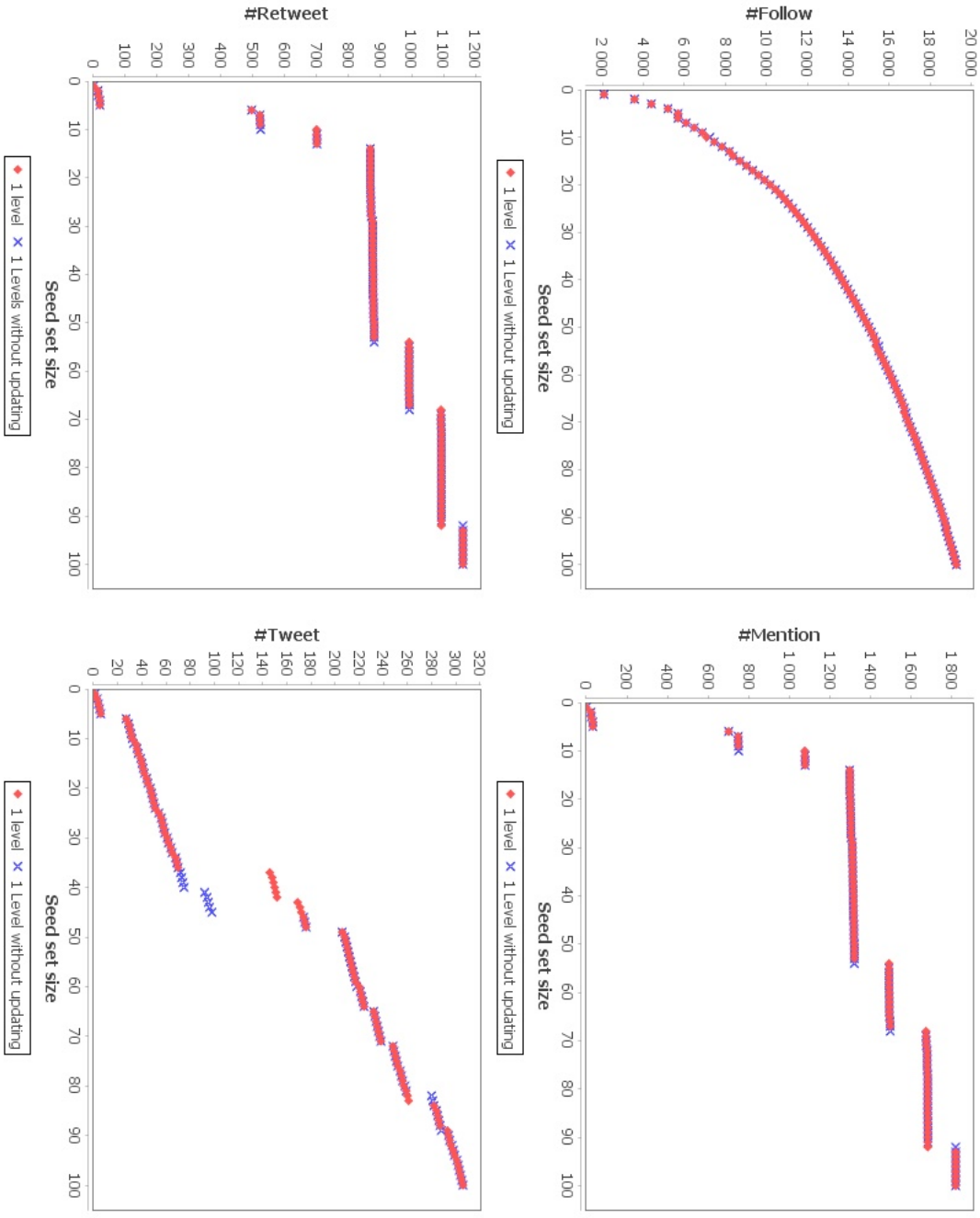


Figure 4.6: Impact of the weight updating step on influence maximization results: 1 Level

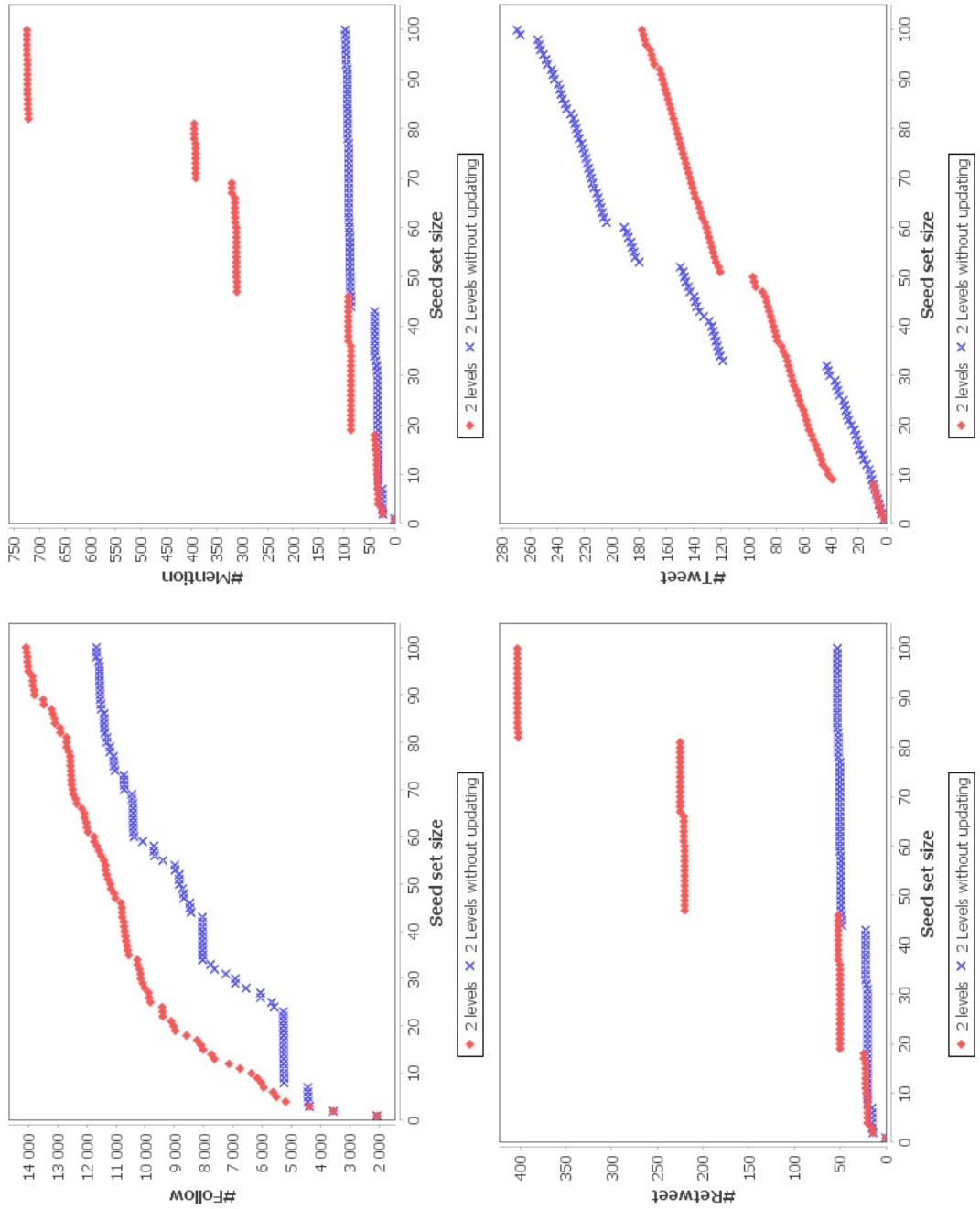


Figure 4.7: Impact of the weight updating step on influence maximization results: 2 Levels

Credit Distribution model (CD) [41] (refer to section 2.4.2 for more details) and to the Opinion-based Cascading (OC) model [99]. In the following experiments we fixed k to 50.

In a first experiment, we compare the number of common selected seeds as shown in Table 4.3. Then, we examine the number of common selected seeds between each couple of models. Indeed, the number of common selected seeds can be seen as a similarity indicator between influence maximization models. Then, it allows to know if there are some similarities between the experimented models. We notice that the Opinion-based Cascading model (OC) has no common seeds with any experimented model. Besides, CD model has no more than nine common seeds with "2 Levels" model that uses an influence measure on the set $\{Inf_1^+, Inf_2^+, Inf_1^{++}, Inf_2^{++}\}$. However, CD has only one common seed with "2 Levels" with $\{Inf_1^{+-}, Inf_2^{+-}, Inf\}$. Furthermore, "2 Levels with Inf" has a little number of common seeds with other experimented models. However, we notice that we have at least 34 common seeds between any couple of models from "2 Levels" with any influence measure from the set $\{Inf_1^+, Inf_2^+, Inf_1^{++}, Inf_2^{++}\}$. Besides, "2 Levels with Inf_2^{+-}" and "2 Levels with Inf_1^{+-}" have 47 common seeds. We explain these observations by the fact that the used opinion-based influence measures are similare because all of them are based on the evidential influence measure, Inf.

In a second experiment, we compare the mean positive and negative opinions of the selected seeds and their neighbors using each maximization model as shown in Table 4.4. This experiment allows the evaluation of each experimented model in terms of the opinion of selected seeds and their neighbors. In Table 4.4, "2 Levels" model that uses the evidential influence measure, Inf, selects influencer spreaders that have a moderate positive and negative opinion, about 0.3. This fact is expected, because Inf does not consider the user's opinion. Besides, the CD model chooses influencers that have a small value of positive and negative opinion, about 0.01, which proves that it is not adaptable for this purpose. In fact, CD does not consider the user's opinion [48]. However, the OC model selects seeds with about 0.41 of mean positive opinion which still an unsatisfactory result for a model that considers the user's opinion about the product.

In another hand, when we consider the user's opinion in the "2 Levels" model, we notice better results in "mean positive opinion" and "mean negative opinion" of selected seeds. Indeed, "2 Levels" that uses an influence measure from the set $\{Inf_1^+, Inf_2^+, Inf_1^{++}, Inf_2^{++}\}$, selects seeds having at least about 0.82 of "mean positive opinion" and at most about 0.08 "mean negative opinion" which is a very good result against the results of existing models (CD and OC). In Table 4.4, we notice that the best maximization model in terms of mean positive and negative opinion is "2 Levels with Inf_2^+". In fact, it gives a maximum value of "mean positive opinion", that equals to 0.85 ± 0.06 (0.95 confidence interval), and a minimum value of "mean negative opinion", that equals to 0.06 ± 0.03. Furthermore, we observe that all results of "2 Levels" model with any influence measure from the set $\{Inf_1^+, Inf_2^+, Inf_1^{++}, Inf_2^{++}\}$ are very near to each others, this observation is explained

Table 4.3: Seed sets intersection

	OC	CD	2 Levels with Inf	2 Levels with Inf_2^{+-}	2 Levels with Inf_1^{+-}	2 Levels with Inf_2^{++}	2 Levels with Inf_1^{++}	2 Levels with Inf_2^{+}	2 Levels with Inf_1^{+}
2 Levels with Inf_1^{+}	0	9	9	17	17	40	35	38	50
2 Levels with Inf_2^{+}	0	7	7	15	15	34	42	50	
2 Levels with Inf_1^{++}	0	9	9	18	18	40	50		
2 Levels with Inf_2^{++}	0	8	8	18	18	50			
2 Levels with Inf_1^{+-}	0	1	16	47	50				
2 Levels with Inf_2^{+-}	0	1	13	50					
2 Levels with Inf	0	1	50						
CD	0	50							
OC	50								

Table 4.4: Mean opinions of selected seeds and their neighbors

Model	Mean positive opinion	Mean negative opinion	Mean positive neighbors opinion	Mean negative neighbors opinion
2 Levels with Inf_1^+	0.83 ± 0.07	0.07 ± 0.03	0.39 ± 0.06	0.21 ± 0.06
2 Levels with Inf_2^+	0.85 ± 0.06	0.06 ± 0.03	0.40 ± 0.06	0.21 ± 0.06
2 Levels with Inf_1^{++}	0.80 ± 0.07	0.08 ± 0.03	0.42 ± 0.07	0.20 ± 0.07
2 Levels with Inf_2^{++}	0.82 ± 0.07	0.07 ± 0.03	0.42 ± 0.07	0.20 ± 0.07
2 Levels with Inf_1^{+-}	0.59 ± 0.06	0.18 ± 0.03	0.39 ± 0.06	0.22 ± 0.06
2 Levels with Inf_2^{+-}	0.62 ± 0.06	0.17 ± 0.03	0.39 ± 0.06	0.22 ± 0.06
2 Levels with Inf	0.30 ± 0.04	0.30 ± 0.02	0.39 ± 0.06	0.21 ± 0.06
CD	0.01 ± 0.02	0.01 ± 0.01	0.24 ± 0.05	0.13 ± 0.05
OC	0.41 ± 0.08	0.20 ± 0.04	0.35 ± 0.08	0.21 ± 0

in Table 4.3 where we find that they have many common seeds.

The neighbors positive and negative opinion are now considered. We find that the best "mean positive opinion" value of seeds neighbors is given by "2 Levels" with Inf_1^{++} and Inf_2^{++}. Indeed, we have got 0.42 ± 0.07 which is the highest value compared to those given by the other proposed influence measures, CD and OC models. This observation can be explained by the fact that Inf_1^{++} and Inf_2^{++} consider the positive opinion of the user's neighbors while estimating the influence. By the same way, we notice that "2 Levels" model with Inf_1^{+-} and Inf_2^{+-} detects seeds with highest "mean negative opinion" of seed's neighbors. In fact, they give a value of 0.22 ± 0.06 which is the maximum value in the last column of the Table 4.4. This fact is explained by the consideration of the negative opinion of the user's neighbors while estimating the influence.

In a last experiment, we compare all experimented models in terms of #Follow, #Mention, #Retweet and #Tweet. This experiment is useful to study and compare the quality of selected seeds using each experimented model. As a result, we have got curves presented in Figure 4.8.

In Figure 4.8, we have four sub-figures in which we present the accumulated #Follow, #Mention, #Retweet and #Tweet respectively. In the accumulated #Follow figure, we notice that all experimented models selected seeds that are followed by many other users except CD and OC models that their seeds do not exceed ten followers in all. Besides, the "2 Levels" model, the "Third scenario with probability opinion" and the "Third scenario

with belief opinion" give almost the same results that are up to 12000 accumulated follow. Furthermore, the results of the first and the second scenarios are very similar and are up to about 9000 accumulated follow.

In a second sub-Figure of Figure 4.8, we have the accumulated #Mention curves. We observe that OC and CD models do not select mentioned seeds and their accumulated #Mention values do not exceed twenty in all. Furthermore, the "2 Levels" model that uses an opinion-based measure (the three scenarios) have better results in terms of accumulated #Mention than "2 Levels" model that uses the evidential influence measure Inf. Besides, the "second scenario with probability opinion" and "the "second scenario with belief opinion" have the best results between all the experimented models in terms of accumulated #Mention. Indeed, they reach over 1100 #Mention from about the twentieth selected seed. From the results of this sub-Figure, we can conclude that the incorporation of the user's opinion in the process of the influence maximization ameliorates the quality of selected seeds in terms of accumulated #Mention.

In a third sub-Figure of Figure 4.8, we study the quality of selected seeds by each experimented model in terms of accumulated #Retweet. We observe that selected seeds using OC or CD models are not retweeted a lot. In fact, their curves do not exceed fifty accumulated #Retweet. In addition, we notice a similar behavior of the proposed influence measures to the accumulated #Mention curves. In fact, we see that the three proposed scenarios have succeed in selecting seeds having a high accumulated #Retweet. Also, the "second scenario with probability opinion" and the "second scenario with belief opinion" give the best results in terms of accumulated #Retweets.

In the last sub-Figure, we study the quality of the selected seeds in terms of accumulated #Tweet. In this sub-Figure we observe a different behavior of OC model. In fact, it succeeds to select some active users in terms of #Tweet. However, it does not reach the activity level of seeds detected by the proposed influence maximization solutions. Besides, we notice that CD model does not exceed twenty accumulated #Tweet in all. In another hand, we notice that the proposed influence maximization solutions have the same shape. Also, the "second scenario with probability opinion" and the "second scenario with belief opinion" detect the best seeds in terms of accumulated #Tweet. Besides, we observe that curves of "2 Levels" model with an opinion-based influence measure exceed the curve of "2 Levels with Inf".

In this section, we present some interesting experiments using real world data to study the behavior of the proposed influence maximization solutions when we incorporate the user's opinion about the product. Our experiments show the performance of the proposed approach. In fact, we notice a good improvement in the quality of selected seeds not only in terms of the opinion about the product but also in terms of #Follow, #Mention, #Retweet and #Tweet. In the next section, we present a set of experiments to study the accuracy of the proposed approach.

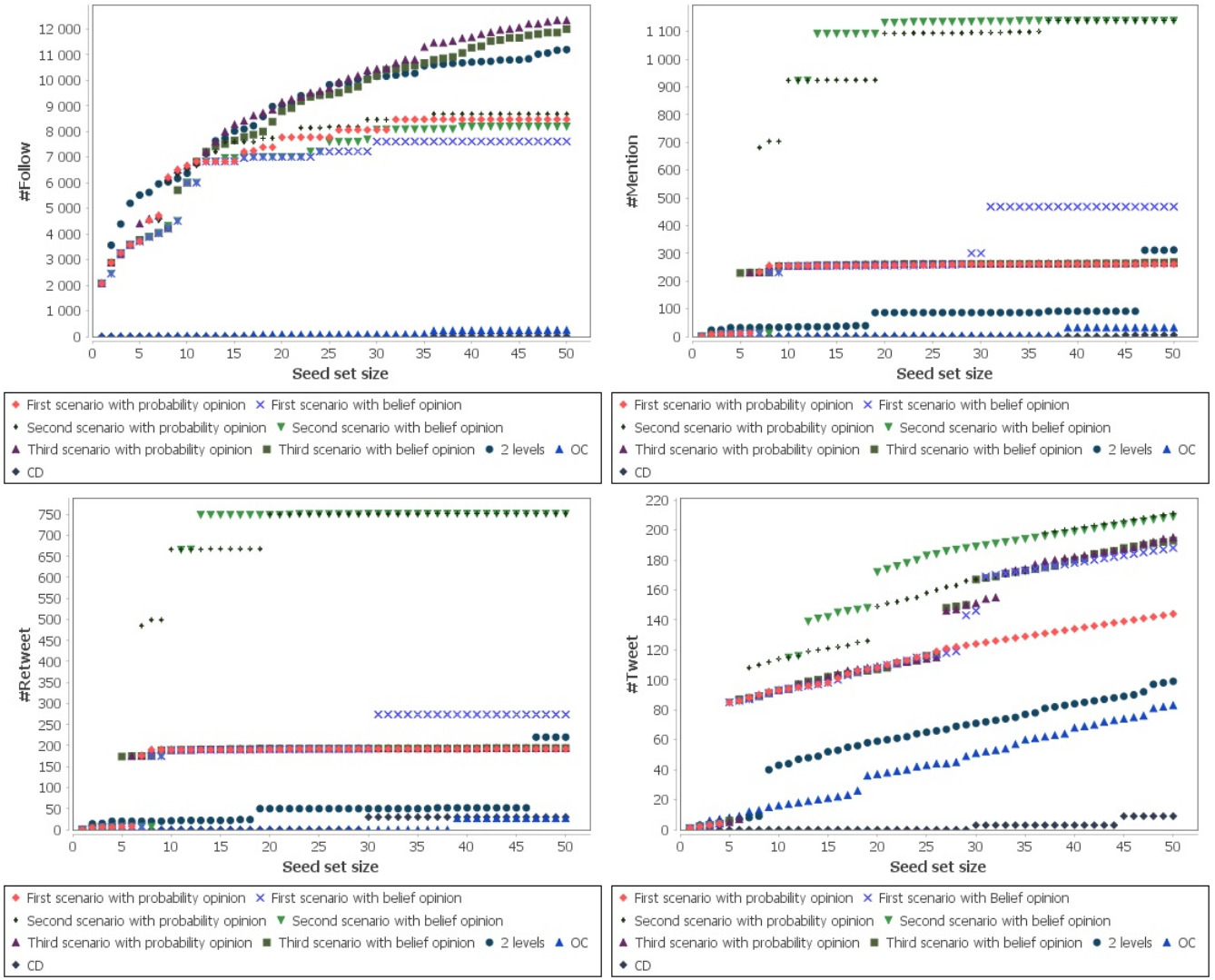


Figure 4.8: Comparison between the opinion based scenarios, the second influence model and the OC model

4.5 Studying the influence behavior on generated data

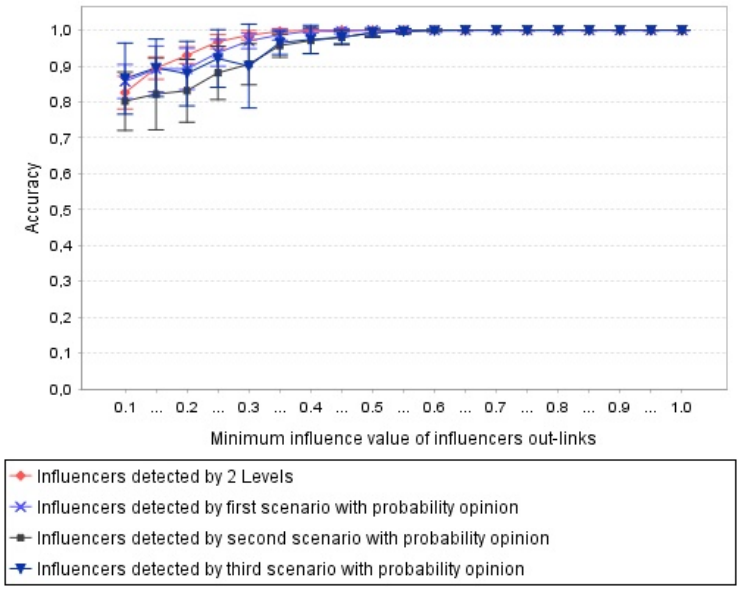
In this section, we use the generated dataset introduced in section 4.2.2 in order to study the behavior of the proposed influence maximization solution while varying influence and user's opinion. In these experiments, we fix the size of the seed set k to 50 and we repeat the random process twenty times. As we said above in section 4.2.2, the process used to generate the data is parameterizable. Then, in our experiments, we vary each parameter and we fix the others in order to study the accuracy of the proposed influence maximization solutions. Finally, we experiment the following influence measures with the "2 Levels" model:

- The evidential influence measure, Inf, (equation (3.23)), called "2 Levels",
- The first measure of the first scenario, Inf_1^+, (equation (3.24)), called "First scenario with probability opinion",
- The first measure of the second scenario, Inf_1^{++}, (equation (3.26)), called "Second scenario with probability opinion",
- The first measure of the third scenario, Inf_1^{+-}, (equation (3.30)), called "Third scenario with probability opinion".

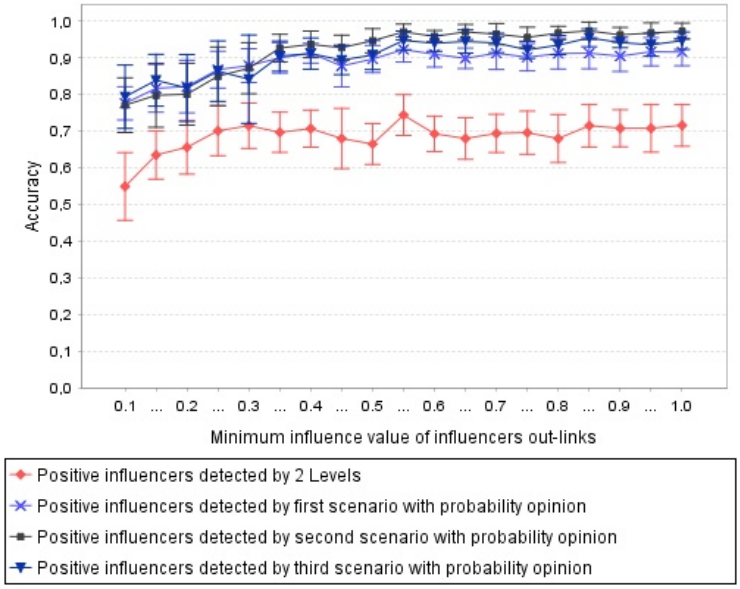
In a first experiment, we vary the minimum influence parameter and we study its impact on detecting influencers and positive influencers as shown in Figure 4.9. We fix the minimum positive opinion of positive influencers to 0.8, the minimum positive and negative opinion of positive influencers neighbors to 0.3 and 0.8 respectively. Figure 4.9a shows the accuracy of detecting influencers by the experimented models while varying the minimum influence value. This figure shows the performance of the proposed models. In fact, even with a small influence value, 0.1, the experimented models succeed in detecting influencers with a good accuracy that is no less than 80%. Besides, we notice that the "2 Levels" model starts having the highest accuracy from the influence value 0.15 until the value 0.6 from where all other models start having an accuracy equals to 1.

In Figure 4.9b, we study the accuracy of detecting positive influencers while varying the "minimum influence" value. In this figure, we observe that the first, the second and the third scenarios give good accuracies of detecting seeds having a positive opinion. Besides, we notice a natural behavior of the "2 Levels" model that does not consider the opinion in its principle, but it keeps giving acceptable accuracies.

In a second experiment, we vary the "minimum positive opinion" value of influencer users. In this experiment, we fixed the minimum influence value to 0.5, the minimum positive and negative opinion of positive influencers neighbors to 0.3 and 0.8 respectively. Figure 4.10 presents the accuracy of detecting influencers having a positive opinion by the mean of each experimented model. In this figure, we notice a similar results to those of the



(a) Accuracy of detected influencers while varying the minimum influence value



(b) Accuracy of detected positive influencers while varying the minimum influence value

Figure 4.9: Accuracy variation while varying the minimum influence value

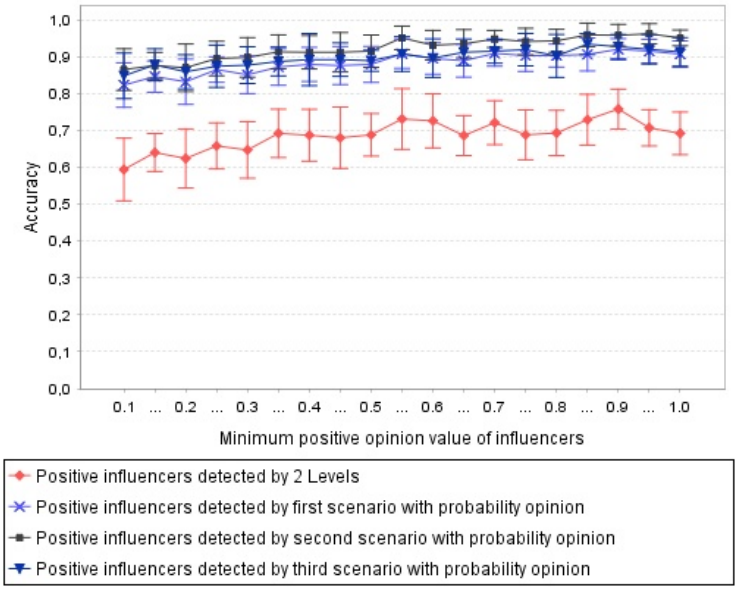


Figure 4.10: Accuracy of detected positive influencers while varying the minimum positive opinion value

Figure 4.9b. In fact, all curves are almost steady. Besides, the best accuracies are given by the second, the third and the first scenarios models respectively.

In a third experiment, we vary the "minimum positive opinion of positive influencers neighbors". In this experiment, we fixed the minimum influence value to 0.4, the minimum positive opinion of influencers to 0.5 and the minimum negative opinion of positive influencers neighbors to 0.8. Figure 4.11 shows the accuracy of detecting positive influencers that exert more influence on positive users. In this figure, we notice a different behavior from the previous figures. In fact, all curves increase gradually when the minimum positive opinion value of positive influencers neighbors increases until getting high accuracies.

In a last experiment, we vary the "minimum negative opinion of positive influencers neighbors". Besides, we fix the minimum influence value to 0.4, the minimum positive opinion of influencers to 0.5 and the minimum positive opinion of positive influencers neighbors to 0.2. Figure 4.12 shows the accuracy of detecting positive influencers that exert more influence on positive and negative users while varying the minimum negative opinion value of positive influencers neighbors. In the first sub-figure 4.12a, we have the accuracy of detected positive influencers influencing negative users. All curves increase when the varied value increases. Besides, the best accuracies values are given by the third scenario which is dedicated to positive influencers that exert more influence on negative users. In the second sub-figure 4.12b, we have the accuracy of detected positive influencers influencing positive users. In this figure, we observe a reverse behavior of curves in Figure 4.12a. In fact, the accuracy decreases when the varied value increases. This behavior is explained by the fact that, when the number of positive influencers influencing negative users increases, the number of those

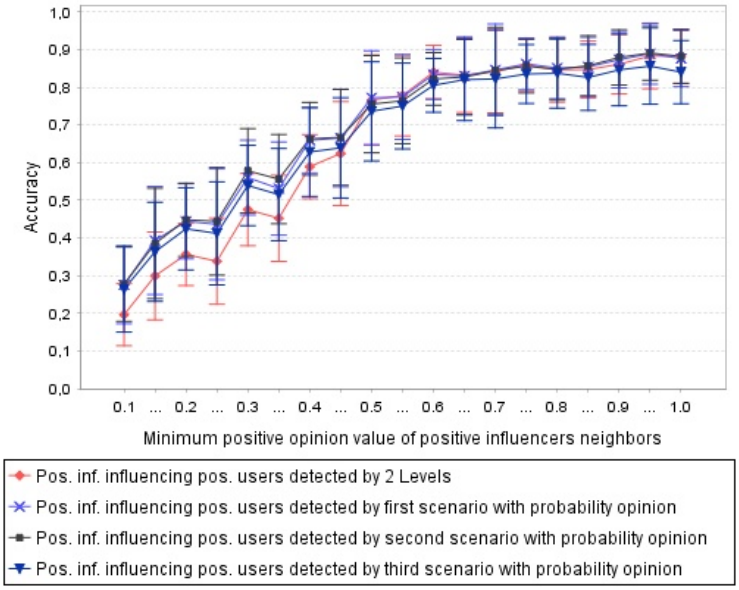


Figure 4.11: Accuracy of detected positive influencers influencing positive users while varying the minimum positive opinion value of positive influencers neighbors

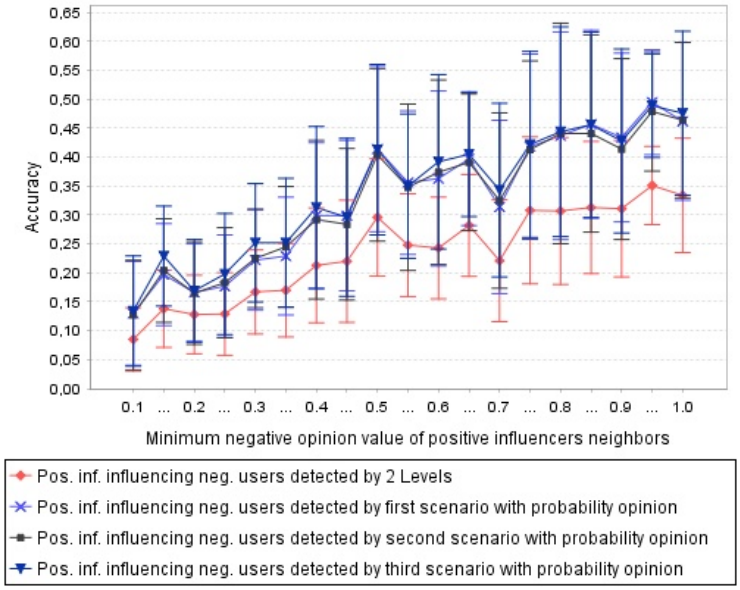
influencing positive users decreases.

To sum up, in this section, we present some results made on generated data. These results show the performance of the proposed approaches. Besides, we notice that the results of the first, the second and the third scenarios are very similar. This behavior is justified by the similarity between their influence measure.

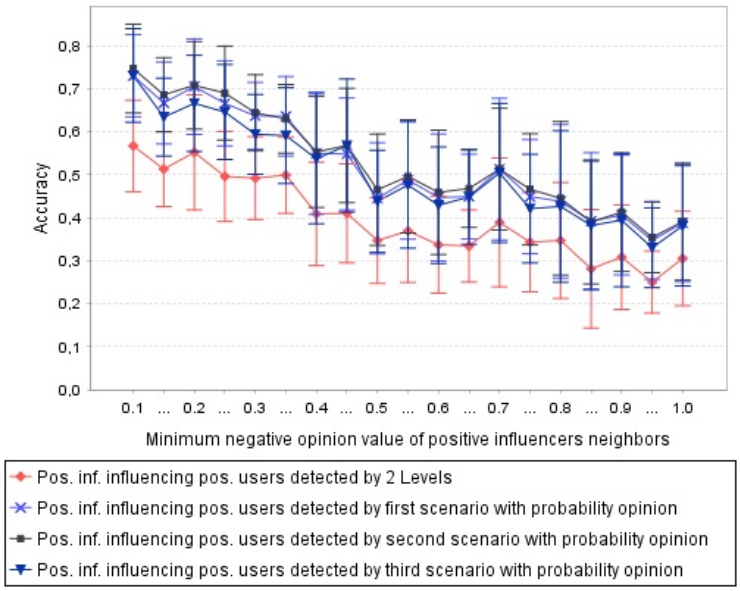
4.6 Conclusion

This chapter is mainly dedicated for experimental studies of the proposed influence maximization solutions that are introduced in Chapter 3. First, we present the used data and the process we apply to estimate the users opinion. Next, we make some experiments on real data, collected from Twitter, to compare the proposed solutions to some existing one. Finally, we study the accuracy of the proposed models on generated data.

The main conclusion from the presented experiments is that the proposed influence maximization solutions ameliorate the quality of selected seeds when compared with existing influence maximization solutions in terms of #Follow, #Mention, #Retweet and #Tweet. This founding makes the proposed influence maximization models very useful to promote a given Viral Marketing campaign. Furthermore, a second important conclusion is about the user's opinion about the product. In fact, our experiments show the importance of this parameter and its contribution to the improvement of the quality of selected influencers not only in terms of the influence criteria, *i.e.* #Follow, #Mention, #Retweet and #Tweet, but



(a) Accuracy of detected positive influencers influencing negative users while varying the minimum negative opinion value of positive influencers neighbors



(b) Accuracy of detected positive influencers influencing positive users while varying the minimum negative opinion value of positive influencers neighbors

Figure 4.12: Accuracy of detected positive influencers influencing positive and negative users while varying the minimum negative opinion value of positive influencers neighbors

also in terms of the opinion of selected seeds. Indeed, we succeed in detecting seeds having a positive opinion about the product.

In the next chapter, we consider another interesting problem that is related to Viral Marketing too. It is the problem of social messages classification in online social network. In fact, we need to know the topics to which each user is interested to. Such an information is helpful to distinguish social influencers by topics of interests.

5

Classification of the social message propagation

Contents

Summary

In chapters 3 and 4, we focus on the problems of measuring and maximizing influence in a social network for Viral Marketing perspectives and we propose some solutions for these problems. In this chapter, we consider another important problem. In fact, we present the solutions we propose to resolve the problem of classifying social messages without any need for access to their content. Also, we introduce a model of information propagation simulation in a social network that consider the message to be propagated while propagating it. This work is published in Jendoubi et al. [51, 50].

5.1 Introduction

The information propagation is a well-known problem, in which we try to simulate the dissemination process used by a given information to go from one user to another through social links. The information propagation simulation is an important task that searches generally to study and understand the propagation process or to estimate the user's influence. In section 2.2, we present an overview of the existing works that tries to simulate such a process. However, we notice that existing information propagation models like ICM and LTM do not consider the propagated content in the propagation process. They just suppose having an information and they simulate its propagation traces. In this chapter, we propose an information propagation algorithm that considers the class of the message while propagating it [51].

A second problem handled in this chapter is the problem of social message classification. We studied the existing classification approaches (the reader can refer to section 2.5 for more details about existing solutions) and we find that these solutions still not yet as efficient as needed. In fact, the social message is characterized by its shortness which causes a real problem for existing classification approaches. This fact justifies the need for classification solutions that resolve this problem. In this chapter, we introduce a classification approach that do not need the content of the message in its classification process [51, 50].

The main contributions presented in this chapter are the following: first, we propose an information propagation model that considers the class of the propagated message while propagating it. This model is used to simulate the propagation traces of a given type of message and to create a dataset of propagation networks (see the definition in section 5.2). A second contribution is that we introduce two model-based classifiers for social messages. The proposed classifiers do not need the content of the message. Then, any type of content can be classified. We just need its propagation traces [51]. Besides, this solution is more adaptable for discrete types of links, *e.g.* a type of link may be "friendship" for example. A third contribution is that we propose the Dynamic Time Warping distance for propagation networks (PrNet-DTW) [50]. The proposed distance has two advantages. In fact, it works with any type of links and it considers the fact that the paths in the propagation network are time dependent. Next, we used the proposed PrNet-DTW distance with the probabilistic and the evidential k-Nearest Neighbors algorithms to classify the propagation traces of social messages.

The remainder of this chapter is organized as follows: Section 5.2 is a glossary of the concepts we use in this chapter. Section 5.3 presents the information propagation algorithm we propose to simulate the propagation of a specific message. Section 5.4 and Section 5.5 introduce the solutions we propose for the problem of social message classification. Finally, Section 5.6 is dedicated for the experiments and results that are made on real and generated data.

5.2 Definitions

In this section, we define some concepts we use in this chapter to present the proposed algorithms.

Definition 7. *Homogeneous social network*: is a social network that is composed by one type of links and one type for nodes. For example, is such a network, nodes may be simple users and links are the friendship relation between them.

Definition 8. *Heterogeneous social network*: is a social network that is composed of several types of links and/or nodes. In fact, in real word social networks, we can find many types of nodes, *e.g.* users, groups, applications, etc. These nodes are connected to each other *via* many types of social links, *e.g.* friendship, membership, colleague, etc. For instance, Figure 5.1a shows a small heterogeneous social network where nodes are either users, a company, application or group of users and links are friendship, professional link, member link, uses and works in. In this chapter, we consider a heterogeneous social network where we have one type of nodes, *i.e.* users, and several types of links. Let's take the example of Figure 5.1b, in this example nodes are users and we have many possible links between them like: friendship, professional link, undefined link and familial link.

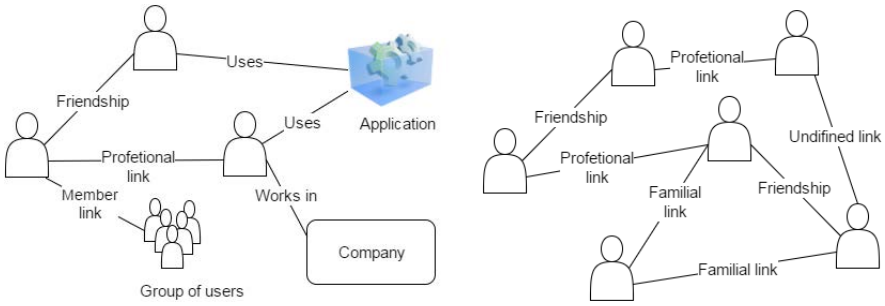
Definition 9. *Social message*: is a message that is sent or published through an online social network. For instance, we find status on Facebook, tweets on Twitter, comments, private messages, etc.

Definition 10. *Propagation strategy*: we mean by propagation strategy the way with which a specific message can propagate from one user to another. Each specific type of message has a specific propagation strategy. It is defined by a probability distribution on types of links for each message class. In fact, it gives to each possible type of links a propagation probability value in the range $[0, 1]$.

Definition 11. *Tendency of a node to propagate a message*: is a propagation parameter. It models the fact that a given node can choose to, either, propagate the message to a subset of its neighbors or not to propagate it.

Definition 12. *Propagation network (PrNet)*: is a graph based data structure that is used to store propagation traces of a given message. It has two main characteristics that distinguish it from an ordinary Directed Acyclic Graph (DAG): its edges are weighted by the type of the relationship between users. Second, its paths are time dependent. For instance, Figure 5.2 presents an example of a propagation network where nodes are network users and directed links model the propagation direction.

Definition 13. *Propagation level*: in the propagation network, we call propagation level the number of links in the path that separates the source of the message and the target node. For example, Figure 5.2 presents a propagation network with two propagation levels.



(a) Many types of nodes and links (b) Many types of links

Figure 5.1: Examples of heterogeneous social networks

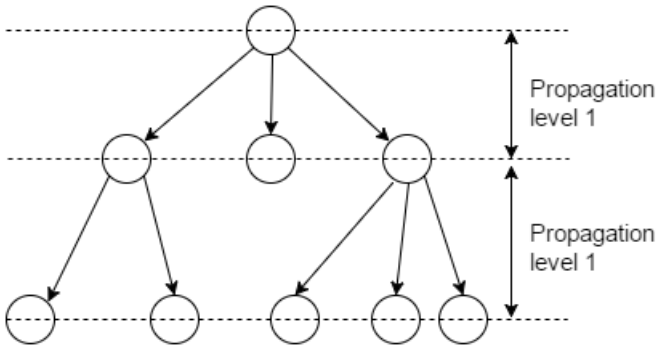


Figure 5.2: Propagation network and propagation levels

Definition 14. *Classification noise*: it is a small biases that may appear between the real class of the message and the observed one. It may leads to a misclassification of the message.

Other definitions related to the theory of belief functions and graph theory can be found in Appendices A and B respectively. In the next section, we introduce the proposed information propagation model.

5.3 Proposed information propagation model

The information propagation in a social network is the process with which the information spreads between network users by reaching one user from another following their relationships in the network. Modeling and simulating the information dissemination process is the challenge of many researchers. In this section, we propose a new information propagation model that fits more with existing real world social networks. The contributions of the proposed model are, mainly, the following:

- We take into account the message class while simulating its propagation. In fact, we assume that each class of messages has some specific propagation characteristics. For instance, a professional message propagates, generally, through professional relations.

- We define a model that simulates the propagation of a social message in a user-to-user network where it is possible to have many types of relationships between its nodes. For instance, a possible relationship may be a professional link or a familial relation.

The main purpose of the proposed algorithm is to simulate the propagation process of a given message and to generate an associated propagation network. To run the algorithm, we need:

1. *Heterogeneous social network*: the algorithm works with a network in which nodes are users and links model the type of the relationship between them. The reader can refer to Figure 5.1b for an example of a heterogeneous network.

2. *Propagation strategy for each possible class of messages*: as the proposed algorithm considers the class of the message in the propagation process, we define a specific propagation strategy in terms of types of links for each message class. The propagation strategy of a given message class can be either defined by an expert or learned from real world propagation.

3. *Class of the message to be propagated*: we need the message class to select an appropriate propagation strategy.

4. *Source of the message that will start the propagation*: it is the node that will trigger the propagation cascade of the message, it is possible to select a random node.

5. *Stopping condition*: in real world propagation, the time is the very common stopping condition of a given message propagation. Indeed, after a period of time the propagation stops. In our case, we define the number of iterations of the algorithm, η, as a stopping condition. The variable η models the number of time instants of the propagation process that the algorithm will consider.

Another important propagation parameter is the tendency of a given node to propagate a message. This parameter is mainly defined to model the intention of a particular node to send the message to a subset of its neighbors or to not send it.

The Algorithm 4 presents the outlines of the proposed information propagation algorithm. It starts by adding the source of the message, $Source$, to the list $ReadyNodes$ that contains nodes having received the message and that will try to propagate it. Next, the algorithm loops until achieving the stopping condition and at each iteration it loops on the ready nodes. For each node in $ReadyNodes$ list, the algorithm tests if the node wants/ready to propagate the message across a boolean random variable that is defined for each node in the network. If the *node* is ready to propagate the message, the algorithm loops on each possible type of link in the network. For each type of link, the algorithm estimates the number e of *node* neighbors that will receive the message. Then it chooses them randomly

from *node* neighbors that are related to it *via* a *LinkType* link. After running through all types of links and choosing the list of neighbors that will receive the message, the algorithm updates the propagation network of the message and the list of ready nodes.

Algorithm 4: Information propagation algorithm

```
begin
    ReadyNodes.add(Source);
    // Source: the source of the message
    // N: number of iterations (stopping condition)
    for i = 1 to η do
        for j = 1 to ReadyNodes.size() do
            node ← ReadyNodes.get(j);
            if node.propagate = true then
                foreach LinkType do
                    e ← node.outdegree() * node.propagationTendency() *
                    Str.linkTypeProportion(); R ← node.randomSelection(e, LinkType);
        PrNet.refine(R);
        R1.addAll(R);
        ReadyNodes.addAll(R1);
        R1.clear();
```

In this section, we introduce a new information propagation algorithm. The proposed algorithm can be used for many purposes like:

- The simulation of the propagation traces of a given type of messages: in this chapter, we mainly use the proposed algorithm for this purpose. Indeed, in our experiments, we use the information propagation algorithm in order to create a training and a testing corpus that will be used to evaluate the performance of our social messages classification approaches.

- The study of the propagation process of each type of messages: in this task, we are looking for understanding the process with which the information goes from one user to another in the network.

- The estimation of the user's influence in the network: in fact, we can assume the fact that *"more the user is influencer, more his messages propagate through the network"*. Then, we can attribute more influence amount to users that are able to make their messages propagate and reach more other users. Also, a given user can be influencer for a given type of messages. Then, in such a case we can define an amount of influence for each specific type of messages.

In the next two sections, we consider the problem of social message classification and we present four algorithms that do not use the content of the message to classify it.

5.4 Classification of propagation networks

In this section, we introduce two classification algorithms adapted for social messages that are published in *Jendoubi et al.* [51]. The first algorithm is totally probabilistic while the second one incorporates the theory of belief functions. The proposed algorithms are useful to classify the propagation network of the message instead of its content. Our work is motivated by the fact of the inefficiency of existent text classification approaches with short texts. Indeed, this is due to the lack of word occurrence in the message. Besides, text processing techniques always need a pre-processing step in which it is necessary to remove URLs, stop words, questions, special characters, etc. When working with tweets, for example, after the pre-processing step, it falls very often on empty messages. Those empty messages can not be classified by a text based classification technique. The proposed approach does not suffer from such a problem. In fact, it does not need the access to the content of the message in order to classify it. We just need its propagation traces. Another advantage of our approach is that it can be used with any content of social messages, *i.e.* text, image, video, etc.

The proposed classifiers are composed of two main steps. In the first step, we define a set of probability or basic belief assignment (BBA) distributions for each possible class of messages. We learn these distributions using a training set of existent propagation traces. The choice of the distribution (probability or BBA) depends on the used classification algorithm (the probabilistic one or the evidential one). The second step is the classification step. This process can be used if we want to classify a new message that we have its propagation traces. In this section, we detail the two steps of the proposed classifiers.

5.4.1 Parameters learning

In the parameters learning step, we learn a model for each class of messages. The main role of this model is to represent the characteristics of the class in order to recognize it in the classification step. The parameters learning algorithm needs a training set that contains a set of propagation networks for each possible message class. We defined each classification model by a set of probability or BBA distributions, one distribution for each propagation level. We fix the number of propagation levels to be considered as input in the algorithm. In this section, we present the parameters learning step for the two proposed algorithms. Then, we highlight the difference between them. Algorithm 5 shows the main steps of the parameters learning algorithms [51].

The parameters learning process works in two main steps. The first step is called *effective computation*. In this step, the algorithm loops on all the propagation networks in the training set. For each propagation network, $PrNet$, the algorithm loops on its propagation levels. For each propagation level in $PrNet$, it computes the number of nodes that have

received the message across each type of links. All computed values are stored in a matrix structure, ψ, where in its lines we have the types of links, and in its columns we have the propagation levels. The second step is called *accrued effective calculation*. As indicated by its name, the algorithm runs through the matrix ψ starting from the second level until the last one. At each level, it sums the current effective with those of the one before. This computation is done in order to preserve the propagation history at each propagation level.

When the accrued effective is calculated for all propagation levels, the algorithm moves to the *computation of probability and BBA distributions*. We note that if we work with the probabilistic classifier, then, the parameters learning algorithm stops in the instruction of probabilities computation, *i.e.* it computes only $ProbaSet$. Besides, if we work with the evidential classifier, we need to run the parameters learning algorithm until its end, *i.e.* the computation of $BbaSet$. $ProbaSet$ is the output of the function $ProbabilitiesCalculation(\psi)$. This function takes as input the structure ψ and for each propagation level it transforms the accrued effective to a probability distribution defined on the types of link. The transformation is done by dividing each value by the sum of the accrued effective of its level. To estimate the $BbaSet$ the algorithm applies the consonant transformation. More details about the consonant transformation and the theory of belief functions can be found in Appendix A.

Algorithm 5: Parameters learning algorithm

begin
  `// Effective computation`
  foreach $PrNet$ *in* $PrNetSet$ **do**
    foreach $Level$ *in* $PrNet$ **do**
      foreach $LinType$ **do**
        $\psi(TypeLink, Level) \leftarrow \psi(TypeLink, Level) + ComputeNodes(TypeLink)$;
  `// Accrued effective calculation`
  for $Level = 2$ *to* $NbrLevels$ **do**
    foreach $TypeLink$ **do**
      $\psi(TypeLink, Level) \leftarrow \psi(TypeLink, Level) + \psi(TypeLink, Level - 1)$;
  `// ProbaSet and BbaSet computation`
  $ProbaSet \leftarrow ProbabilitiesCalculation(\psi)$;
  $BbaSet \leftarrow ConsonantTransformation(ProbaSet)$;

5.4.2 Classification

The next step of the social message classifiers is the classification step. It uses the outputs of the parameters learning step as a classifier. In fact, it compares the propagation network of a new coming message to the class of messages model. Then, it attributes to the message the class of the model that fits more to its propagation network. In this section, we present the classification step of the probabilistic and the evidential classifiers together as there is a

similarity between them, then we highlight the difference between them. Algorithm 6 shows the main steps of the classification algorithms [51].

The first step of the classification algorithm transforms the propagation network of the message to be classified, $PrNet$, to a set of probability or BBA distributions, the choice of the distribution depends to the used algorithm. This step is done using the parameter learning algorithm (algorithm 5) by running it on $PrNet$. In the next step, the algorithm loops on the considered classes. Then, for each class it loops on the propagation levels. For each possible class and for each propagation level, the algorithm computes the distance between the distribution of the message and the distribution of the class. Then, it stores these distances in a matrix structure as shown in algorithm 6. We note that if we are working with the probabilistic classifier, we compute the values of the structure $ProbaDist$, and if we are working with the evidential classifier we compute the values of the structure $BbaDist$. In the last step, the algorithm gives a class for each propagation level in the propagation network of the message. The class of the level is chosen as the nearest class, *i.e.* with minimum distance value, from the training set in that level.

Algorithm 6: Classification algorithm

```
begin
    (ProbaPr, BbaPr) ← ParameterLearning(PrNet);
    for i = 1 to NbrClasses do
        foreach Level do
            ProbaDist(i, Level) ← Distance(ProbaPr, Probasets(i));
            BbaDist(i, Level) ← Distance(BbaPr, BbaSets(i));
    foreach Level do
        ProbaClasses(Level) ← ClassMinDistance(ProbaDist(:, Level));
        BbaClasses(Level) ← ClassMinDistance(BbaDist(:, Level));
```

In the second step of the algorithm, we need to estimate the distance between the message to be classified and the classes in the training set. In the literature we find many useful distances like:

- The Chebyshev distance:

$$d_C\left(X_1, X_2\right) = \max_i abs\left(X_1\left(i\right) - X_2\left(i\right)\right) \tag{5.1}$$

- The Manhattan distance:

$$d_M\left(X_1, X_2\right) = \sum_{i=1}^{n} abs\left(X_1\left(i\right) - X_2\left(i\right)\right) \tag{5.2}$$

where n is the size of the vectors X_1 and X_2.

- The Euclidean distance:

$$d_E(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (X_1(i) - X_2(i))^2} \tag{5.3}$$

- The Jousselme distance [53] that is more adaptable for the evidential classifier because it considers the size of each focal element while computing the distance:

$$d_J(X_1, X_2) = \sqrt{\frac{1}{2} (X_1(i) - X_2(i))^T \underline{\underline{\Lambda}} (X_1(i) - X_2(i))} \tag{5.4}$$

  such that $\underline{\underline{\Lambda}}$ is a $2^n \times 2^n$ matrix and $\Lambda(A, B) = \frac{|A \cap B|}{|A \cup B|}$. In the case where X_1 and X_2 are not BBAs distributions, then, $|A \cap B| = 0$ if $A \neq B$.

In this section, we introduced a new classification approach for social messages. The contributions of the proposed approach are that it does not need any access to the content of the message to be classified, and it is a standard classification approach that can be used with any type of social contents. In the next section, we present an improvement of the solutions that are presented in this section.

5.5 Dynamic time warping distance and k-NN classifiers

In the previous section, we introduced two propagation network (PrNet) classifiers that are based on mathematical distances like the Euclidean distance and the Jousselme distance. This solution needs to transform the PrNet to a set of probability or BBA distributions. Then, it computes the distance between these distributions instead of PrNets. This transformation leads to a loss of information that may be significant in the classification step. Another drawback is that these classifiers do not work with continuous types of links and a discretization step is always needed in such a case. To overcome these problems, we introduce a solution in which we do not need any transformation for the propagation network. In this new solution, we propose a measure that quantifies the distance between two propagation networks, *i.e.* the distance between the PrNet to be classified and each PrNet in the training set, and we incorporate it in the probabilistic and the evidential k-NN classifiers. These solutions are published in *Jendoubi et al.* [50]. In this section, we present the new distance-based solution.

5.5.1 Proposed propagation network DTW distance

In section 5.2, we define the propagation network as a DAG that has two extra characteristics: its edges are weighted by the type of the relationship between users, and its paths are time

dependent. Our purpose is to classify the propagation network of the message without any transformation that may lead to the loss of the information. The solution that we propose is to use a distance based approach like k-NN. Then, we need a distance metric that measures the similarity between PrNets.

Graph similarity measures are used to evaluate the similarity or the distance between two graphs. In the literature, we find a lot of methods [18, 90, 36] that are used for this purpose. There is in particular the family of *graph edit distances*. The edit distance was, firstly, proposed by [62] to evaluate the distance between two strings. Then, it was adapted for graphs [89]. Graph edit distances have as purpose to determine the minimal cost to transform a graph to a target one. Then, it computes the number of needed edit operations (deletion, insertion or substitution) to transform the first graph into the second one. Also, it attributes a cost to each operation and finally it computes the distance between them. In [36], the authors present a survey for graph edit distances. *Maximal common sub-graph based distances* [19, 93] are, also, used to measure graph similarity. The idea behind these methods consists of finding the maximal common sub-graph between input graphs. The bigger the sub-graph is, more common are the graphs. Also, we find a graph similarity measure that is based on finding the *common minimal super-graph* [33].

However, all these distances do not consider the time dimension which is a character of the PrNet. Then, comes the need of proposing a distance that fits more to the characteristics of the weighted time dependent DAGs like the PrNet. As a solution to this problem, we propose the *Dynamic Time Warping* distance for propagation networks similarity (PrNet-DTW).

5.5.1.1 Dynamic Time Warping distance

The Dynamic Time Warping distance (DTW) [78] was first proposed to measure the similarity between two speech sequences. The advantage of this measure is that it considers the fact that the speech is time dependent. Recently, [75] propose to use it to measure the similarity between two time series in order to analyze satellite images. A time series or a sequence is a time ordered list of elements. DTW distance considers the order of appearance of each element in the time series while computing the distance between them. Let $TS1 = (b1_1, b1_2, \ldots, b1_{T1})$ and $TS2 = (b2_1, b2_2, \ldots, b2_{T2})$ be two time series. $DTW\,(TS1_i, TS2_j)$ is the DTW distance between the sub-sequences $TS1_i$ and $TS2_j$, and it is defined as [75]:

$$DTW\,(TS1_i,\, TS2_j) = \delta\,(b1_i,\, b2_j) + \min \begin{cases} DTW\,(TS1_{i-1},\, TS2_{j-1}) \\ DTW\,(TS1_i,\, TS2_{j-1}) \\ DTW\,(TS1_{i-1},\, TS2_j) \end{cases} \tag{5.5}$$

Note that $\delta(b1_i, b2_j)$ is a distance between the two elements $b1_i \in TS1$ and $b2_j \in TS2$. As mentioned in [75], the implementation of this recursive function leads to an exponential temporal complexity. To resolve this problem, authors propose to use the memoization technique as a solution to speed up the computation. Hence, we need a $|T1| \times |T2|$ matrix, Ξ, in which we record previous results in order to avoid their computation in next iterations. This computation technique maintains the time and space complexity of the DTW distance to $O(|T1| \times |T2|)$. Algorithm 7 shows the outlines of the DTW computation using the memoization technique [75]. The algorithm starts by estimating the values in the first column and the first line of the matrix Ξ. Then, it loops on the other elements, and estimate each one by using the smallest distance from the left value, the upper value and the diagonal one until reaching the element $\Xi[T1, T2]$. The value of $\Xi[T1, T2]$ represents the DTW distance between the two sequences taken as input of the algorithm.

Algorithm 7: DTW algorithm [75]

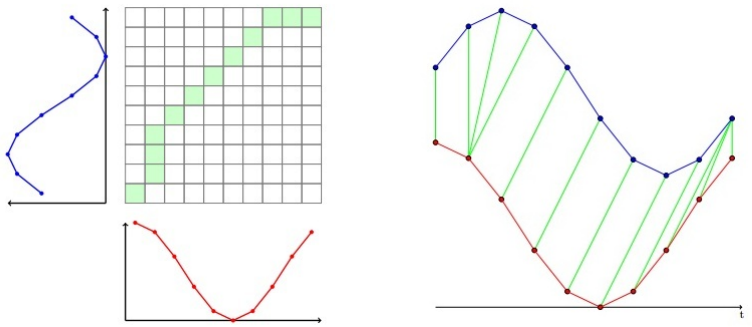
begin
- $\Xi[1,1] \leftarrow \delta(b1_1, b2_1)$;// Ξ is the cost matrix
- **for** $i = 2$ *to* $T1$ **do**
  - $\Xi[i,1] \leftarrow \Xi[i-1,1] + \delta(b1_i, b2_1)$;
- **for** $j = 2$ *to* T **do**
  - $\Xi[1,j] \leftarrow \Xi[1,j-1] + \delta(b1_1, b2_j)$;
- **for** $i = 2$ *to* $T1$ **do**
  - **for** $j = 2$ *to* $T2$ **do**
    - $$\Xi[i,j] \leftarrow \delta(b1_i, b2_j) + \min \begin{cases} \Xi(i-1, j-1) \\ \Xi(i, j-1) \\ \Xi(i-1, j) \end{cases};$$
- *return* $\Xi[T1, T2]$;

Example 13. Figure 5.3 shows a computation example of the DTW distance between two sequences and the alignment between them. As we see in the figure, the DTW algorithm coordinates each value in the first sequence with one or more values in the second sequence until achieving the last values and finding an alignment between the two sequences. □

5.5.1.2 Propagation Network DTW distance

The *Propagation Network Dynamic Time Warping* distance (PrNet-DTW) is used to measure the distance between two propagation networks by considering, simultaneously, the time dependencies of its paths and the weights defined on its links. The algorithm 8 shows the outlines of the PrNet-DTW algorithm. In the first step, it transforms each PrNet to a set of dipaths. A dipath is defined as a finite sequence of vertices connected with edges that are directed to the same direction.

Example 14. Figure 5.4 presents an example of a propagation network and its correspond-



(a) The matrix and the warping path (b) The alignment between the two sequences

Figure 5.3: Dynamic Time Warping distance example [75]

ing dipathes as needed by the proposed algorithm. As shown in the figure, all dipaths start from the source of the message and we have as dipaths as leafs in the network. □

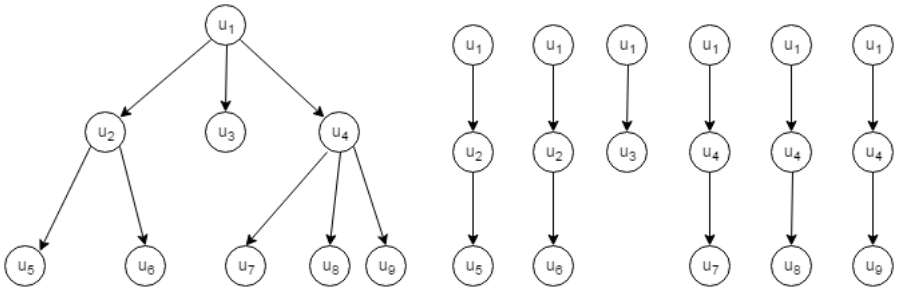
In the second step, the PrNet-DTW algorithm loops on $DipathSet1$. At each iteration, it fixes a Dipath and computes the DTW distance between the fixed Dipath and each Dipath in $DipathSet2$. Then, the algorithm takes the minimal DTW value to be the distance between the current Dipath and $PrNet2$. Finally, PrNet-DTW computes the mean of the minimal distances computed in the second step. The resulting value, $Distance$, is the PrNet-DTW distance between $PrNet1$ and $PrNet2$.

Algorithm 8: PrNet-DTW algorithm

```
begin
    DipathSet1 ← PrNet1.TransformToDipathSet();
    DipathSet2 ← PrNet2.TransformToDipathSet();
    for i = 1 to DipathSet1.size() do
        D ← maxValue;
        for j = 1 to DipathSet2.size() do
            D ← min(D, DTW(DipathSet1.get(i), DipathSet2.get(j)));
        Distance ← Distance + D;
    Distance ← Distance/DipathSet1.Size();
```

5.5.2 Classification with PrNet-DTW

To classify propagation networks of social messages, we need a distance-based classification approach to be used with the proposed PrNet-DTW distance. For this purpose, we choose the probabilistic and evidential k-NN algorithms because these classifiers are distance-based and they can be used together with the proposed PrNet-DTW distance to classify propagation traces of social messages. The second reason of this choice is the simplicity of implementation and use of these algorithms. In this section, we present two k-NN based



(a) Propagation network example (b) Dipathes example

Figure 5.4: Example of a propagation network and its dipathes

approaches which are the probabilistic k-NN and the evidential k-NN.

5.5.2.1 Probabilistic k nearest neighbors

The probabilistic k nearest neighbors (k-NN) is a well known supervised algorithm that is, generally, used for classification. It needs as input a set of training examples where their classes are known, and of course the object to be classified. Besides, we have to specify a measure of distance that is used to quantify the matching between the new object l and every object in the training set. First, k-NN starts by computing the distance between l and every object in the training set. Then, it selects the first k nearest neighbors, *i.e.* that have the shortest distance from l. Finally, the object l is classified according to the majority vote principle, *i.e.* the algorithm chooses the class that has the maximum occurrence count in the k nearest neighbors set to be the class of l. The k-NN technique is surveyed in [14].

Example 15. Figure 5.5 gives an illustration example of the k-NN algorithm. In the figure, when $k = 3$ the class of the new object is C_2. However, when $k = 5$ the class of the new object is C_1. □

5.5.2.2 Evidential k Nearest Neighbors

The evidential k-NN algorithm [27] is an extension of the probabilistic k-NN. The evidential k-NN uses the theory of belief functions in its classification step. As explained above, the probabilistic k-NN sorts the training examples according to their distances from the object l to be classified. Then, it chooses the k nearest neighbors to l. However, according to [27], the distance value between l and its nearest neighbors may be significant. The evidential k-NN differs from the probabilistic algorithm in the decision rule, *i.e.* the probabilistic algorithm classifies a new object using the majority vote principle, but the evidential algorithm uses, also, the distance values between the new object and its k nearest neighbors. Let $\Omega = \{C_1, C_2, ..., C_n\}$ be the set of all possible classes, it is the frame of discernment, and let d_j be the distance between l and the j^{th} nearest neighbor. The idea behind the evidential k-NN

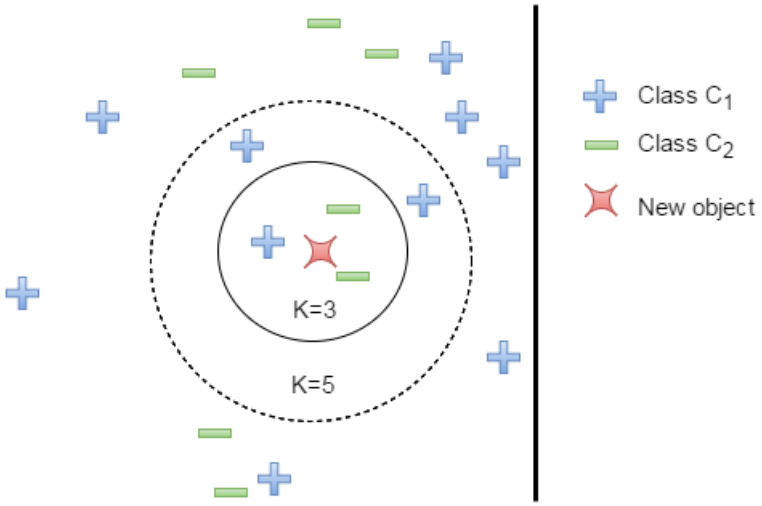


Figure 5.5: k-NN algorithm example

consists on representing each object of the k neighbors by a BBA distribution defined as:

$$m(\{C_i\}) = \chi \tag{5.6}$$
$$m(\Omega) = 1 - \chi \tag{5.7}$$
$$m(A) = 0 \, \forall A \in 2^C \setminus \{C_i\} \tag{5.8}$$

such that $0 < \chi < 1$. If d_j is big, then, the j^{th} nearest neighbor is considered as giving a little information about the class of l. In that case, χ has to take a small value. On the other hand, if the distance d_j is small, *i.e.* j^{th} nearest neighbor is near l, then, χ has to take a big value. According to this reasoning χ is a decreasing function of d_j. The function χ is defined as follows [27]:

$$\chi = \chi_0 \kappa_i(d_j) \tag{5.9}$$
$$\kappa_i(d_j) = e^{-\gamma_i d_j^{\beta}} \tag{5.10}$$

where $\gamma_i > 0$ and $\beta \in \{1, 2, \ldots\}$. After estimating a BBA distribution for each nearest neighbor, the evidential k-NN makes a decision about the class of l according to the following steps:

- It combines all BBA distributions using a combination rule.
- It applies the pignistic transformation [83] in order to obtain a pignistic probability distribution.
- It chooses the class that has the biggest pignistic probability to be the class of l.

In this section, we introduce the PrNet-DTW distance. PrNet-DTW is useful to measure the distance between two propagation networks which are acyclic directed graphs that have time dependent edges. Then, we propose to use the proposed distance to classify propagation networks by incorporating it in the probabilistic and the evidential k-NN algorithms. The next section is dedicated to the experiments. In fact, we test the proposed algorithms and show their performance in classifying the propagation networks of social messages.

5.6 Experiments and results

In this section, we present some experiments that show the performance of the proposed algorithms. We made our experiments using two datasets that were collected from Twitter and we made experiments on each one of them.

5.6.1 Datasets

In this section, we present two datasets and we explain the process we use to create them.

5.6.1.1 Twitter network data

Twitter network data is a dataset that was collected from Twitter using NodeXL[1] V 1.0.1.245 [45]. It is a free and open-source template for Microsoft Excel, allows many nice functionality like data import from social networks, networks metrics, graph visualization, etc. We obtained the network shown in Figure 5.6. It is a directed network in which nodes are Twitter users and links are the follow relationship between them. The network contains 97 vertices and 350 directed edges.

Twitter network dataset contains only the structure of the social network. However, we need also the types of the relationship between the network users. Then we defined the following four generic relations:

- "*Professional*" link, for professional relations like colleagues, office mates, etc.
- "*Familial*" link like sisters, brothers, cousins, etc.
- "*Friendly*" link to model the friendship.
- "*Undefined*" link to model the case were we do not know the kind of the relation.

Furthermore, for each link, we choose its type from the list defined above. Then, we obtain a heterogeneous social network we use as input of the proposed information propagation algorithm in order to generate a training and a testing sets for each type of message from the following:

[1] https://nodexl.codeplex.com/

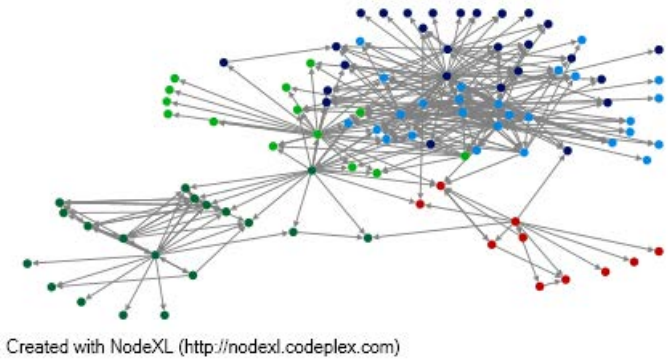


Figure 5.6: Network visualization

- "*Spam*": is a kind of malicious messages.
- "*Professional*": may be a kind of a product or service sent to an enterprise, an information sent between colleagues, etc.
- "*Familial*": is for messages sent between the members of a family.

For each class of messages we define a propagation strategy. Each strategy is a set of values in $[0, 1]$ defined on types of links and interpreted as "*the proportion of the node neighbors that will receive the message from the type of link*". Then each strategy is composed by four proportions as we have four types of links.

Example 16. Let define a strategy of a professional message as follows:

$$\{(Professional,\ 0.7),(Familial,\ 0.25),(Friendly,\ 0.25),(Undefined,\ 0.1)\} \tag{5.11}$$

□

To be as near as possible to the reality, we disrupt each defined strategies using a noise rate, *i.e.* see the definition of the classification noise in section 5.2. The noise value can be added to or removed from the proportions of the of strategy.

Example 17. Let consider a noise rate of 0.1 in the strategy defined in the previous example:

$$\{(Professional,\ 0.8),(Familial,\ 0.35),(Friendly,\ 0.15),(Undefined,\ 0.2)\} \tag{5.12}$$

□

We fix the number of propagation levels in the propagation network to three (the stopping condition of the propagation algorithm is three iterations). Then we run the proposed propagation algorithm to create a training set for each propagation strategy containing a set of 100 propagation networks. Also, we created a testing set of size 100.

5.6.1.2 Real propagation data

To obtain this second dataset, we implemented a crawler for Twitter using the java library Twitter4j (see section 4.2.1 for more details). We crawled the Twitter network for the period between 08/09/2014 and 03/11/2014. We collected tweets, users profiles, mentions, retweets and followership relations. Next, we filtered the obtained data and we kept only tweets that talk about smartphones, after, we deleted users that do not have any tweet in the dataset. Finally, we classified the resulting set of tweets into three classes:

- "*Android*"
- "*Galaxy*"
- "*Windows*"

In fact, if a given tweet contains the name of a class C, it is considered of that class. For example, a given tweet that contains the word "Android", is classified to the class "*Android*".

In this stage, we have a dataset that contains a set of Twitter users, followership links between them, who mentions whom, who retweets from whom and classified tweets into the three classes. In a second stage, we extract the propagation traces of each class of messages. For this purpose, we consider that a tweet of class C is propagated from a user u to a user v if and only if:

- the user u posts a tweet of class C before v.
- and at least one of the following relations between u and v exists:
  1. v follows u,
  2. u mentions v in a tweet of class C,
  3. v retweets a tweet of class C written by u.

Definitions of "follow", "mention", "retweet" and "tweet" can be found in section 2.3.1.1. At the end of this step, we get a table containing the propagation links of each class of messages. Next, we extract propagation networks (PrNets) such that each PrNet has to have one source. Then, we consider a node as a source if and only if it is the first to send the message, *i.e.* it does not receive the message from any other node. Table 5.1 presents some statistics about the data set.

In the next step, we define the types of links on the social network. On Twitter, there exist many possible relationships, the first one is explicit which is the follow relation, the second and the third relations are implicit which are the mention and the retweet. Another property of Twitter is that between two users u and v we can have a follow, a mention and/or a retweet relationship. We assign to each relationship a weight [12] and we assign

Table 5.1: Statistics of the data set

	Android	**Galaxy**	**Windows**
#User	6435	4343	5775
#Follow	9059	4482	12466
#Tweet	81840	8067	11163
#Retweet	3606	2873	2632
#Mention	6092	5965	3441
#Prop. links	7623	6819	11400
#PrNet	224	161	219

to each link a vector of weights having the form (w_f, w_m, w_r) where w_f is the weight of follow relationship, w_m is the weight of mention relationship and w_r is the weight of retweet relationship.

Let Sc_u be the set of successors of u, Pc_u the set of predecessors of u, Tc_u the set of tweets of u, $Rt_u(v)$ the set of tweets of u that are retweeted by v, $Mt_u(v)$ the set of tweets of u in which v is mentioned and Mt_u the set of tweets in which u mentions another user. We compute weights [12] as follows:

$$w_f(u,v) = \frac{|Sc_u \cap Pc_v| + 1}{|Sc_u|} \tag{5.13}$$

$$w_m(u,v) = \frac{|Mt_u(v)|}{|Mt_u|} \tag{5.14}$$

$$w_r(u,v) = \frac{|Rt_u(v)|}{|Tc_u|} \tag{5.15}$$

In the next sections, we use the presented datasets to show the performance of the proposed classifiers.

5.6.2 Results on generated propagation

In this experiment, we use the first dataset, and we compare the accuracy of the proposed algorithm in classifying the generated propagation networks of social messages. We use the Euclidean distance (equation (5.3)) for the probabilistic classifier and the Jousselme distance (equation (5.4)) for the evidential one. To obtain accurate results we run the experimental process ten times. At each running, we generate a new training set and testing set, we use the process described in section 5.6.1.1 to generate the datasets. The result of each running of the algorithms is an accuracy value for each algorithm. To obtain the classification accuracy, we compute the percentage of correctly classified messages. Next, for each classifier, we take the mean and the 95% confidence interval of the ten classification accuracies.

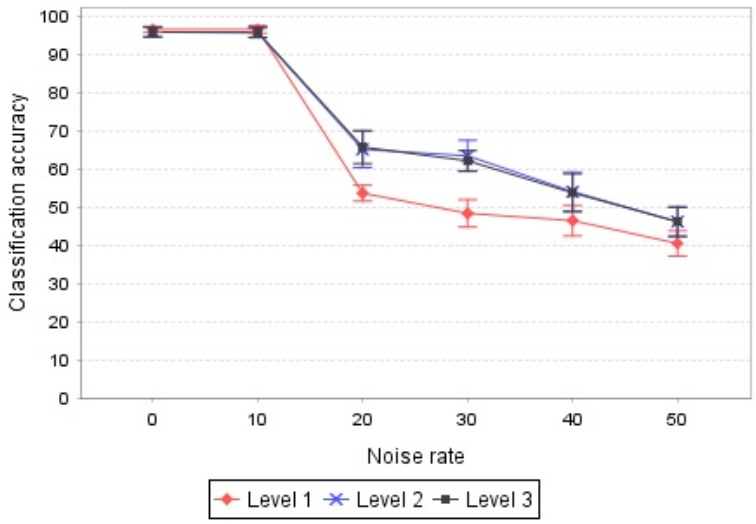
Figure 5.7 shows the impact of propagation levels on the classification accuracy of probabilistic results (Figure 5.7a) and the evidential results (Figure 5.7b). We noticed that the

accuracy increases when the propagation level increases, we observe this fact starting from the noise level 20%. In Figure 5.7a we observe that the curve of the second level coincides with the curve of third level and practically there is no improvement in the accuracy. However, in Figure 5.7b (evidential results), we note that the accuracy increases with the propagation level, this fact is observed starting from the noise rate 20%. Then, the accuracy of the third level is greater than the one of the first and the second levels, and the accuracy of the second level is higher than of the first one. Therefore, more the message propagates in the network, more we can characterize it, more accurate its predicted class.

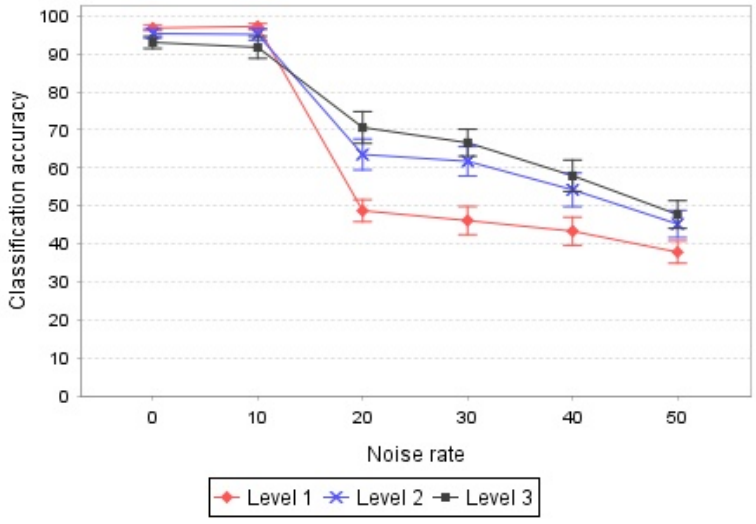
In Figure 5.8, we compare the accuracy of the probabilistic and evidential classifiers in the third propagation level. We notice that without noise (0%) we have good classification rates. In fact, the probabilistic accuracy is about 96% (with a 95% confidence interval of ± 1.27) and the evidential accuracy is equal to 93% (with a 95% confidence interval of ± 1.60). However, in a real world case, the absence of noise arises rarely. In the case where the noise rate increases, the curve shows that the classification accuracy decreases. Furthermore, we observe that the evidential (Belief) curve starts to be upper than the probabilistic (Proba) one. This fact appears from the noise rate 20% where we have an evidential accuracy equals to 70.7% (± 4.33) and a probabilistic one equals to 65.8% (± 4.18). Thus, we can conclude that the evidential classifier is more robust against the noise and gives better classification accuracy than the probabilistic classifier. In fact, the mass function used in the evidential algorithm considers some imprecision which mitigates the effect of the noise on the classification accuracy.

In a second experiment, we compare the probabilistic and the evidential classifiers with the PrNet-DTW k-NN and the PrNet-DTW belief k-NN in terms of the classification accuracy. We fix k to 5, the propagation level to 8, and we used the conjunctive combination rule for the PN-DTW belief k-NN. Figure 5.9 presents the obtained results. We notice that the curves of the four algorithms have the same shape when increasing the noise rate. Furthermore, we observe that the PN-DTW belief k-NN and PN-DTW k-NN give better classification accuracy until reaching the noise rate 40%. Finally, we noticed that when we consider more propagation levels in the probabilistic and the evidential classifiers become more robust against noise. In fact, there is an amelioration of the classification accuracy with the noise rate 20% and 30% which is not the case in Figure 5.8 where we considered only three levels of propagation.

In this section, we make some experiments on the first dataset. These experiments show the dependance of the probabilistic and the evidential classifiers to the propagation level. Besides, the proposed classifiers are very useful to classify social messages without any access to their content. According to these experiments, the belief classifier is better than the probabilistic one when we have a noise rate greater than 20%. Furthermore, according to Figure 5.9, we conclude that the best classifier that is robust to noise, is the PrNet-DTW belief k-NN classifier.



(a) Probabilistic results



(b) Evidential results

Figure 5.7: The impact of the propagation level on the classification accuracy

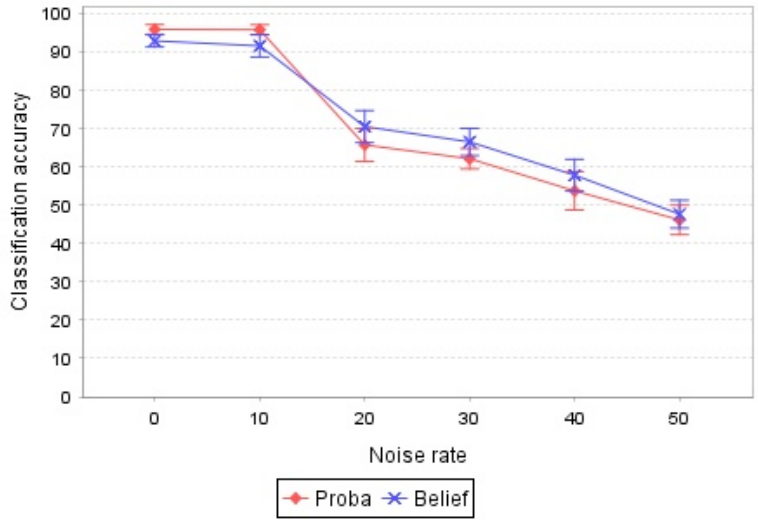


Figure 5.8: Comparison between probabilistic results and evidential results (level three)

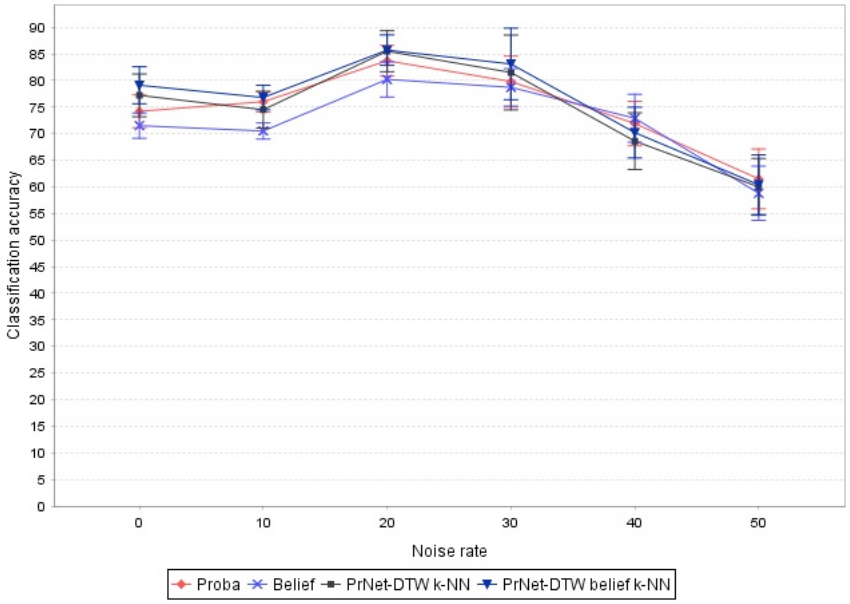


Figure 5.9: Comparison between the four proposed classifiers

Table 5.2: Comparison between PrNet classifiers

	Probabilistic classifier	Belief classifier	PrNet-DTW k-NN	PrNet-DTW Belief k-NN
Accuracy	51.97% ±2.04	52.25% ±1.99	**88.69%** ±3.39	**89.92%** ±3.20

5.6.3 Results on real world propagation

In this section, we test the performance of the proposed algorithms on real world propagation networks, and we compare the four classifiers between each other. The probabilistic and the evidential propagation network classifiers [51] works with a discrete type of links, then, a discretization step is needed. The main purpose of the discretization is to transform continuous types of links to discrete one. For this purpose, we use the following control structure: if the weight value (w_f, w_m or w_r) is greater than 0 we replace it by 1 in the discrete weight vector elsewhere we keep it null (equals 0).

Example 18. If the link is weighted by the vector ($w_f = 0.5$, $w_m = 0$, $w_r = 0.25$), the output after the discretization step will be (1, 0, 1). □

Next, we use the cross validation principle to divide the dataset into training and testing sets. Then, we split, randomly, our data set into two subsets: the first one contains 90% of PrNets and it is used for training and the second one (10%) is used for testing. We do this division ten times. Besides, we choose the Euclidean distance to evaluate the $\delta(a_i, b_j)$ in the computation process of the PrNet-DTW.

The k-NN algorithm is known to be dependent to k value, and varying k may vary the classification accuracy. Then, to see the impact of the parameter k, we made this first experiment. In fact, we run the two k-NN based algorithms with multiple k values and we obtained results in Figure 5.10. We note that odd values are more appropriate to k when we use the Probabilistic k-NN. Moreover, the PrNet-DTW belief k-NN has not the same behavior as the PrNet-DTW k-NN. In fact, the curve of the evidential classifier is more stable than the curve of the probabilistic one and the variation of the value of k does not have a great effect on the classification accuracy.

A second experiment is done to evaluate and compare the proposed classification algorithms. We fix the parameter k to 5 and we obtain results in table 5.2. As shown in table 5.2, the probabilistic and the belief classifiers do not give good classification accuracy, this behavior is a consequence of the discretization step that leads to the loss of the information given by weights. In contrast, the two PrNet-DTW based classifiers show their performance. Indeed, we have got good accuracy rates: 88.69% (±3.39, for a 95% confidence interval) and 89.92% (±3.20) respectively. We also see that the PrNet-DTW belief classifier gives slightly better results than the probabilistic one.

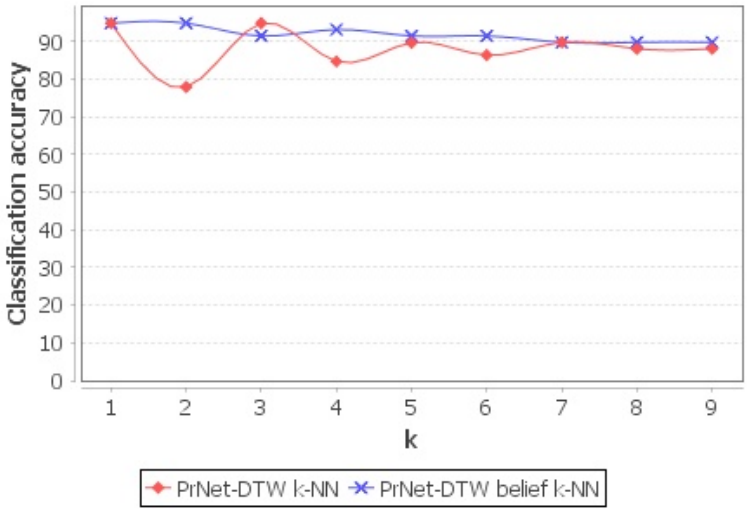


Figure 5.10: k variation

5.7 Conclusion

In conclusion, this chapter introduces a new classification approach for social messages that do not need any access to their content. The idea behind the proposed approach is that to classify the propagation traces of the social message instead of classifying its content. A second important contribution that is presented in this chapter, is an information propagation algorithm that considers the class of the message to be propagated in the propagation process.

According to the presented experiments, we can conclude that the proposed social message classifiers are useful to characterize a given social message without any access to its content. Then, we just need the propagation traces of the message in order to determine its class. The second important contribution of the proposed classifiers is that they are adaptable with any type of social content, *i.e.* text, image, video, etc. Another interesting conclusion is that the proposed classifiers can be sorted according to their performance against the noise as follows: the best classifier is PrNet-DTW Belief k-NN, then we have PrNet-DTW probabilistic k-NN, next the belief classifier and the last one is the probabilistic classifier.

The next chapter is dedicated to present some conclusions of this thesis. Besides, we introduce a new generalization idea for the proposed social message classifiers in order to be more adaptable for the Viral Marketing, and to be useful to identify the influencers according to their topics of interest.

6

Conclusion and perspectives

In this thesis, we focus on proposing new solutions to improve the effectiveness of a Viral Marketing campaign. In fact, Viral Marketing exploits the world of mouth effect and uses social networks to promote a product, a brand, etc. Scientifically, this problem is translated to the problem of influence maximization in social networks. We start our work by identifying the drawbacks of existing solutions. In fact, existing influence maximization approaches lack of considering many important influence markers like the user's activity in the network and his opinion about the object of the viral marketing campaign, *e.g.* a product or a brand. Then we focus on resolving some of these disadvantages. Another important problem we found while studying existing approaches, is about social messages classification that is an interesting step if we want to consider the topic to which a given social network user is interested to.

In a first step of this thesis, we focus on the problem of influence maximization. In fact, we study the influence aspects in Twitter. Next, we propose a new evidential influence measure for Twitter users. We use the theory of belief functions in the estimation process in order to take profit from the robustness of this theory in managing imperfect data and especially its performance in managing conflict while combining many pieces of information. The proposed influence measure contracts many influence aspects like the strength of the user's relationships, its activity in the networks, the propagation of its tweets, etc. Also, it is possible to adapt it to other social networks, we just need to define a set of link weights for the influence aspects we want to consider.

After defining an evidential influence measure, we search to consider the user's opinion about the product. For this purpose, we define three Viral Marketing scenarios that may arise. In the first scenario, we are interested to influencer users having a positive opinion about the product or the object of the Viral Marketing campaign. In the second scenario, our focus is on influencers having a positive opinion about the product and exert more influence on users that have a positive opinion too. In the last scenario, we look for positive opinion influencers that exert more influence on users having a negative opinion. For each

defined scenario, we propose two influence measures that take into account specificities of each scenario.

Next, we introduce two influence maximization models that can be used with any proposed influence measure. The first model considers the influence that exerts a given node on its direct out-neighbors. This influence model target influencers having many direct neighbors. Besides, it is very useful for products having some quality issues as it is more prudent in selecting influencers. The second maximization model takes into account the influence in two hopes, that is to say, it considers the influence that exerts a given user on his out-neighbors and on out-neighbors of his out-neighbors. This second model is very adaptable if we want to reach and influence a maximum number of users in the network.

To prove the performance of the proposed influence maximization solutions we present several experiments. First we introduce a Viral Marketing case study, in which the main goal is to promote the propagation of smartphones on Twitter. We run these experiments on a real world dataset collected from Twitter. Then, a second set of experiments is done on generated data in order to study the behavior and the accuracy of each proposed influence measure and to compare the performance of the proposed Viral Marketing scenarios.

In the proposed case study, we compare the selected smartphones' seeds using the proposed two models to those selected by existing models. Our results prove the performance of our solutions against existing ones especially when we compare the evidential models to the credit distribution [41] model that we consider to be the closest in its principle to our models. Next, we compare the proposed influence measures using the second influence maximization model to opinion-based cascading model [99] which is an existing solution that considers the user's opinion. According to these experiments, we notice that the fact of considering the user's opinion has ameliorated the quality of selected seeds. Also, we ameliorate the mean positive opinion against to the opinion-based cascading model.

The second set of experiments is made on generated data. The purpose of these experiments is to study the performance of the proposed influence measures. Indeed, we generate the data in such a way that we know the influencers, the positive influencers and the positive influencers influencing positive and negative users respectively. Next, we study the accuracy of our solutions in detecting each type of influencers. According to our experiments, the proposed influence measures succeed in getting good accuracy values especially when detecting influencers where we have an accuracy of about 100% obtained by all measures.

In a next step of this thesis, we focus on the problem of social message classification. We define a new classification approach that ignores the message content and considers its propagation traces. In fact, the idea behind the proposed solutions is that the propagation of a given message is directed by its content, then, we search to classify the propagation network of a given message in order to have a more clearer idea about its content. For this purpose, we propose two main classification ways, the first one is based on a model that summarizes the training set. The second one is distance based, then, we need the training

set to classify each new coming message. Next, we conduct some experiments to prove the performance of the proposed solutions and we succeed in getting good classification rates.

In this thesis, we achieve many new findings and good results that ameliorate those given by existing solutions. However, many other ameliorations still not yet resolved. In the following, we introduce some perspectives for future works:

- A first perspective that comes in mind is to generalize the proposed social message classification approach to be useful with the proposed influence maximization models. In fact, in a Viral Marketing campaign we need to know if a detected influencer is interested in the product or not. Then, in such a case we need to know the topics to which the influencer is interested to. Furthermore, it is important to consider this information in the influence maximization process. Indeed, in our work we assume to have only one topic while maximizing the influence. However, this is not always the case in the real world social networks. This idea will be done in two main steps:
  - In a first step, we will search to propose a more generic classification approach for social messages. The idea here is to consider the content of the message and its propagation traces in the classification process to determine the set of topics to which a given user is interested. Besides, we want to know if the user has some interest or preference about a given topic. Such an information is very helpful if we want to find influencers that are more interested in a given product or brand.
  - In a second step, the main purpose is to integrate the existence of many topics and topic preferences in the influence maximization model. In fact, we need a more generic model that considers many topics of interest in the social network and the fact of preferring some topics. This information will be very useful if we search to detect good seeds to promote the Viral Marketing campaign.
- Another interesting idea for future works is about maximizing the influence within communities. A community is defined as a set of users or vertices that are connected more densely to each other than to other users from other communities [101]. People in the same community generally have some common properties. For example, they may be friends that attended the same school or they are from the same town. The idea here is to minimize the number of selected influencers and the time spent to find them. In fact, we will search to find influencers at the scale of the community instead of the social network and it is obvious that the community is smaller than the social network.
- A third interesting perspective is to generalize the proposed influence maximization models by adapting them to other social networks like Facebook and LinkedIn. In fact, each social network has some specific characteristics that distinguish it from the others. For example, Facebook allows its users to express their feeling or reaction

about a given post. Then, someone may love a post or finds it funny[1] for example. All these specificities may be very informative for influence maximization.

- A last perspective will be about defining an updating technique for influence maximization. Indeed, every day online social networks collect a huge amount of data that may contain much new information about a given product. Then, a detected influencer spreader today, maybe not an influencer after a period of time or it may appear another more influencer user. On the other hand the influence maximization process may be expensive and time consuming in some cases. Hence, an updating approach for social influencers may be very interesting. Its main purpose is to update the set of seeds by adding some new ones and/or deleting those that are no more influencers. The updating will be done without running again all the influence maximization process. In fact, we want to use the seed set we have, and the new coming data in order to generate a new updated seed set.

[1]Constine J., Facebook Enhances Everyone's Like With Love, Haha, Wow, Sad, Angry Buttons, https://techcrunch.com/2016/02/24/facebook-reactions/, Posted on 24/02/2016, Seen on 30/08/2016.

A

Theory of belief functions

Contents

Summary

The theory of belief functions is very useful for the processing of imperfect information. Indeed, it provides a rich framework for the information modeling and processing, also, it allows the decision making. In this thesis, we are mainly using this theory to combine many pieces of information that come from different sources and to manage the conflict that may arise between them. In this appendix, we present this theory and we detail some of its basic concepts that are used in this document.

A.1 Introduction

The theory of belief functions was first introduced by Dempster in his paper [26]. Next, Shafer published his book "*A mathematical theory of evidence*" [79] in which he developed the basic concepts of this theory. Next, many works appeared either to develop new tools to enrich the theory or to use it in an application domain as in our case. This theory has other names like Dempster-Shafer theory and evidence theory. In this thesis, we use the theory of belief functions especially for information fusion and conflict management.

In this appendix, we give a brief overview of the theory of belief functions, we detail its basic concepts and we give some examples to explain them. First, we present some concepts that are used to model the information and to present the different pieces of information in a same universe (Section A.2). Next, we move on to the information fusion (Section A.3). In fact, having many pieces of information is not always understandable, then, we need to fuse them into one compact piece of information that summarizes those given as input. Finally, a decision making step is, always, needed (Section A.4). Hence, we introduce some well known tools that are generally used for making decision in the belief functions framework.

A.2 Information modeling

In this section, we present some functions that are very useful to model the information in order to make its processing easier. First, we present the basic belief assignment (also called mass function), then, we introduce some of its transformations.

A.2.1 Mass function

The first think to define while using the theory of belief functions is the *frame of discernment* or in other words the set of all possible decisions or choices in the given problem. Suppose that $\Omega = \{C_1, C_2, ..., C_n\}$ is our frame of discernment where $C_i \cap C_j = \emptyset$, $C_i, C_j \in \Omega$ and $C_i \neq C_j$. In fact, the conjunction between Ω elements is not allowed. Next, we define the *power set*, 2^Ω, which is the set of all subsets of Ω:

$$2^\Omega = \{\emptyset, \{C_1\}, \{C_2\}, \{C_1, C_2\}, ..., \{C_1, C, ..., C_n\}\} \tag{A.1}$$

Example 19. Let's consider the well known example of the murder of Mr. Jones introduced by [84]. Big Boss has a team of assassins composed of three members which are Peter, Paul and Mary. We need to define a frame of discernment that contains all possible assassins: Ω is formed by: Peter (Pe), Paul (Pa) and Mary (Ma), $\Omega = \{Pe,\ Pa,\ Ma\}$, and its corresponding power set is:

$$2^\Omega = \{\emptyset, \{Pe\}, \{Pa\}, \{Pe, Pa\}, \{Ma\}, \{Pe, Ma\}, \{Pa, Ma\}, \{Pe, Pa, Ma\}\} \tag{A.2}$$

□

The *mass function*, also called *basic belief assignment (BBA) function*, m^Ω, is defined by the following mapping:

$$\begin{aligned} 2^\Omega &\rightarrow [0,1] \\ A &\mapsto m^\Omega(A) \end{aligned} \tag{A.3}$$

The amount $m^\Omega(A)$ is the mass value assigned to the subset $A \subseteq \Omega$. The mass function has the following property

$$\sum_{A \subseteq \Omega} m^\Omega(A) = 1 \tag{A.4}$$

In the case where we have $m^\Omega(A) > 0$, A is called *focal element*. If $m^\Omega(\emptyset) = 0$, m^Ω is said to be a normalized mass function. In many cases, we may have $m^\Omega(\emptyset) \geq 0$. The mass given to the empty set, $\emptyset$, is the mass value that is not given to any subset. It is called *fusion inconsistency value*. It appears generally when we combine many pieces of information and it is caused, generally, by the non idempotence of the combination rule and the conflict between information sources, *i.e.* the degree of contradiction between them. It can be redistributed using the following transformation:

$$m^\Omega(A) = \frac{m^\Omega(A)}{1 - m^\Omega(\emptyset)} \tag{A.5}$$

$$m^\Omega(\emptyset) = 0 \tag{A.6}$$

The mass value given to the set Ω is the mass that can not be given to its subsets and it is called *total ignorance*. When we compare a BBA distribution to a probability distribution, we notice that the BBA allows a subset of Ω to be a focal element when we have some doubt about the decision, while the probability theory forces the equiprobability in such a case.

Example 20. Let us take the same example of the murder of Mr. Jones. One day, Big Boss decided that Mr. Jones has to be killed. Then, he selected a killer from his team using a dice, if he obtain an even number, then the killer is a female, else, the killer is a male. Let's help the Judge to find the murder. We know that Mr. Jones has been murdered and the sex of the murder was selected through a dice. However, there is no information about the choice between Peter and Paul in the case of an odd number.

Knowing this information, we can define the following BBA on Ω:
$m_1^\Omega(\{Pe, Pa\}) = 0.5$ and $m_1^\Omega(\{Ma\}) = 0.5$. □

There are some particular mass functions like *the simple mass function*, also called *simple BBA* [79, 82]. A BBA is said to be simple if it has two focal elements, the first one

is a subset of Ω, $A \subseteq \Omega$, and the second one is Ω. Let $\alpha \in [0,1]$, the simple BBA m^Ω is defined as:

$$m^\Omega(A) = \begin{cases} 1-\alpha & A \subseteq \Omega \\ \alpha & A = \Omega \\ 0 & otherwise \end{cases} \tag{A.7}$$

Example 21. Suppose that a new information comes to the Judge that said: the killer may be Paul or Marry and our belief on this assumption is 0.6 and their is no new information about Peter.

We model this information using a simple mass function defined on Ω: $m_2^\Omega(\{Pa, M\}) = 0.6$ and $m_2^\Omega(\{\Omega\}) = 0.4$. □

Another very useful particular mass function is the *consonant mass*. This BBA is characterized by its nested focal elements, *i.e.* $A_1 \subseteq A_2 \subseteq \ldots \subseteq \Omega$.

Example 22. The following BBA is consonant: $m_3^\Omega(\{Pe\}) = 0.2$, $m_3^\Omega(\{Pe, Pa\}) = 0.4$ and $m_3^\Omega(\Omega) = 0.4$. □

In the next section, we present some possible transformations of the mass function.

A.2.2 Mass transformations

In this section, we introduce some transformations of the mass function that are very useful to present differently the same piece of the information. The *belief function*, bel^Ω, represents the minimal amount of support that is given to the subset A. In other words, it represents the total belief given to $A \subseteq \Omega$ [79]. As the mass function, bel^Ω is also a mapping from 2^Ω to $[0,1]$. The belief given to A is obtained through the following equation:

$$bel^\Omega(A) = \begin{cases} 0 & if\ A = \emptyset \\ \sum_{\emptyset \neq B \subseteq A} m^\Omega(B) & \forall A \subseteq \emptyset,\ A \neq \emptyset \end{cases} \tag{A.8}$$

The mass function, m^Ω, that produces bel^Ω can be retrieved using the following equation:

$$m^\Omega(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|} bel^\Omega(B), \forall A \subseteq \Omega \tag{A.9}$$

Another important function is the *plausibility function*, pl^Ω, which represents the maximum amount of support that can be given to a subset A if another information become available. It is also a mapping from 2^Ω to $[0,1]$ and it is estimated using the following equation:

$$pl^{\Omega}(A) = bel^{\Omega}(\Omega) - bel^{\Omega}\left(\bar{A}\right), \forall A \subseteq \Omega \quad (A.10)$$

$$pl^{\Omega}(A) = \sum_{B \cap A = \emptyset} m^{\Omega}(B), \forall A \subseteq \Omega \quad (A.11)$$

The mass function that produces pl^{Ω} can be retrieved as in the case of the belief function, bel^{Ω}, using the following equation:

$$m^{\Omega}(A) = \sum_{B \subseteq A} (-1)^{|A|-|B|-1} pl^{\Omega}\left(\bar{B}\right), \forall A \subseteq \Omega \quad (A.12)$$

Example 23. Consider the mass function of the first example. Table A.1 presents a computation example of bel^{Ω} and pl^{Ω}. □

Table A.1: Mass, belief and plausibility example

	m^{Ω}	bel^{Ω}	pl^{Ω}
$\emptyset$	0	0	0
$\{Pe\}$	0	0	0.5
$\{Pa\}$	0	0	0.5
$\{Pe, Pa\}$	0.5	0.5	0.5
$\{Ma\}$	0.5	0.5	0.5
$\{Ma, Pe\}$	0	0.5	1
$\{Ma, Pa\}$	0	0.5	1
Ω	0	1	1

A.2.3 From a probability to a BBA

The transition from a probability distribution to a BBA distribution is, eventually, possible. This transformation is called *consonant transformation* or *inverse pignistic transformation* [3, 4]. Let Pr^{Ω} be a probability distribution defined on Ω. To transform Pr^{Ω} into m^{Ω}, we first order the probabilities given to singletons of Ω as:

$$Pr^{\Omega}(C_1) \geq Pr^{\Omega}(C_2) \geq \ldots \geq Pr^{\Omega}(C_n) \quad (A.13)$$

Next, we use the following equations to obtain a BBA:

$$m^{\Omega}(\{C_1, C_2, \ldots, C_n\}) = n.Pr^{\Omega}(C_n) \quad (A.14)$$

$$m^{\Omega}(\{C_1, C_2, \ldots, C_{n-1}\}) = (n-1).\left(Pr^{\Omega}(C_{n-1}) - Pr^{\Omega}(C_n)\right) \quad (A.15)$$

$$\ldots \quad (A.16)$$

$$m^{\Omega}(\{C_1, C_2\}) = (2).\left(Pr^{\Omega}(C_2) - Pr^{\Omega}(C_3)\right) \quad (A.17)$$

$$m^{\Omega}(\{C_1\}) = (1).\left(Pr^{\Omega}(C_1) - Pr^{\Omega}(C_2)\right) \quad (A.18)$$

In the next section, we detail some tools that are very useful to combine the information.

A.3 Information fusion

After the information modeling step, comes the information fusion which is an essential step when working with belief functions. It allows the combination of many pieces of information that comes from different and independent sources. The strength of the theory of belief functions comes from its strength in the management of the conflict during the information fusion step. Besides, it performs many tools for this purpose. Indeed, each of these has its specific characteristics and according to these characteristics we can select an appropriate tool for a given case. In the literature, we find many combination rules like Dempster's rule [26], Conjunctive combination rule [80], Dubois et Prade rule [31], Yager rule [97], PRC6 [68, 69], etc.

The Dempster's rule [26] is the first combination rule that was introduced to combine pieces of evidence in the theory of belief functions. Having two mass functions, m_1^Ω and m_2^Ω, that comes from two distinct sources, we can obtain the combined BBA distribution, $m_{1\oplus 2}^\Omega$, using the following equation:

$$m_{1\oplus 2}^\Omega(A) = \begin{cases} \frac{\sum_{B\cap C=A} m_1^\Omega(B).m_2^\Omega(C)}{1-\sum_{B\cap C=\emptyset} m_1^\Omega(B).m_2^\Omega(C)}, & \forall A \subseteq \Omega,\ A \neq \emptyset \\ 0 & if\ A = \emptyset \end{cases} \tag{A.19}$$

The resulting BBA , $m_{1\oplus 2}^\Omega$, is normalized, *i.e.* $m_{1\oplus 2}^\Omega(\emptyset) = 0$.

Another very useful combination rule is the *Conjunctive rule of combination CRC* [80]. It is used to fuse two mass functions, m_1^Ω and m_2^Ω, that have two distinct sources as follows:

$$m_{1\circledast 2}^\Omega(A) = \sum_{B\cap C=A} m_1^\Omega(B).m_2^\Omega(C),\ \forall A \subseteq \Omega \tag{A.20}$$

The BBA $m_{1\circledast 2}^\Omega$, result of the CRC, is not normalized, *i.e.*

$$m_{1\circledast 2}^\Omega(\emptyset) = \sum_{B\cap C=\emptyset} m_1^\Omega(B).m_2^\Omega(C) \geq 0$$

Example 24. Table (A.2) is an example of the Dempster's rule and CRC. □

A.4 Decision making

As mentioned above, to use the theory of belief functions, generally, we start by transforming each given piece of information to a BBA distribution. Next, we combine the obtained BBAs

Table A.2: Combination rules example

	m_1^Ω	m_2^Ω	Dempster's rule	CRC
$\emptyset$	0	0	0	0.15
$\{Pe\}$	0	0	0	0
$\{Pa\}$	0	0	0.1765	0.15
$\{Pe, Pa\}$	0.5	0.3	0.4118	0.35
$\{Ma\}$	0.5	0	0.4118	0.35
$\{Ma, Pe\}$	0	0	0	0
$\{Ma, Pa\}$	0	0.3	0	0
Ω	0	0.4	0	0

in order to obtain a BBA, m^Ω, that summarizes all pieces of information. Then, we move on to the decision making step. This third step can be done using the pignistic probability, noted $BetP^\Omega$ [83]. It is a mapping from Ω to $[0, 1]$ and it is calculated from the combined BBA, m^Ω, as follows:

$$BetP^\Omega(C_i) = \sum_{d_i \in A,\, A \in \Omega} \frac{m^\Omega(A)}{card(A).(1 - m^\Omega(\emptyset))} \tag{A.21}$$

where $card(A)$ is a function that computes the cardinality of the subset A. The chosen decision according to the pignistic probability is the decision, C_i, having the maximum pignistic probability value.

Example 25. Let return to the example of the murder of Mr. Jones. We have $m_1^\Omega(\{Pe, Pa\}) = 0.5$, $m_1^\Omega(\{Ma\}) = 0.5$, $m_2^\Omega(\{Pa, Ma\}) = 0.6$ and $m_2^\Omega(\{\Omega\}) = 0.4$. We use the Dempster's rule to combine these two pieces of information, we have got: $m^\Omega(\{Ma\}) = 0.5$, $m^\Omega(\{Pa\}) = 0.3$ and $m^\Omega(\{Pe, Pa\}) = 0.2$. Next, we compute $BetP^\Omega$ and we obtain: $BetP^\Omega(Ma) = 0.5$, $BetP^\Omega(Pe) = 0.1$ and $BetP^\Omega(Pa) = 0.4$. As a result, the murder of Mr. Jones seems to be Mary as it has the maximum pignistic probability. □

A.5 Conclusion

In conclusion, this appendix is dedicated to the theory of belief functions. This theory is very helpful for the process of the imperfect information that may be imprecise, uncertain, etc. In this thesis, the theory of belief functions is used to define the influence measures and to combine the influence aspects. Besides, it is also used to define an evidential classifier for social messages.

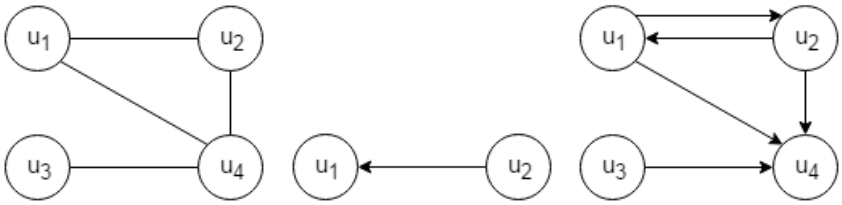
B

Graph theory: Basic concepts

Contents

Summary

The graph theory and its tools are very helpful if we are working with social networks. In fact, a social network is a graph where nodes are people, groups of people, etc, and links are the relationships between them. In this appendix, we explain some essential concepts of the graph theory that are very useful for social networks.



(a) Graph (b) Directed edge (c) Digraph

Figure B.1: Example of graphs

B.1 Introduction

The social network analysis field is based on the graph theory. Indeed, the use of graphs to model the social relationships is very helpful as it allows the analyzer to study the structural properties of the network. In the literature, we find many works that use the structure of the network to select influencers, to simulate the process with which the information propagates through the network [55, 56], etc. We also use the graph structure of the social network in our work, which justify the interest of this appendix.

The remainder of this appendix is organized as follows: Section B.2 defines some useful basic concepts from the graph theory, and Section B.3 introduces some centrality measures.

B.2 Basic concepts definitions

In this section, we define some concepts from the graph theory that are used in this document.

Definition 15. A *Graph* G is a couple (V, E) where V is a finite set of elements called *vertices* or *nodes*, $v \in V$, and E is a finite set of pairs of elements from V called *edges*, $(u, v) \in E$, $u, v \in V$. An edge represents a relationship between two nodes. If the edges have a specific direction, they are said to be *directed edges*. Besides, a graph having directed edges is called *directed graph* or *digraph*.

Example 26. Figure B.1a is an example of a graph where u_1, u_2, u_3, and u_4 are vertices and links between them are edges. Figure B.1b is an example of an edge, it starts from u_2, the source, to its destination u_1. Figure B.1c is an example of a digraph. □

Definition 16. The *degree* of a vertex, v, in a G is the number of neighbors of v:

$$degree\,(v, G) = |N\,(v)\,| \tag{B.1}$$

where $N\,(v)$ is the set of neighbors of the vertex v.

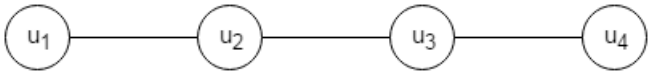


Figure B.2: A path relating A to D

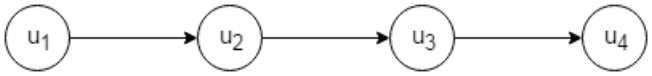


Figure B.3: A dipath from A to D

Example 27. The degree of the node u_1 in the graph of the Figure B.1a is two as it has two neighbors u_2 and u_4. □

Definition 17. The *indegree* of a vertex, v, in a directed graph is the number of links having v as source. □

Definition 18. The *outdegree* of a vertex, v, in a directed graph is the number of links having v as destination.

Example 28. Let take the digraph of Figure B.1c, the indegree of u_1 is equal to one and its outdegree is two. □

Definition 19. The *strength* of a node, v, in a weighted graph is the sum of weights of links that relates v to its neighbors.

Definition 20. A *path* in a graph G between two nodes u and v is a sequence of distinct edges connecting a sequence of vertices. A *directed path* or a *dipath* is a sequence of vertices connected with directed edges that are directed to the same direction. A shortest path between two nodes in the graph is called *geodesic*.

Example 29. Figure B.2 is an example of path that relates u_1 to u_4 and Figure B.3 is an example of a directed path that starts from u_1 to u_4. □

Definition 21. A *cycle* is a path that starts and ends in the same vertex. A *directed cycle* is a cycle with a directed path.

Definition 22. A *directed acyclic graph (DAG)* is a graph that does not contain cycles.

Example 30. Figure B.4 is an example of a directed acyclic graph. □

The reader can refer to [15] and [96] for more details and examples about the graph theory.

B.3 Centrality measures

Centrality measures search to find the centric nodes in the graph. In the literature, we find many definitions for the centrality of a vertex in a given graph. It may be, for example, the

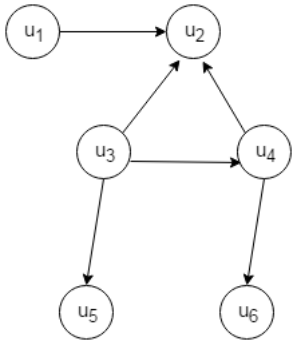


Figure B.4: Directed acyclic graph

degree of the node, then the node having the maximum degree is the centric node in the graph [73]. In this section, we present some examples of these measures and for more details the reader can refer to the book of *Newman* [73].

Definition 23. The *betweeness centrality* measures "the extent to which the focal vertex lies on a large number of shortest paths between various third parties" [20]. Given a graph G, the betweeness can be estimated as follows [34]:

$$c_b(v) = \sum_{\{u,w\} \subset V/\{v\}} \frac{g'(u,v,w)}{g(u,w)} \tag{B.2}$$

where $g(u,w)$ is the number of geodesics in G and $g'(u,v,w)$ is the number of (u,w) geodesics in G containing v. A geodesic between two nodes is a path with a minimum number of links between them. We have $c_b(v) = 0$ when $g(u,w) = 0$. The work of [8] introduces an algorithm to estimate the betweeness of a node in a graph.

Definition 24. The *closeness centrality* measures the mean distance between a given vertex v to other vertices in the network. It is defined by the following equation [77]:

$$c_c(v) = \frac{1}{\sum_{u \in V} g(v,u)} \tag{B.3}$$

The degree, the betweeness and the closeness centrality measures are very useful to determine the structural importance of a given node in a given graph and more specifically in a social network.

B.4 Conclusion

To sum up, in this appendix we detail some basic concepts from the graph theory that are very useful in the social networks analysis field. Also, we introduce some centrality measures

which are the degree, the betweeness and the closeness centrality. In this thesis, we used graphs to model and manage the social network.

C

Publications

The proposed contributions has been the subject of the following publications:

1. Siwar Jendoubi, Arnaud Martin, Ludovic Liétard and Bouteheina Ben Yaghlane. *Classification of message spreading in a heterogeneous social network.* In the proceedings of the 15th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Montpellier, France, pp 66-75, (07/2014).

2. Siwar Jendoubi, Arnaud Martin, Ludovic Liétard, Bouteheina Ben Yaghlane and Hend Ben Hadji. *Dynamic Time Warping Distance for Message Propagation Classification in Twitter.* In the proceedings of the 13th european conference on symbolic and quantitative approaches to reasoning with uncertainty, Compiègne, France, pp 419-428, (07/2015).

3. Siwar Jendoubi, Arnaud Martin, Ludovic Liétard, Hend Ben Hadji and Bouteheina Ben Yaghlane. *Maximizing positive opinion influence using an evidential approach.* In the proceedings of the 12th International FLINS Conference: Uncertainty Modelling in Knowledge Engineering and Decision Making, Roubaix, France, pp 168-174, (08/2016).

4. Siwar Jendoubi, Arnaud Martin, Ludovic Liétard, Hend Ben Hadji and Bouteheina Ben Yaghlane. *Two evidential data based models for influence maximization in Twitter.* Knowledge-Based Systems Journal, http://dx.doi.org/10.1016/j.knosys.2017.01.014 (2017).

Bibliography

[1] Ahmed, S., Ezeife, C.I.: Discovering influential nodes from trust network. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing. pp. 121–128 (March 2013)

[2] Anderson, R.M., May, R.M.: Infectious Diseases of Humans. Oxford University Press (1991)

[3] Aregui, A., Denoeux, T.: Consonant belief function induced by a confidence set of pignistic probabilities. In: ECSQARU. pp. 344–355 (October 2007)

[4] Aregui, A., Denoeux, T.: Constructing consonant belief functions from sample data using confidence sets of pignistic probabilities. Int. J. Approx. Reasoning 49(3), 575–594 (2008)

[5] Aslay, C., Barbieri, N., Bonchi, F., Baeza-Yates, R.: Online topic-aware influence maximization queries. In: Proceedings of the 17th International Conference on Extending Database Technology (EDBT). pp. 24–28 (March 2014)

[6] Azaza, L., Kirgizov, S., Savonnet, M., Leclercq, E., Faiz, R.: Influence assessment in twitter multi-relational network. In: Proceedings of the 11th International Conference on Signal-Image Technology and Internet-Based Systems, SITIS 2015. pp. 436–443 (2015)

[7] Baccianella, S., Esuli, A., Sebatiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of the Seventh conference on International Language Resources and Evaluation. pp. 2200–2204 (May 2010)

[8] Bandes, U.: A faster algorithm for betweenness centrality. Journal of Mathematical Sociology 25(2), 163–177 (2001)

[9] Banerjee, S., Ramanathan, K., Gupta, A.: Clustering short texts using wikipedia. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 787–788. ACM (2007)

[10] Barbieri, N., Bonchi, F., Manco, G.: Topic-aware social influence propagation models. In: Proceedings of the 2012 IEEE 12th International Conference on Data Mining. pp. 81–90 (2012)

[11] Baumeister, R., Bratslavsky, E., Finkenauer, C., Vohs, K.: Bad is stronger than good. Review of General Psychology 5(4), 323–270 (2001)

[12] Ben Jabeur, L.: Leveraging social relevance: Using social networks to enhance literature access and microblog search. Ph.D. thesis, Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier) (October 2013)

[13] Ben Jabeur, L., Tamine, L., Boughanem, M.: Active microbloggers: Identifying influencers, leaders and discussers in microblogging networks. In: Proceedings of the 19th International Symposium String Processing and Information Retrieval. pp. 111–117 (October 2012)

[14] Bhatia, N., Vandana: Survey of nearest neighbor techniques. (IJCSIS) International Journal of Computer Science and Information Security 8(2), 302–305 (2010)

[15] Bondy, J.A., Murty, U.S.R.: Graph theory with applications, vol. 290. North-Holland: Elsevier (1976)

[16] Bozorgi, A., Haghighi, H., Zahedi, M.S., Rezvani, M.: Incim: A community-based algorithm for influence maximization problem under the linear threshold model. Information Processing and Management 000, 1–12 (2016)

[17] Brown, P., Feng, J.: Measuring user influence on twitter using modified k-shell decomposition. In: Proceedings of ICWSM'11 Workshops. pp. 18–23 (2011)

[18] Bunke, H.: On a relation between graph edit distance and maximum common subgraph. Pattern Recognition Letters 18(8), 689–694 (August 1997)

[19] Bunke, H., Foggia, P., Guidobaldi, C., Sansone, C., Vento, M.: A comparison of algorithms for maximum common subgraph on randomly connected graphs. In: International Workshop on Structural, Syntactic, and Statistical Pattern Recognition. pp. 123–132. Springer (August 2002)

[20] Butts, C.T.: Social network analysis: A methodological introduction. Asian Journal of Social Psychology 11, 13–41 (2008)

[21] Cha, M., Haddadi, H., Benevenuto, F., Gummadi, K.P.: Measuring user influence in twitter: The million follower fallacy. In: Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM). pp. 10–17 (May 2010)

[22] Chen, D., Lü, L., Shang, M.S., Zhang, Y.C., Zhou, T.: Identifying influential nodes in complex networks. Physica A: Statistical mechanics and its applications 391(4), 1777–1787 (2012)

[23] Chen, W., Collins, A., Cummings, R., Ke, T., Liu, Z., Rincon, D., Sun, X., Wang, Y., Wei, W., Yuan, Y.: Influence maximization in social networks when negative opinions may emerge and propagate. In: Procedings of SIAM SDM. pp. 379–390 (April 2011)

[24] Cheung, C.M., Lee, M.K.: Online consumer reviews: Does negative electronic word-of-mouth hurt more? In: Proceeding of the fourteenth americas conference on information systems. p. 143 (August 2008)

[25] Choudhury, M.D., Lin, Y.R., Sundaram, H., Candan, K.S., Xie, L., Kelliher, A.: How does the data sampling strategy impact the discovery of information discusion in social media? In: ICWSM'10. pp. 34–41 (2010)

[26] Dempster, A.P.: Upper and Lower probabilities induced by a multivalued mapping. Annals of Mathematical Statistics 38, 325–339 (1967)

[27] Denœux, T.: A k-Nearest Neighbor Classification Rule Based on Dempster-Shafer Theory. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans 25(5), 804–813 (mai 1995)

[28] Denœux, T., Sriboonchitta, S., Kanjanatarakul, O.: Evidential clustering of large dissimilarity data. Knowledge-Based Systems 106, 179–195 (2016)

[29] Derczynski, L., Ritter, A., Clark, S., Bontcheva, K.: Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing. pp. 198–206 (2013)

[30] Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of KDD'01. pp. 57–66 (2001)

[31] Dubois, D., Prade, H.: Representation and combination of uncertainty with belief functions and possibility measures. Computational Intelligence 4, 244–264 (1988)

[32] Dubois, E., Gaffney, D.: The multiple facets of influence: Identifying political influentials and opinion leaders on twitter. American Behavioral Scientist 58(10), 1260–1277 (2014)

[33] Fernández, M.L., Valienteb, G.: A graph distance metric combining maximum common subgraph and minimum common supergraph. Pattern Recognition Letters 22(6-7), 753–759 (May 2001)

[34] Freeman, L.C.: A set of measures of centrality based upon betweenness. Sociometry 40(1), 35–41 (1977)

[35] Gao, C., Wei, D., Hu, Y., Mahadevan, S., Deng, Y.: A modified evidential methodology of identifying influential nodes in weighted networks. Physica A 392(21), 5490–5500 (November 2013)

[36] Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. International Journal of Future Computer and Communication 13(1), 113–129 (Février 2010)

[37] Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters 12(3), 211–223 (August 2001)

[38] Gomez-Rodriguez, M., Balduzzi, D., Schölkopf, B.: Uncovering the temporal dynamics of diffusion networks. In: ICML'11. pp. 561–568 (2011)

[39] Gomez Rodriguez, M., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. In: KDD'10. pp. 1019–1028 (2010)

[40] Goyal, A., Bonchi, F., Lakshmanan, L.V.S.: Learning influence probabilities in social networks. In: WSDM'10. pp. 241–250 (Février 2010)

[41] Goyal, A., Bonchi, F., Lakshmanan, L.V.S.: A data-based approach to social influence maximization. In: Proceedings of VLDB Endowment. pp. 73–84 (August 2012)

[42] Goyal, A., Lu, W., Lakshmanan, L.V.S.: Celf++: Optimizing the greedy algorithm for influence maximization in social networks. In: Proceedings of the 20th international conference companion on World wide web. pp. 47–48 (2011)

[43] Granovetter, M.: Threshold models of collective behavior. American journal of sociology pp. 1420–1443 (1978)

[44] Guille, A., Hacid, H., Favre, C., Zighed, D.A.: Information diffusion in online social networks: a survey. SIGMOD Rec. 42(1), 17–28 (March 2013)

[45] Hansen, D.L., Shneiderman, B., Smith, M.A.: Analysing social media network with nodeXL insights from a connected world. Elsevier Inc. (2011)

[46] He, W., Zhab, S., Li, L.: Social media competitive analysis and text mining: A case study in the pizza industry. International Journal of Information Management 33, 464–472 (2013)

[47] Hu, X., Sun, N., Zhang, C., Chua, T.S.: Exploiting internal and external semantics for the clustering of short texts using world knowledge. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management. pp. 919–928. ACM (2009)

[48] Jendoubi, S., Martin, A., Liétard, L., Ben Hadj, H., Ben Yaghlane, B.: Maximizing positive opinion influence using an evidential approach. In: Poceeding of the 12th international FLINS conference (August 2016)

[49] Jendoubi, S., Martin, A., Liétard, L., Ben Hadj, H., Ben Yaghlane, B.: Two evidential data based models for influence maximization in twitter. Knowledge-Based Systems, http://dx.doi.org/10.1016/j.knosys.2017.01.014 (2017)

[50] Jendoubi, S., Martin, A., Liétard, L., Ben Yaghlane, B., Ben Hadj, H.: Dynamic time warping distance for message propagation classification in twitter. In: Proceeding of ECSQARU. pp. 419–428 (July 2015)

[51] Jendoubi, S., Martin, A., Liétard, L., Yaghlane, B.B.: Classification of message spreading in a heterogeneous social network. In: Proceeding of IPMU. pp. 66–75 (July 2014)

[52] Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: Proceeding of European conference on machine learning. pp. 137–142 (1998)

[53] Jousselme, A.L., Grenier, D., Bossé, E.: A new distance between two bodies of evidence. Information Fusion 2, 91–101 (2001)

[54] Jurvetson, S.: What exactly is viral marketing? Red Herring 78, 110–112 (2000)

[55] Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of KDD'03. pp. 137–146 (August 2003)

[56] Kempe, D., Kleinberg, J., Tardos, E.: Influential nodes in a diffusion model for social networks. In: Prceedings of the 32th International Colloquium on Automata, Languages and Programming. pp. 1127–1138 (2005)

[57] Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. Theory of computing 11(4), 105–147 (2015)

[58] Kermack, W.O., McKendrich, A.G.: A contribution to the mathematical theory of epidemics. In: Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character. 772, vol. 115, pp. 700–721 (August 1927)

[59] Kimura, M., Saito, K.: Tractable models for information diffusion in social networks. In: Proceedings of the 10th european conference on Principles and Practice of Knowledge Discovery in Databases: PKDD. pp. 259–271 (2006)

[60] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of KDD'07. pp. 420–429 (August 2007)

[61] Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. ACM Transactions on the Web (TWEB) 1(1), Article 5 (2007)

[62] Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10(8), 707–710 (1966)

[63] Li, D., Xu, Z.M., Chakraborty, N., Gupta, A., Sycara, K., Li, S.: Polarity related influence maximization in signed social networks. PLoS ONE 9(7), e102199 (July 2014)

[64] Ligett, T.M.: Interacting particle systems. Springer (1985)

[65] Liu, Z.g., Pan, Q., Dezert, J., Martin, A.: Adaptive imputation of missing values for incomplete pattern classification. Pattern Recognition 52, 85–95 (2016)

[66] Liu, Z.g., Pan, Q., Dezert, J., Mercier, G.: Credal c-means clustering method based on belief functions. Knowledge-Based Systems 74, 119–132 (2015)

[67] Lo, Y.W., Potdar, V.: A review of opinion mining and sentiment classification framework in social networks. In: Digital Ecosystems and Technologies, 2009. DEST'09. 3rd IEEE International Conference (Juin 2009)

[68] Martin, A., Osswald, C.: Human experts fusion for image classification. Information and Security: An International Journal, Special issue on Fusing Uncertain, Imprecise and Paradoxist Information (DSmT) 20, 122–143 (Mai 2006)

[69] Martin, A., Osswald, C.: Toward a combination rule to deal with partial conflict and specificity in belief functions theory. In: International Conference on Information Fusion. Québec, Canada (juillet 2007)

[70] Mohamadi-Baghmolaei, R., Mozafari, N., Hamzeh, A.: Trust based latency aware influence maximization in social networks. Engineering Applications of Artificial Intelligence 41, 195–206 (March 2015)

[71] Mostafa, M.M.: More than words: Social networks text mining for consumer brand sentiments. Expert Systems with Applications 40, 4241–4251 (2013)

[72] Nemhauser, G.L., Wolsey, L.A., Fisher, M.L.: An analysis of approximations for maximizing submodular set functions-i. Mathematical Programming 14(1), 265–294 (1978)

[73] Newman, M.E.J.: Networks: An introduction. Oxford University Press (2010)

[74] Othman, M., Hassan, H., Moawad, R., El-Korany, A.: Opinion mining and sentimental analysis approaches: A survey. Life Science Journal 11(4), 321–326 (2014)

[75] Petitjean, F., Inglada, J., Gancarski, P.: Satellite image time series analysis under time warping. IEEE Transactions on Geoscience and Remote Sensing 50(8), 3081–3095 (2012)

[76] Rudat, A., Buder, J.: Making retweeting social: The influence of content and context information on sharing news in twitter. Computers in Human Behavior 46, 75–84 (2015)

[77] Sabidussi, G.: The centrality index of a graph. Psychometrika 31(4), 581–603 (1966)

[78] Sakoe, H., Chiba, S.: A dynamic programming approach to continuous speech recognition. Proceedings of the Seventh International Congress on Acoustics, Budapest 3, 65–69 (1971)

[79] Shafer, G.: A mathematical theory of evidence. Princeton University Press (1976)

[80] Smets, P.: The Combination of Evidence in the Transferable Belief Model. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(5), 447–458 (1990)

[81] Smets, P.: Belief Functions: the Disjunctive Rule of Combination and the Generalized Bayesian Theorem. International Journal of Approximate Reasoning 9, 1–35 (1993)

[82] Smets, P.: The canonical decomposition of a weighted belief. In: Kaufman., M. (ed.) International Joint Conference on Artificial Intelligence. pp. 1896–1901. San Mateo, USA (1995)

[83] Smets, P.: Decision making in the TBM: the necessity of the pignistic transformation. International Journal of Approximate Reasonning 38, 133–147 (2005)

[84] Smets, P., Kennes, R.: The Transferable Belief Model. Artificial Intelligent 66, 191–234 (1994)

[85] Sriram, B., Fuhry, D., Demir, E., Ferhatosmanoglu, H., Demirbas, M.: Short text classification in twitter to improve information filtering. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 841–842. ACM (2010)

[86] Sung, J., Moon, S., Lee, J.G.: The influence in twitter: Are they really influenced? In: Behavior and Social Computing, pp. 95–105. Springer International Publishing (2013)

[87] Taylor, S.E.: Asymmetrical effects of positive and negative events: the mobilization-minimization hypothesis. Psychological bulletin 1(110), 67–85 (1991)

[88] Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of HLT-NAACL. pp. 252–259 (2003)

[89] Tsai, W.H., Fu, K.S.: Error-correcting isomorphisms of attributed relational graphs for pattern analysis. IEEE Transactions on Systems, Man and Cybernetics 9(12), 757–768 (December 1979)

[90] Wallisa, W., Shoubridgeb, P., Kraetzb, M., Rayc, D.: Graph distances using graph union. Pattern Recognition Letters 22(6-7), 701–704 (May 2001)

[91] Wang, H., Yang, Q., Fang, L., Lei, W.: Maximizing Positive Influence in Signed Social Networks, vol. 9483, pp. 356–367. Springer International Publishing (2015)

[92] Wang, Y., Wang, H., Li, J., Gao, H.: Efficient influence maximization in weighted independent cascade model. In: Proceedings of International Conference on Database Systems for Advanced Applications. pp. 49–64. Springer (2016)

[93] Wang, Y., Maple, C.: A novel efficient algorithm for determining maximum common subgraphs. In: The Proceedings of the Ninth International Conference on Information Visualisation. pp. 657–663. IEEE (July 2005)

[94] Wei, D., Deng, X., Zhang, X., Deng, Y., Mahadeven, S.: Identifying influential nodes in weighted networks based on evidence theory. Physica A 392(10), 2564–2575 (Mai 2013)

[95] Wen, S., Haghighi, M.S., Chen, C., Xiang, Y., Zhou, W., Jia, W.: A sword with two edges: Propagation studies on both positive and negative information in online social networks. IEEE Trans. Computers 64(3), 640–653 (2013)

[96] West, D.B.: Introduction to graph theory, vol. 2. Prentice hall Upper Saddle River (2001)

[97] Yager, R.R.: On the Dempster-Shafer Framework and New Combination Rules. Informations Sciences 41, 93–137 (1987)

[98] Yao, Q., Shi, R., Zhou, C., Wang, P., Guo, L.: Topic-aware social influence minimization. In: Proceedings of the 24th International Conference on World Wide Web. pp. 139–140. ACM (2015)

[99] Zhang, H., Dinh, T.N., Thai, M.T.: Maximizing the spread of positive influence in online social networks. In: Proceedings of ICDCS. pp. 317–326 (July 2013)

[100] Zhou, K., Martin, A., Pan, Q.: A similarity-based community detection method with multiple prototype representation. Physica A 438, 519–531 (November 2015)

[101] Zhou, K., Martin, A., Pan, Q., Liu, Z.g.: Median evidential c-means algorithm and its application to community detection. Knowledge-Based Systems 74, 69–88 (2015)

[102] Zubiaga, A., Spina, D., Martinez, R., Fresno, V.: Real-time classification of twitter trends. Journal of the Association for Information Science and Technology 66(3), 462–473 (2015)