

Une approche de visualisation analytique pour comparer les modèles de propagation dans les réseaux sociaux

Jason Vallet, Bruno Pinaud, et Guy Melançon*

*Univ. Bordeaux, LaBRI, UMR CNRS 5800
F-33405 Talence, France
prenom.nom@labri.fr

Résumé. Les modèles de propagation d'informations, d'influence et d'actions dans les réseaux sociaux sont nombreux et diversifiés rendant le choix de celui approprié à une situation donnée potentiellement difficile. La sélection d'un modèle pertinent pour une situation exige de pouvoir les comparer. Cette comparaison n'est possible qu'au prix d'une traduction des modèles dans un formalisme commun et indépendant de ceux-ci. Nous proposons l'utilisation de la réécriture de graphes afin d'exprimer les mécanismes de propagation sous la forme d'un ensemble de règles de transformation locales appliquées selon une stratégie donnée. Cette démarche prend tout son sens lorsque les modèles ainsi traduits sont étudiés et simulés à partir d'une plate-forme de visualisation analytique dédiée à la réécriture de graphe. Après avoir décrit les modèles et effectué différentes simulations, nous exhibons comment la plate-forme permet d'interagir avec ces formalismes, et comparer interactivement les traces d'exécution de chaque modèle grâce à diverses mesures soulignant leurs différences.

1 Introduction

Les réseaux sociaux sont l'objet d'une recherche intense depuis plusieurs années (Carrington et al., 2005; Newman et al., 2006; Scott et Carrington, 2011). Leur étude donne lieu à différentes questions concernant leur évolution, qu'il s'agisse d'analyser comment les interactions se sont mises en place, ou alors de comprendre l'état du système qu'ils décrivent. Parmi ces interrogations, l'étude des phénomènes de propagation dans les réseaux a suscité un intérêt soutenu au sein de la communauté, multipliant les domaines d'applications, allant de la sociologie (Granovetter, 1978; Macy, 1991) à l'épidémiologie (Hethcote, 2000; Dodds et Watts, 2005; Bertuzzo et al., 2010) en passant par la publicité virale et le placement de produits (Domingos et Richardson, 2001; Chen et al., 2010).

Nous nous intéressons dans cet article à l'étude de la propagation dans les réseaux sociaux. Notre objectif est de proposer une méthodologie permettant de comparer des modèles préexistants et documentés de propagation. Le grand nombre et les différentes variations de ces derniers offrent un assortiment de solutions compliquant le choix d'un modèle particulier. Afin de faciliter cette tâche, il convient de pouvoir comparer effectivement les modèles et non seulement les résultats finaux obtenus suite à leur application.

Cette ambition rejoint Kempe et al. (2003) qui proposent une généralisation de différents types de modèles de propagation. Ces résultats permettent de voir les modèles dans un cadre entièrement mathématique où chacun des algorithmes devient une solution à un problème d'optimisation commun. À l'inverse, nous adoptons une perspective résolument algorithmique dont l'objectif est de venir en appui à une approche exploratoire.

Il n'existe à notre connaissance pas de formalisme unifiant toutes les approches permettant d'effectuer une comparaison des modèles, de leur formulation, leur complexité ou leurs performances. La première contribution de cet article est donc de proposer un tel cadre unificateur basé sur un formalisme solide : la réécriture de graphes.

La propagation est généralement vue comme un phénomène global au réseau bien qu'elle émerge en réalité de la somme d'une multitude d'évènements y agissant localement. La plupart des modèles consistent donc en un ensemble de règles décrivant les situations dans lesquelles une entité peut influencer ses voisins. Bien que chacun de ces évènements soit décrit localement et succinctement, l'application répétée de transformations locales permet de faire émerger le comportement du modèle au niveau global. Dans ce formalisme, un modèle correspond alors à un ensemble de règles de transformation couplé à une stratégie qui régule et ordonne l'application de ces mêmes règles.

Les modèles auxquels nous nous intéressons par la suite considèrent un réseau social dont la topologie est fixée. Les règles décrivent alors comment évoluent les états des sommets du réseau. Dans leurs travaux, Kejžar et al. (2008) se sont intéressés, de façon similaire, à l'évolution du caractère topologique d'un réseau social. Partant d'un réseau pré-existant, les auteurs proposent une série de règles permettant de modifier les connexions entre les acteurs du réseau, autorisant ainsi la création ou suppression de liens. Leur travail rejoint donc notre approche consistant à exploiter la réécriture comme mécanisme pour exprimer leurs modèles d'évolution des réseaux. Cependant, leur article est davantage orienté vers l'analyse des résultats asymptotiques probabilistes sur l'évolution de la taille et la densité des réseaux ainsi produits.

L'intérêt de notre approche basée sur une description commune des modèles tient également à la possibilité d'étudier et de comparer ceux-ci de manière expérimentale. La plupart des travaux s'intéressent à l'objectif atteint au terme d'une propagation (couverture du réseau, vitesse de propagation, etc.), en revanche, il est plus difficile d'établir des résultats expliquant *comment* se déroule la propagation et *pourquoi* cet objectif est atteint. L'utilisation d'un formalisme commun nous permet, au contraire, la réalisation de ce type d'investigation.

De plus, cette méthodologie prend un sens particulier lorsque l'étude des modèles se fait de manière visuelle et interactive. En manipulant le modèle (en lançant des simulations, en isolant une règle, etc.), l'utilisateur est à même de développer une connaissance du modèle ainsi que de mesurer et suivre son comportement au fil du déroulement. Pour ces raisons, nous présentons une plate-forme de visualisation analytique – exploitant une version étendue de PORGY (Pinaud et al., 2012) (voir Fig. 1) – pour, simultanément, construire les réseaux et règles de réécriture, simuler la propagation selon différentes stratégies (*i.e.* les modèles) et comparer les traces d'exécution de ces dernières à l'aide de divers critères.

L'article introduit d'abord la terminologie propre aux modèles de propagation des réseaux et décrit deux modèles particuliers (section 2). Ces modèles sont ensuite exprimés à l'aide de règles de réécriture illustrant ainsi le pouvoir d'expression et l'utilisabilité du formalisme (section 3). Nous montrons enfin comment la plate-forme de visualisation peut être utilisée pour étudier les modèles et exhiber leurs différences (section 4).

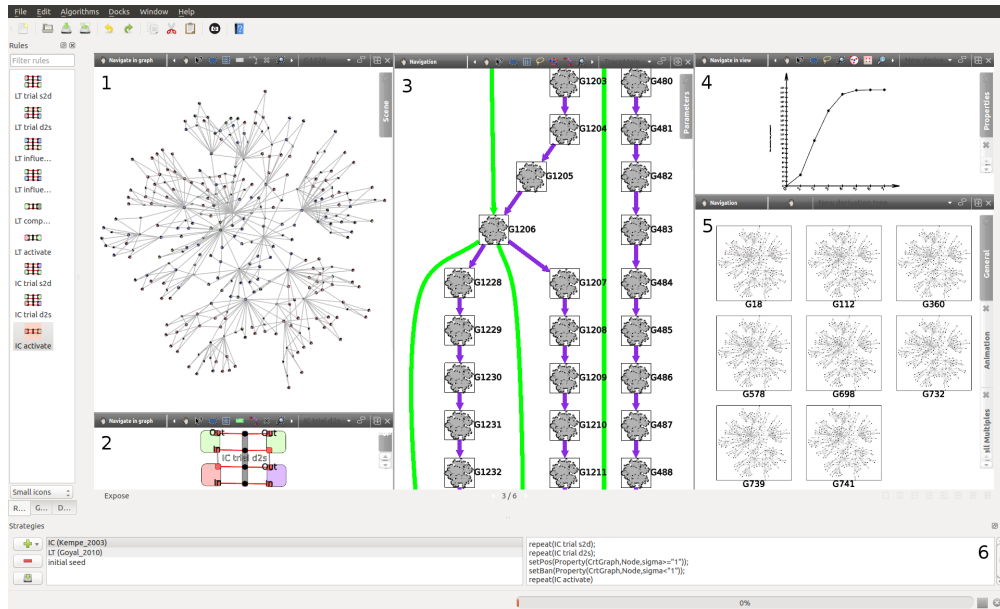


FIG. 1: Interface de PORGY : (1) le réseau social sur lequel on applique la propagation ; (2) édition d’une règle ; (3) portion de l’arbre de dérivation, conservant une trace complète des calculs réalisés (le graphe (1) représente un des sommet de celui-ci) ; (4) courbe montrant l’évolution du nombre de sommets actifs ; (5) autre représentation de l’arbre de dérivation ; (6) éditeur de stratégies.

2 Modélisation de la propagation dans les réseaux sociaux

Un réseau social (Brandes et Wagner, 2003) est un graphe $G = (V, E)$ constitué d’un ensemble d’individus (les *sommets* du graphe) V et d’un ensemble d’arêtes E liant les individus deux à deux (symbolisant par exemple une reconnaissance mutuelle). Deux individus sont voisins s’ils sont connectés par une arête et nous notons $N(w)$ l’ensemble des voisins d’un sommet w . La propagation dans un réseau se résume ainsi : lorsqu’un individu réalise une action spécifique (annoncer une nouveauté, transmettre une rumeur, partager une vidéo, *etc.*), il devient *actif*. Il indique alors son changement d’état à ses voisins leur offrant la possibilité de s’activer à leur tour et de réitérer ce processus pour l’ensemble de leurs propres voisins. L’activation se propage dès lors à travers l’ensemble du réseau.

Cette définition de la propagation est évidemment simplifiée. Les modèles de propagation existant possèdent tous plus ou moins de spécificités afin de les rendre le plus représentatif possible des phénomènes observables dans un véritable réseau. Ainsi, certains optent pour des activations entièrement probabilistes (*e.g* Chen et al. (2011); Wonyeol et al. (2012)) où la simple présence d’un voisin actif peut suffire pour permettre la propagation. D’autres s’appuient sur des seuils (*e.g* Watts (2002); Kempe et al. (2005); Goyal et al. (2010)) construits au fil du déroulement de la propagation et qui peuvent être utilisés pour mesurer l’influence d’un utilisateur sur ses voisins ou représenter sa tolérance à la réalisation d’une action (plus un

utilisateur est sollicité, plus il sera enclin à s'activer ou inversement).

Face à cette diversité, nous nous limitons dans la suite de l'article à illustrer la faisabilité de notre approche sur deux modèles représentatifs : un modèle à cascades indépendantes (**IC**, Kempe et al. (2003)) utilisé comme base pour de nombreux cas, et un modèle à seuils linéaires (**LT**, Goyal et al. (2010)) qui exploite un principe d'activation non probabiliste contrairement au modèle précédent :

Le modèle à cascades indépendantes IC. Ce modèle comporte de nombreuses variations (*e.g.* Gomez-Rodriguez et al. (2010); Watts (2002)) permettant, par exemple, la propagation d'opinions divergentes dans un même réseau (Chen et al., 2011). Nous le décrivons sous une forme basique, telle que proposée par Kempe et al. (2003).

Soit un sous-ensemble de sommets $A_0 \subset V$ *activés* au temps $t = 0$ et les probabilités $p_{v,w}$, définies pour toutes paires de sommets voisins $\{v, w\}$ pour représenter l'influence de $v \in A_0$ sur $w \notin A_0$. Une série de nouveaux ensembles de sommets activés A_{t+1} est calculé à partir de A_t . Pour chacun des sommets dans A_t , on visite les voisins $w \in N(v)$ qui n'ont pas déjà été activés (mais qui peuvent déjà avoir été visités); en d'autres mots, $w \in N(v) \setminus \cup_{i=0}^t A_i$. Un sommet w peut alors devenir actif avec une probabilité $p_{v,w}$, auquel cas il est ajouté à A_{t+1} . L'algorithme s'arrête lorsque A_{t+k} est vide (pour $k \geq 0$).

Modèle à seuil linéaire LT. Ce modèle suit un déroulement différent du précédent. Il le rejoint cependant en ce qu'un sommet ne change plus d'état dès lors qu'il est activé. On suppose donné, soit aléatoirement, soit appris selon un historique d'actions connues, les probabilités $p_{v,w}$. Chaque sommet $w \in V$ est aussi équipé d'un *seuil* θ_w . Soit S_w l'ensemble des voisins du sommet w qui sont activés. On détermine l'ensemble A_{t+1} en calculant pour chaque sommet w non encore activé la valeur d'influence jointe $p_w(S) = 1 - \prod_{v \in S_w} (1 - p_{v,w})$. Le sommet w devient ainsi actif dès que l'influence de ses voisins excède son seuil d'activation, c'est à dire lorsque $p_w(S) \geq \theta_w$.

3 Traduction des modèles et règles de réécriture

Nous abordons ici la description du formalisme qui va permettre d'exprimer les modèles présentés ci-dessus dans un langage algorithmique commun.

Les formalismes graphiques sont utiles lorsqu'il s'agit de décrire des structures complexes de manière visuelle et intuitive comme c'est le cas, par exemple, pour les diagrammes UML, la représentation de preuves, le design de micro-processeurs, les workflows, *etc.* La réécriture de graphes développée en informatique est un concept central des modèles de calcul distribué, et de la modélisation des systèmes complexes en général. D'un point de vue théorique, la réécriture de graphes possède de solides fondements logiques, algébriques et catégoriels (Courcelle, 1990; Rozenberg, 1997); tandis que d'un point de vue pratique, la réécriture possède de nombreux champs d'application en spécification formelle, programmation et simulation (Ehrig et al., 1997a,b). Plusieurs langages et outils existent pour manipuler ce formalisme tels que PROGRES (Schürr et al., 1997), GROOVE (Rensink, 2003), GrGen (Geiß et al., 2006) et GP (Plump, 2009).

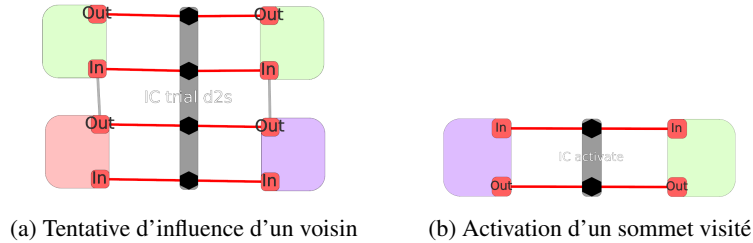


FIG. 2: Règles utilisées pour l'expression du modèle à cascades indépendantes. Les sommets activés sont en verts et les sommets visités sont en violets. Les sommets rouges sont dans un état non activé (mais ils peuvent néanmoins avoir été visités).

3.1 Réécriture de graphes

Les éléments de base du calcul de la réécriture sont des sommets du graphe, additionnellement équipés de *ports*, auxquels les arêtes vont se connecter. Plus généralement, les sommets, ports et arêtes vont avoir des propriétés associées à une valeur (par exemple la probabilité $p_{v,w}$ ou le nom donné à un port) qui permettront de les distinguer entre eux, une combinaison spécifique des ces propriétés pouvant être identifiée comme un *état*.

Une *règle* de réécriture consiste en un couple $L \Rightarrow R$ où L et R sont eux-mêmes des graphes (souvent petits). L et R sont respectivement appelés les membres *gauche* et *droit* de la règle. L'*application* d'une règle sur un graphe G se fait en localisant dans G un sous-graphe H isomorphe à L et en le "remplaçant" par R . La notion d'isomorphisme doit toutefois être étendue pour tenir compte des états des sommets, des ports et des arêtes. La règle doit également, le cas échéant, préciser comment traiter les arêtes incidentes aux ports de H qui ne sont pas mentionnées dans L . De plus, une règle peut aussi calculer les valeurs de plusieurs propriétés de R en fonction de celles de H .

Des exemples de règles sont donnés figure 2. La règle 2a concerne une paire de sommets, l'un dans l'état activé (*vert*), l'autre étant non activé (*rouge*), connectés par une arête allant du port *In* du sommet activé vers le port *Out* du sommet non activé. La règle maintient la connexion entre les sommets et modifie l'état du sommet *rouge*, le faisant passer dans l'état *violet* signifiant que le sommet a été visité (ses propriétés ont été lues et il est possible de tenter de l'activer). La règle 2b ne concerne qu'un seul sommet ; son application est donc potentiellement réalisable sur tout sommet du graphe à condition qu'il soit dans le même état (ici visité, symbolisé par la couleur *violette*). La règle consiste simplement à modifier l'état du sommet, le faisant passer de l'état *violet* à l'état *vert* (activé).

Le défi consiste la plupart du temps à savoir prédire le comportement de la réécriture répétée de règles sur un graphe. En effet, l'exécution des règles n'est pas déterministe puisque leur ordre d'application n'est, *a priori*, pas précisé, mais aussi parce qu'elles peuvent être appliquées sur de multiples instances H du membre gauche L de la règle. Il devient alors intéressant de savoir si le calcul de la réécriture converge et s'il est confluent.

Dans cet optique, il peut également être tentant de chercher à conditionner l'ordre d'application des règles afin de guider le comportement du calcul. A cette fin, il est possible de définir une *stratégie* d'application des règles. Une stratégie permet de choisir un ensemble des règles à appliquer, préciser leur ordre, le nombre de répétitions, et l'endroit où celles-ci peuvent être ap-

pliquées. Pour plus de détails sur le langage de stratégies utilisé par la plate-forme PORGY, sa formalisation et ses propriétés en tant que langage formel, le lecteur pourra consulter l'article de Fernandez et al. (2014).

3.2 Traduction des modèles de propagation

Le premier défi qui se pose à nous est de pouvoir donner, pour chacun des modèles présentés dans la section 2, un ensemble de règles et une stratégie d'application de ces dernières qui permet d'émuler le fonctionnement du modèle.

Notre démarche de traduction d'un modèle de propagation quelconque en une série de règles de réécriture et leur stratégie d'application est aisément généralisable. Celle-ci a pu être appliquée à tous les modèles rencontrés dans la bibliographie étudiée. Pour la clarté de la discussion concernant les étapes à suivre, nous présentons uniquement la traduction du modèle de propagation à cascades indépendantes (section 2). Ce dernier illustre tout à fait les opérations à réaliser et toute procédure de traduction d'autres modèles suit un déroulement similaire.

Le motif (membre gauche d'une règle) principal à rechercher pour faire évoluer la propagation consiste à identifier un couple de sommets voisins dont l'un est activé et l'autre ne l'est pas. Il est ainsi nécessaire de conserver pour chaque sommet son état actuel. De manière similaire, chaque arête devra préserver les probabilités d'influence $p_{v,w}$ et $p_{w,v}$ que ses extrémités v et w pourront imposer l'une sur l'autre lorsque l'un des sommets s'activera.

La stratégie employée consiste, pour chaque sommet non actif, à calculer puis stocker l'influence de ses voisins actifs en ne conservant que la valeur pour le sommet qui a l'influence la plus forte. Le sommet non actif passe alors dans l'état visité (règle de la figure 2a). Le parcours du voisinage est, de cette manière, contrôlé par la stratégie tandis que les actions à effectuer sur le réseau sont contrôlées par les règles.

L'application de la règle entraîne plusieurs modifications locales pour chaque couple de sommets actif(v)/visité(w) : *a*) génération d'un nombre aléatoire $r \in [0, 1]$; *b*) stockage dans une propriété σ_w marquant l'influence maximum rencontrée jusqu'à présent (initialisé à 0) des voisins actifs de w telle que $\sigma_w = \max\left(\frac{p_{v,w}}{r}, \sigma_w\right)$; et *c*) nous marquons l'arête liant v et w afin d'exclure cette configuration particulière lors des prochaines recherches de correspondances. Ce dernier point représente le fait qu'un noeud actif ne peut tenter d'activer un voisin plus d'une fois. Après avoir fait le tour des voisins, si w a été suffisamment influencé ($\sigma_w > 1$), il devient actif à son tour (règle de la figure 2b).

La procédure décrite ci-dessus forme une stratégie qui sera répétée tant qu'un sommet actif peut influencer un de ses voisins (il reste une arête non marquée qui permet d'appliquer la règle 2a). Chaque application de règle entraîne l'ajout d'un sommet sur l'arbre de dérivation et d'une arête (de couleur violette) pour montrer la succession des opérations. Les points de départ et d'arrivée d'une stratégie (enchaînement de plusieurs règles) sont eux représentés par une arête verte (Fig. 1). Cet arbre peut rapidement atteindre une taille conséquente, rendant la lisibilité difficile. Nous pouvons néanmoins le filtrer et ne conserver que les arêtes vertes et les sommets correspondants (Fig. 4).

4 Visualisation analytique et comparaison des modèles

Nous détaillons dans cette partie comment la plate-forme de visualisation PORGY (Pinaud et al., 2012) est utilisée pour comparer deux applications des modèles de propagation présentés au début de cet article. Nous avons utilisé le modèle de Wang et al. (2006) pour générer un réseau social aléatoire de 300 sommets. Le réseau obtenu a 597 arêtes. Les conditions de départ sont identiques pour les deux modèles : même ensemble initial de sommets activés et même distribution de probabilités d'influence entre les sommets. Notre objectif n'est pas de montrer que tel modèle de propagation est meilleur que tel autre (ceci nécessiterait de nombreuses simulations pour calculer les résultats moyens sur les modèles probabilistes) mais plutôt de comprendre comment les modèles fonctionnent ¹.

Les applications successives de la stratégie décrite précédemment permettent aux sommets actifs de transmettre l'*information* ou l'*action* représentant le sujet de la propagation à leur voisins. Chaque exécution de règle va créer un état intermédiaire du graphe d'origine qui sera conservé dans la trace de la propagation (passage des différents sommets de *non visité* à *visité* puis potentiellement *actif*). Cet historique va pouvoir être exploité pour étudier et comparer le graphe à un instant donné ou pour reconstituer et suivre le chemin emprunté par le processus d'activation des sommets.

Un arbre de dérivation (voir fig.1) est ainsi créé et maintenu pour fournir toutes ces informations. En visualisant des états successifs, nous pouvons observer cette progression. La figure 3 présente quelques vignettes d'une vue de type *Small-Multiples*, qui est une partie du graphe analysé, et montre l'évolution de l'état de ses sommets. Les différents temps t représentent les applications successives des stratégies (un sommet non visité à t peut donc se retrouver activé à $t + 1$). L'arbre de dérivation nous permet immédiatement de montrer quel est le modèle qui nécessite le moins d'étapes de calcul ou le moins de lancements de stratégie avant d'arriver à terme car sa branche est la plus courte (figure 4). Nous avons ainsi une première approximation de la complexité en temps des algorithmes.

Lier la profondeur de l'arbre (donc le temps) avec d'autres mesures nous permet de considérer l'évolution de différents paramètres tout au long de la propagation. Nous pouvons, par exemple, aborder la notion de *vitesse de propagation*, valeur indiquant l'évolution du nombre de sommets actifs en fonction du temps. La figure 4 présente l'évolution de cette valeur pour une exécution des modèles à cascades indépendantes (partie droite, courbe du haut) et à seuils linéaires (courbe du bas). Les courbes présentées n'ont pas les mêmes échelles puisqu'elles sont calculées indépendamment pour chacun des modèles. Malgré ceci, nous pouvons observer que le modèle à cascades va parvenir à activer environ 80% des sommets contre seulement 18% pour celui à seuils (pour un même nombre d'étapes de réécritures), démontrant le fort impact sur les performances des modèles, dans un premier temps, des valeurs utilisées pour l'initialisation des probabilités d'influence, et dans un second temps, du choix de l'ensemble de sommets initialement activés. Nous avons utilisé une loupe (fonctionnalité de PORGY) sur le haut de chaque axe pour rendre les valeurs lisibles. Nous pouvons aussi noter qu'après la première application de la stratégie, le nombre de sommets actifs est très proche pour chacun des modèles, les différences apparaissant et se confirmant par la suite.

1. Tous les éléments (logiciels, données, ...) nécessaires pour reproduire les résultats présentés sont disponibles à <http://tulip.labri.fr/TulipDrupal/?q=porgy>

Visualisation analytique et propagation sur réseaux

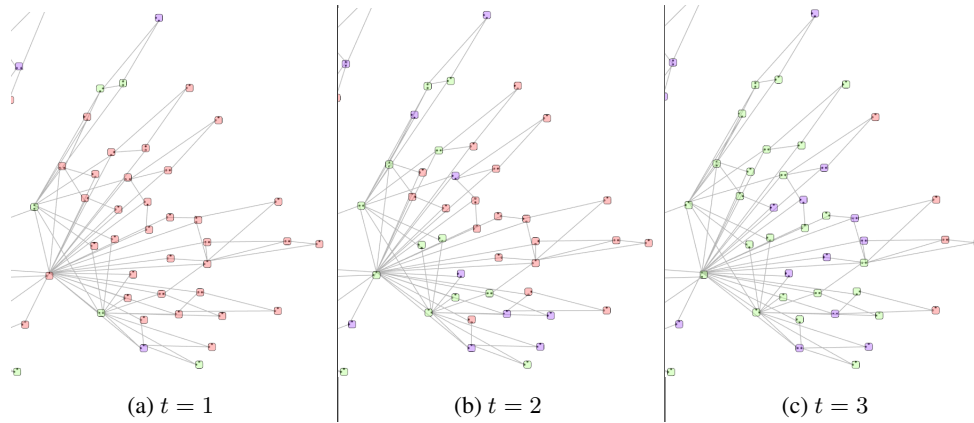


FIG. 3: Sous-communauté extraite du graphe montrant l'évolution de la propagation à différents pas de temps. Deux sommets sont activés à $t = 0$ et seulement six ne seront que visités à la fin de la propagation. (Légende : vert = actif, violet = visité, rouge = non visité et non actif)

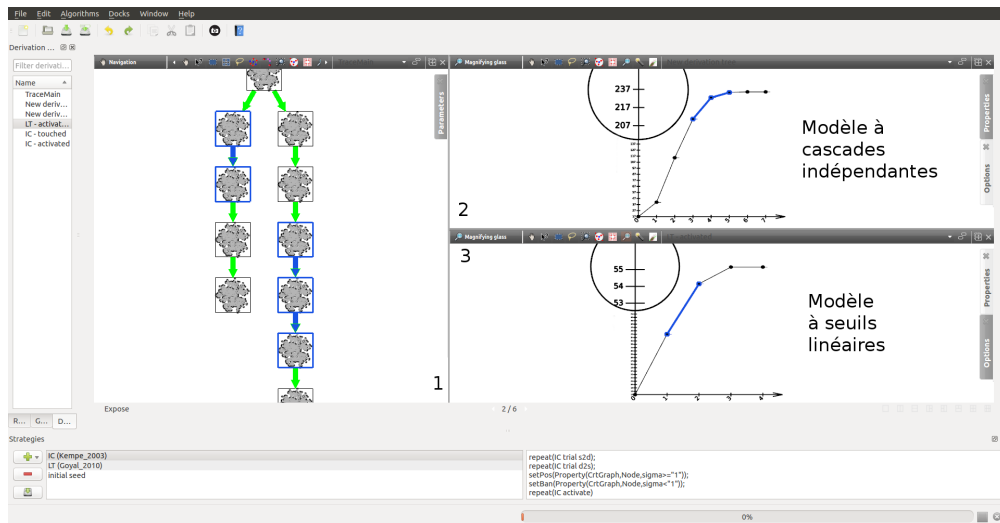


FIG. 4: Interface de PORGY : (1) portion de l'arbre de dérivation simplifié (affichage filtré pour présenter uniquement les transitions entre stratégies) ; (2) et (3) Évolution de la *vitesse de propagation* (nombre de sommets actifs) en fonction de la profondeur dans la branche de l'arbre de dérivation pour les modèles à cascades indépendantes et à seuils linéaires.

Il peut aussi être intéressant de voir l'état du graphe quand le nombre de sommets activés atteint un seuil. Puisque les différentes vues sont liées, la sélection d'un sommet/arête (en bleu sur la figure 4) de l'arbre de dérivation entraîne sa sélection dans la courbe et *vice-versa*. De manière similaire, la sélection d'un sommet lors d'une étape de la propagation sera immédiatement répercutée à l'ensemble des étapes contenant ce même élément, rendant la sélection visible même sur les sommets de l'arbre de dérivation. D'après la méthodologie employée par l'application PORGY, tant qu'un élément n'est pas modifié par une règle, il n'est jamais changé. En conséquence, la sélection d'un sommet d'intérêt dans l'un des graphes intermédiaires représentant le réseau permet de savoir directement quand cet élément a changé d'état, surtout si l'on travaille sur la version complète de l'arbre de dérivation (montrant le détail des applications de règles).

Finalement, le type de mesure évoluant selon le temps peut être généralisé à d'autres propriétés des modèles de propagation. La notion de *sommet visité*, introduite précédemment, peut présenter un intérêt, dans le cas où un message doit seulement être vu et non nécessairement redistribué ou propagé par les utilisateurs. Cette *vitesse de connaissance* du contenu de la propagation est observable de manière similaire à la vitesse de propagation. De plus, en considérant ces deux mesures, nous pouvons en proposer une troisième exprimant l'*efficacité d'une propagation*, calculable grâce au rapport entre le nombre de sommets activés à un instant t et ceux visités/influencés au moment précédent $t - 1$.

5 Conclusion et travaux futurs

Nous avons présenté un formalisme basé sur la réécriture de graphes vu comme un langage commun à l'expression de tous les modèles de propagation sur réseaux. Lorsque la propagation n'entraîne pas de modifications de la topologie du graphe, le modèle consiste en un ensemble réduit de règles gérant les transitions d'état des sommets du graphe. Le stockage des états des sommets, les règles et le langage de stratégie qui pilote leur application facilite la gestion du caractère probabiliste des modèles.

Nous envisageons d'étendre notre étude à un panel plus large de modèles de propagation afin de démontrer le caractère "universel" de notre approche. Cela exige aussi de pouvoir multiplier les simulations sur des réseaux de tailles conséquentes. Cet aspect pose un défi en raison de la complexité liée à la recherche de motifs correspondant aux membres gauches des règles – d'abord parce que nous faisons face à un problème *NP-Complet* (isomorphisme de sous-graphes), mais aussi à cause de l'explosion combinatoire qu'elle engendre et qui doit être gérée à l'aide du langage de stratégie (d'application des règles).

La formulation des modèles à l'aide de réécritures offre une possibilité nouvelle qui permettra de combiner propagation dans le réseau et évolution de la topologie du réseau sur lequel la propagation a lieu. Là encore, un langage de stratégie facilitera la gestion de l'application simultanée ou alternée de ces deux types de transformations. Il n'existe pas, à notre connaissance, de tels modèles. Nous espérons ainsi pouvoir proposer des modèles réalistes d'évolution de réseaux, dont le réalisme tiendrait à la fois aux caractères structurels des réseaux produits, mais aussi à leur qualité en terme de circulation de l'information.

Remerciements

Ce travail a été en partie financé par les projets ANR JCJC EVIDEN 2010-JCJC-0201-01 et le projet PIA SpeedData (PIAO17298-398711). Les auteurs tiennent également à remercier Hélène Kirchner pour ses commentaires, remarques et conseils avisés.

Références

- Bertuzzo, E., R. Casagrandi, M. Gatto, I. Rodriguez-Iturbe, et A. Rinaldo (2010). On spatially explicit models of cholera epidemics. *J. of The Royal Society Interface* 7(43), 321–333.
- Brandes, U. et D. Wagner (2003). Visone – analysis and visualization of social networks. In *Graph Drawing Software*, pp. 321–340. Springer.
- Carrington, P., J. Scott, et S. Wasserman (2005). *Models and Methods in Social Network Analysis*. Structural Analysis in the Social Sciences. Cambridge University Press.
- Chen, W., A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincón, X. Sun, Y. Wang, W. Wei, et Y. Yuan (2011). Influence maximization in social networks when negative opinions may emerge and propagate. In *11th SIAM Int. Conf. on Data Mining, SDM*, pp. 379–390.
- Chen, W., C. Wang, et Y. Wang (2010). Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *16th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD '10*, pp. 1029–1038.
- Courcelle, B. (1990). Graph Rewriting : An Algebraic and Logic Approach. In J. van Leeuwen (Ed.), *Handbook of Theoretical CS, Volume B : Formal Models and Semantics*, pp. 193–242. Elsevier Science Publishers and MIT Press.
- Dodds, P. et D. Watts (2005). A generalized model of social and biological contagion. *J. of Theoretical Biology* 232(4), 587–604.
- Domingos, P. et M. Richardson (2001). Mining the network value of customers. In *7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD '01*, pp. 57–66.
- Ehrig, H., G. Engels, H.-J. Kreowski, et G. Rozenberg (Eds.) (1997a). *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 2 : Applications, Languages, and Tools*. World Scientific.
- Ehrig, H., H.-J. Kreowski, U. Montanari, et G. Rozenberg (Eds.) (1997b). *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 3 : Concurrency, Parallelism, and Distribution*. World Scientific.
- Fernandez, M., H. Kirchner, et B. Pinaud (2014). Strategic Port Graph Rewriting : An Interactive Modelling and Analysis Framework. In D. Bošnački, S. Edelkamp, A. L. Lafuente, et A. Wijs (Eds.), *GRAPHITE 2014*, Volume 159 of *EPTCS*, pp. 15–29.
- Geiß, R., G. V. Batz, D. Grund, S. Hack, et A. Szalkowski (2006). GrGen : A Fast SPO-Based Graph Rewriting Tool. In A. Corradini, H. Ehrig, U. Montanari, L. Ribeiro, et G. Rozenberg (Eds.), *ICGT*, Volume 4178 of *LNCS*, pp. 383–397. Springer.
- Gomez-Rodriguez, M., J. Leskovec, et A. Krause (2010). Inferring networks of diffusion and influence. In *16th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD '10*, pp. 1019–1028.

- Goyal, A., F. Bonchi, et L. V. Lakshmanan (2010). Learning influence probabilities in social networks. In *3rd ACM Int. Conf. on Web Search and Data Mining, WSDM '10*, pp. 241–250.
- Granovetter, M. (1978). Threshold models of collective behavior. *American J. of Sociology* 83(6), 14–20.
- Hethcote, H. (2000). The mathematics of infectious diseases. *SIAM Review* 42(4), 599–653.
- Kejžar, N., Z. Nikoloski, et V. Batagelj (2008). Probabilistic inductive classes of graphs. *J. of Mathematical Sociology* 32(2), 85–109.
- Kempe, D., J. Kleinberg, et É. Tardos (2003). Maximizing the spread of influence through a social network. In *Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD '03*, pp. 137–146.
- Kempe, D., J. Kleinberg, et É. Tardos (2005). Influential nodes in a diffusion model for social networks. In L. Caires, G. Italiano, L. Monteiro, C. Palamidessi, et M. Yung (Eds.), *Automata, Languages and Programming*, Volume 3580 of *LNCS*, pp. 1127–1138. Springer.
- Macy, M. W. (1991). Chains of cooperation : Threshold effects in collective action. *American Sociological Review* 56(6), 730–747. ISSN : 0003-1224.
- Newman, M., A.-L. Barabási, et D. J. Watts (2006). *The structure and dynamics of networks*. Princeton Studies in Complexity. Princeton University Press.
- Pinaud, B., G. Melançon, et J. Dubois (2012). Porgy : A visual graph rewriting environment for complex systems. *Computer Graphics Forum* 31(3), 1265–1274.
- Plump, D. (2009). The Graph Programming Language GP. In S. Bozapalidis et G. Rahonis (Eds.), *CAI*, Volume 5725 of *LNCS*, pp. 99–122. Springer.
- Rensink, A. (2003). The GROOVE Simulator : A Tool for State Space Generation. In J. L. Pfaltz et al. (Ed.), *AGTIVE*, Volume 3062 of *LNCS*, pp. 479–485. Springer.
- Rozenberg, G. (Ed.) (1997). *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1 : Foundations*. World Scientific.
- Schürr, A., A. J. Winter, et A. Zündorf (1997). The PROGRES Approach : Language and Environment. In *Ehrig et al. (1997a)*, pp. 479–546. World Scientific.
- Scott, J. et P. J. Carrington (2011). *The SAGE Handbook of Social Network Analysis*. SAGE.
- Wang, L., F. Du, H. P. Dai, et Y. X. Sun (2006). Random pseudofractal scale-free networks with small-world effect. *The European Physical J. B - Condensed Matter and Complex Systems* 53(3), 361–366.
- Watts, D. J. (2002). A simple model of global cascades on random networks. *Proc. of the National Academy of Sciences* 99(9), 5766–5771.
- Wonyeol, L., K. Jinha, et Y. Hwanjo (2012). Ct-ic : Continuously activated and time-restricted independent cascade model for viral marketing. In *Proc. of ICDM*, pp. 960–965.

Summary

Numerous propagation models in social networks describing social influence can be found in the literature, making it difficult to select an appropriate model in a given situation. Selecting the most relevant model requires the ability to objectively compare them. This comparison

Visualisation analytique et propagation sur réseaux

can only be made at the cost of describing models based on a common formalism and independent from them. We propose to use graph rewriting to formally describe the propagation mechanisms as local transformation rules applied according to a strategy. This approach makes complete sense when supported by a visual analytics framework dedicated to graph rewriting. The paper first presents several models and illustrates them through selected simulations. We then show how our visual analytics framework allows to interactively manipulate models, and underline their differences based on measures computed on simulation traces.