Apprentissage de représentations pour la prédiction de propagation d'information dans les réseaux sociaux

Simon Bourigault

THÈSE DE DOCTORAT DE
l'UNIVERSITÉ PIERRE ET MARIE CURIE

Spécialité

Informatique

École doctorale Informatique, Télécommunications et Électronique (Paris)

Présentée par

Simon BOURIGAULT

Pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE

Sujet de la thèse :

Apprentissage de représentations pour la prédiction de propagation d'information dans les réseaux sociaux

soutenue le 10 novembre 2016

devant le jury composé de :

M. Patrick Gallinari	Professeur	Directeur de thèse
M. Sylvain Lamprier	Maître de conférence	Encadrant de thèse
Mme. Christine Largeron	Professeure	Examinatrice
M. Christophe Marsala	Professeur	Examinateur
M. Fabrice Rossi	Professeur	Rapporteur
M. Julien Velcin	Maître de conférence	Rapporteur

Résumé Dans ce manuscrit, nous étudions la diffusion d'information dans les réseaux sociaux en ligne. Des sites comme Facebook ou Twitter sont en effet devenus aujourd'hui des *media d'information* à part entière, sur lesquels les utilisateurs échangent de grandes quantités de données. La plupart des modèles existant pour expliquer ce phénomène de diffusion sont des modèles génératifs, basés sur des hypothèses fortes.En particulier, tous ces modèles sont basés sur le graphe social et font l'hypothèse que la diffusion d'information a lieu uniquement sur ce graphe, qu'il s'agisse des liens d'amitié sur Facebook, ou des *followers* sur Twitter. Cela pose plusieurs problèmes. Par exemple, pour des raisons de confidentialité, il est courant que le graphe social soit caché. Face à cette observation, nous considérerons dans ce manuscrit le problème de la prédiction de diffusion dans le cas où le graphe social est inconnu, et où seules les actions des utilisateurs peuvent être observées.

— Nous proposons, dans un premier temps, une méthode d'apprentissage du modèle *independent cascade* consistant à ne pas prendre en compte la dimension temporelle de la diffusion. Des résultats expérimentaux obtenus sur des données réelles montrent que cette approche permet d'obtenir un modèle plus performant et plus robuste.
— Nous proposons ensuite plusieurs méthodes de *prédiction de diffusion* reposant sur des techniques *d'apprentissage de représentations*. Celles-ci nous permettent de définir des modèles plus compacts, et plus robustes à la parcimonie des données.
— Enfin, nous terminons en appliquant une approche similaire au problème de *détection de source*, consistant à retrouver l'utilisateur ayant lancé une rumeur sur un réseau social. En utilisant des méthodes d'apprentissage de représentations, nous obtenons pour cette tâche un modèle beaucoup plus rapide et performant que ceux de l'état de l'art.

Abstract In this thesis, we study information diffusion in online social networks. Websites like Facebook or Twitter have indeed become information medias, on which users create and share a lot of data. Most existing models of the information diffusion phenomenon relies on strong hypothesis about the structure and dynamics of diffusion. In this document, we study the problem of diffusion prediction in the context where the social graph is unknown and only user actions are observed.

— We propose a learning algorithm for the independant cascades model that does not take time into account. Experimental results show that this approach obtains better results than time-based learning schemes.
— We then propose several representations learning methods for this task of diffusion prediction. This let us define more compact and faster models.
— Finally, we apply our representation learning approach to the source detection task, where it obtains much better results than graph-based approaches.

Computers grow so wise and incomprehensible that when your surpassing creations find the answers you asked for, you can't understand their analysis and you can't verify their answers. You have to take their word on faith.

Or you use information theory to flatten it for you, to simplify reality and pray to whatever Gods survived the millennium that your honorable twisting of the truth hasn't ruptured any of its load-bearing pylons.

Maybe the Singularity happened years ago. We just don't want to admit we were left behind.

Peter Watts - *Blindsight*

Remerciements

Je tiens à remercier, dans le désordre :

Mes encadrants, messieurs Lamprier et Gallinari pour leur accompagnement et leurs conseils au cours de ces quatre années de travail, ainsi que pour tous leurs retours durant la rédaction du présent manuscrit.

Les rapporteurs pour leurs remarques sur la première version du manuscrit, ainsi que l'ensemble du jury.

L'équipe MLIA et mes collègues thésards en particulier pour l'atmosphère de travail au sein du laboratoire.

L'ensemble du personnel administratif et technique du laboratoire, sans qui aucune recherche ne pourrait avoir lieu.

Mes parents, pour leur soutien et leur présence tout au long de mes études dont ce manuscrit constitue aujourd'hui l'aboutissement.

Ma compagne ainsi que mes amis et anciens camarades de master, pour tout.

Sommaire

Résumé des principales notations utilisées

N	Nombre d'utilisateurs étudiés
$U = \{u_0, u_1, \ldots, u_{N-1}\}$	Ensemble des utilisateurs
E	Ensemble des liens utilisateurs (orientés)
Preds_j	Ensemble des prédécesseurs de u_j
Succs_i	Ensemble des successeurs de u_i
$D = \{(u_i, t_i^D), (u_j, t_j^D), \ldots\}$	Un épisode de diffusion
t_i^D	Instant auquel u_i devient infecté dans D. Vaut $+\infty$ si u_i n'est jamais infecté.
U_t^D	Ensemble des utilisateurs infectés dans D avant le temps t, i.e $t_i^D < t$.
U_∞^D ou U^D	Ensemble des utilisateurs infectés dans D.
s_D	Utilisateur source de l'épisode D
$\hat{U}^D$	Ensemble des utilisateurs infectés dans D moins l'utilisateur source : $\hat{U}^D = U^D \setminus \{s_D\}$
$\mathcal{D}$	Ensemble des épisodes de diffusion
$\mathcal{D}_{i,j}^?$	Ensemble des épisodes de diffusion où il est possible que u_i ait contaminé u_j, i.e. tels que $u_i \in U_\infty^D$ et $u_j \in U_\infty^D$ avec $t_i^D < t_j^D$. On parlera aussi d'*exemples positifs* ou de *co-participations* pour le couple (u_i, u_j).
$\mathcal{D}_{i,j}^-$	Ensemble des épisodes de diffusion où il est impossible que u_i ait réussi à contaminer u_j, i.e. tels que $t_i^D < \infty$ et $t_j^D = \infty$. On parlera aussi de *contre-exemples* pour le couple (u_i, u_j).
$p_{i,j}$	Probabilité de transmission d'information de u_i à u_j
P_j^D	Probabilité que u_j soit infecté dans D
d	Nombre de dimensions de l'espace de représentation
$z_i \in \mathbb{R}^d$	Représentation-source de u_i dans $\mathbb{R}^d$
$\omega_i \in \mathbb{R}^d$	Représentation-récepteur de u_i dans $\mathbb{R}^d$
Q	Taille du dictionnaire utilisé pour représenter le contenu.
$w_D \in \mathbb{R}^Q$	Vecteur représentant le contenu associé à l'épisode D
$f_\theta : \mathbb{R}^Q \to \mathbb{R}^d$	Fonction paramétrée permettant d'obtenir une représentation du contenu dans $\mathbb{R}^d$

Chapitre 1

Introduction

1.1 Développement des réseaux sociaux en ligne

Au cours des dix dernières années, les réseaux sociaux en ligne (ou OSN, *Online Social Networks*) ont pris une importance capitale dans la vie personnelle et professionnelle de millions, voire de milliards, de personnes. Facebook, lancé en 2004 et ouvert au public en 2006, compte aujourd'hui près d'un milliard d'utilisateurs quotidiens. De leur coté, les utilisateurs de Twitter, lancé en 2006, génèrent environ 500 millions de tweets chaque jour, dans 35 langues différentes.

D'une façon plus générale, l'émergence du « web 2.0 » [O'Reilly, 2005] a fait des utilisateurs le moteur de nombreux services en ligne. Beaucoup de sites internet proposent aujourd'hui une personnalisation propre à chaque utilisateur dans divers domaines : relations professionnelles (LinkedIn, Viadeo), création de contenus artistiques (Youtube, DeviantArt, Soundcloud...), évaluation de produits (Amazon, Epinions, GoodReads), etc..

Ainsi, lorsqu'il définit le terme « web 2.0 » en 2005 [O'Reilly, 2005], O'Reilly déclare :

> « Le service devient automatiquement meilleur à mesure que son nombre d'utilisateurs augmente. »

Le terme important est ici « automatiquement ». Selon O'Reilly, le service devient meilleur *sans qu'aucune action* ne soit nécessaire de la part du service lui-même : ce sont les utilisateurs qui, en agissant et en interagissant par le biais de ce service, lui donnent une utilité, un intérêt et donc une valeur. Pour cette raison, la plupart des grands réseaux sociaux en ligne, à commencer par Facebook et Twitter, proposent des API permettant à des développeurs tiers d'intégrer certaines fonctionnalités de ces réseaux sur d'autres sites. Il est aujourd'hui possible, sur internet, de « liker », partager, tweeter, commenter ou évaluer pratiquement n'importe quel contenu. Ces actions sont généralement visibles des autres utilisateurs, et contribuent à la valeur du contenu concerné.

L'importance de ces interactions a également fait des réseaux sociaux en ligne un *média d'information* à part entière. Les utilisateurs sont aujourd'hui en mesure d'accéder à une très grande quantité d'information par le biais des réseaux sociaux, et de partager cette information avec de nombreuses autres personnes à travers le monde. S'ils se contentent parfois de relayer des informations issues des médias traditionnels (articles de journaux, podcast de radio, etc.), les utilisateurs peuvent aussi devenir des *sources* : les événements importants sont ainsi souvent rapportés et discutés en premier lieu sur Twitter. Chaque jour, des contenus de natures variées partagés de cette façon « font le buzz » et deviennent mondialement connus en étant rapidement vus et relayés par de nombreuses personnes et sites web. Ce mode de fonctionnement a d'ailleurs motivé le développement de techniques de marketing dit « viral », consistant pour les annonceurs à encourager leurs clients à diffuser un contenu publicitaire, de manière à atteindre d'autres clients potentiels grâce au bouche-à-oreille.

1.2 Diffusion d'information

Dans ce manuscrit, nous étudions ce phénomène de *diffusion d'information*. Les premiers travaux sur ce sujet sont issus des sciences sociales et datent des années 70.

Le modèle fondateur de Bass [Bass, 1969] prend la forme d'équations différentielles régissant l'évolution du nombre de consommateurs ayant adopté un produit considéré. D'autres modèles basés sur le même principe mais intégrant plus de paramètres dans leur modélisation ont ensuite été proposés [Newman, 2003, Hethcote, 2000].

Plus récemment, la grande quantité de données rendues disponibles par le développement des réseaux sociaux a permis l'application de modèles basés sur le graphe social. Ceux-ci considèrent l'information comme un virus infectant progressivement les individus d'une population en passant de l'un à l'autre en suivant les arcs d'un graphe social. Contrairement au modèle de Bass, ces modèles visent à modéliser ou à prédire l'infection de *chaque* utilisateur, et non pas seulement le taux d'infection d'une population fixée. Les plus classiques sont l'*Independent Cascade Model* (IC) et le *linear Threshold Model* (LT) [Kempe et al., 2003, Goldenberg et al., 2001, Granovetter, 1978].

Dans le modèle IC, chaque arête du graphe est associée à une probabilité de transmission, indiquant la probabilité pour un utilisateur donné d'infecter chacun de ses voisins : le modèle est *centré sur l'émetteur*. Dans le modèle LT, en revanche, chaque utilisateur est associé à un seuil indiquant à partir de quel niveau d'influence externe il devient lui-même infecté : le modèle est *centré sur le receveur*. À partir de ces idées de base, différentes approches visant notamment à prendre en compte la dimension temporelle de la diffusion ont été proposées.

Tous ces modèles basés sur le graphe social font l'hypothèse que la diffusion d'information a lieu uniquement sur ce graphe, qu'il s'agisse des liens d'amitié sur Facebook, ou des

followers sur Twitter. Cela pose plusieurs problèmes. Tout d'abord, les utilisateurs étant souvent inscrits sur plusieurs sites internet, la diffusion peut avoir lieu sur plusieurs réseaux en parallèle. De plus, les liens explicites renseignés sur un réseau social en ligne ne sont pas toujours les plus pertinents pour expliquer la diffusion d'information [Cha et al., 2010]. Enfin, pour des raisons de confidentialité, il est courant que le graphe social soit caché.

Face à ces observations, nous considérerons dans ce manuscrit le problème de la prédiction de diffusion dans le cas où le graphe social est inconnu, et où seules les actions des utilisateurs peuvent être observées.

1.3 Projection des utilisateurs

Dans la plupart de nos travaux présentés ici, nous utilisons *l'apprentissage de représentations* pour modéliser les relations entre les utilisateurs. L'apprentissage de représentations a été récemment appliqué à des domaines variés tel que les modèles de langue [Mikolov et al., 2013b], la prédiction de playlists [Chen et al., 2012], la traduction automatique [Graves et al., 2013] ou encore la reconnaissance vocale [Cho et al., 2014]. Le principe est de projeter des éléments (mots, utilisateurs, items) dans un espace de représentation $\mathbb{R}^d$ de dimension fixée, de façon à ce que les distances entre ces éléments dans l'espace de représentation modélisent certaines relations entre eux.

Dans notre cas, nous projetons les utilisateurs dans un espace latent de façon à ce que les distances entre eux représentent l'influence qu'ils ont les uns sur les autres, leur propension à se transmettre de l'information, leurs similarités ou leurs liens d'amitiés. L'utilisation de cette approche nous permet d'obtenir des modèles moins complexes que les modèles de diffusion basés sur le graphe du réseau social, et de régulariser les relations entre utilisateurs. En particulier, deux utilisateurs proches (deux amis par exemple) seront naturellement proches des mêmes *autres* utilisateurs, modélisant le principe : « les amis de mes amis sont mes amis ».

1.4 Tâches

Dans cette section, nous définissons de façon informelle les tâches étudiées dans ce manuscrit. Une définition plus précise de chacune d'entre elles sera donnée dans les chapitres correspondants.

1.4.0.1 Prédiction de diffusion

Le but de cette tâche est de prédire, étant donné un ou plusieurs utilisateurs initialement infectés par une information (ou *sources*), quels seront les utilisateurs infectés dans le futur, à la fin de la diffusion. Notons bien qu'il s'agit ici d'une tâche *prédictive*, et non pas *explicative* : le but est uniquement de retrouver *quels* utilisateurs seront infectés, par *par qui* ou *comment*. Un modèle adapté à cette tâche peut ensuite être utilisé pour de la prédiction de buzz ou de la maximisation d'influence [Kempe et al., 2003].

1.4.0.2 Détection de source

Cette tâche est l'inverse de la précédente. Le but est de retrouver la source d'une information à partir de l'ensemble des utilisateurs finalement infectés par celle-ci. Suivant le contexte, un tel modèle peut servir à retrouver la source d'une fausse rumeur, l'origine d'une fuite ou le point départ d'un virus informatique au sein de réseau.

1.5 Contributions

Dans cette section, nous décrivons brièvement les différentes contributions réalisées au cours de cette thèse et présentées dans ce manuscrit.

1.5.0.1 Apprentissage atemporel du modèle IC (chapitre 3)

> Lamprier, S., Bourigault, S., and Gallinari, P. (2015). Extracting diffusion channels from real-world social data: A delay-agnostic learning of transmission probabilities. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM

Dans le modèle IC, chaque paire d'utilisateurs (u_i, u_j) d'un réseau social est associée à une probabilité de transmission $p_{i,j}$. Les premiers articles étudiant ce modèle considéraient des probabilités de transmission fixées *a priori*. Toutefois, dans la plupart des applications, celles-ci doivent être *apprises* à partir d'un ensemble d'exemples d'apprentissage. Cet ensemble d'apprentissage prend généralement la forme d'un ensemble de séquences d'utilisateurs infectés correspondant chacune à une information se diffusant dans le réseau. Par exemple, sur Youtube, chaque séquence contiendra l'ensemble des utilisateurs ayant visionné une vidéo donnée. Le but est est alors de trouver les probabilités de transmission maximisant la vraisemblance de cet ensemble d'apprentissage.

La difficulté vient du fait que ces exemples nous indiquent seulement *quand* un utilisateur a été infecté par une information, et pas *par qui*. Un algorithme d'apprentissage de type Espérance-Maximisation a été proposé par [Saito et al., 2008]. Malheureusement, celui-ci

fait l'hypothèse qu'un utilisateur ne peut avoir été infecté que par un voisin infecté au *pas de temps* précédent.

Dans notre première contribution [Lamprier et al., 2015], nous considérons pour notre part que cette hypothèse de discrétisation du temps est trop forte, et nous proposons une version relaxée de cette approche en considérant qu'un utilisateur peut avoir été infecté par n'importe quel autre utilisateur précédemment infecté. Nous adaptons l'algorithme en conséquence. De plus, nous remarquons que la parcimonie des données d'apprentissage peut conduire à un problème de sur-apprentissage, que nous proposons de limiter en introduisant un mécanisme de régularisation. Nous comparons notre méthode d'apprentissage à celle de [Saito et al., 2008] sur des données réelles et observons un gain important sur la tâche de prédiction de diffusion.

1.5.0.2 Apprentissage de représentations pour le modèle IC (chapitre 5)

> Bourigault, S., Lamprier, S., and Gallinari, P. (2016b). Representation learning for information diffusion through social networks: An embedded cascade model. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 573–582, New York, NY, USA. ACM

Dans notre première contribution, nous proposions une méthode permettant d'apprendre des probabilités de transmission entre chaque paire d'utilisateurs d'une population fixée. Ces probabilités étaient apprises de façon *libre*, c'est sans aucune contrainte a priori sur leurs valeurs relatives. En pratique, on sait que les graphes de réseaux sociaux présentent de nombreuses propriétés particulières : faible diamètre, distribution des degrés en loi de puissance, structures de communautés... De plus, apprendre une probabilité de transmission pour chaque paire d'utilisateur pose un problème de complexité.

Pour résoudre ces deux problèmes, nous proposons dans notre deuxième contribution [Bourigault et al., 2016b] d'apprendre des représentations latentes des utilisateurs, dans un espace euclidien $\mathbb{R}^d$. Ces représentations ont pour but de modéliser les comportements, interactions et similarités des utilisateurs, et sont apprises de façon à ce que chaque probabilité de transmission $p_{i,j}$ du modèle IC puissent se déduire de la distance séparant les représentations des deux utilisateurs correspondants. Nous adaptons l'algorithme d'apprentissage à cette formulation. Nous observons une amélioration des performances en prédiction de diffusion par rapport à notre première contribution.

1.5.0.3 Modélisation par Diffusion de Chaleur (chapitre 6)

> Bourigault, S., Lagnier, C., Lamprier, S., Denoyer, L., and Gallinari, P. (2014). Learning social network embeddings for predicting information diffusion. In

Proceedings of the 7th ACM International Conference on Web Search and Data Mining, WSDM '14, pages 393–402, New York, NY, USA. ACM

Nos deux contributions précédentes étaient des modèles *explicatifs* : nous proposions d'apprendre des probabilités de transmission d'information que nous utilisions ensuite pour *simuler* le processus de diffusion au sein d'un graphe reliant les utilisateurs.

Dans notre troisième contribution [Bourigault et al., 2014], nous n'utilisons pas de modèle explicatif : les utilisateurs sont projetés dans un espace de représentation $\mathbb{R}^d$ de façon à ce que la diffusion d'information au sein de cette population puisse être vue comme un processus de diffusion de chaleur dans cet espace. La diffusion est donc modélisée de façon *continue*, et non plus de façon itérative comme dans les deux contributions précédentes. Le modèle obtenu est beaucoup plus rapide en inférence.

Nous proposons également une extension de ce modèle permettant de prendre en compte le contenu de l'information se diffusant. Cette extension améliore les résultats de façon importante.

1.5.0.4 Détection de source (chapitre 7)

Bourigault, S., Lamprier, S., and Gallinari, P. (2016a). Learning distributed representations of users for source detection in online social networks. In *Proceedings of the 2016 European conference on Machine Learning and Knowledge Discovery in Databases*, ECML PKDD'16. Springer-Verlag

Enfin, dans notre quatrième contribution [Bourigault et al., 2016a], nous étudions la tâche de détection de source. Là encore, notre approche consiste à projeter les utilisateurs dans un espace latent. Nous proposons d'utiliser les positions des utilisateurs infectés par une information donnée pour calculer une représentation de cette information dans l'espace, et utilisons celle-ci pour retrouver l'utilisateur source.

Nous comparons cette méthode à diverses approches graphiques issues de l'état de l'art, et montrons qu'elle nous permet d'obtenir de meilleurs résultats dans différents contextes applicatifs tout en étant bien moins complexe à utiliser en inférence.

Nous présentons également deux extensions permettant de prendre en compte le contenu de l'information se diffusant et d'apprendre l'importance de chaque utilisateur dans la détection.

Remarquons que les différentes contributions ne sont pas présentées, au sein de ce manuscrit, dans l'ordre chronologique de leurs publications. Cette organisation est plus logique, puisqu'elle nous permet de décrire d'abord des méthodes d'apprentissage pour le modèle IC (deux premières contributions) avant de proposer un modèle de diffusion continu plus rapide, puis de terminer en étudiant la tâche annexe de détection de source.

Première partie

Modélisation de la diffusion

Dans cette première partie, nous présentons un état de l'art sur la modélisation et la prédiction de la diffusion d'information. En particulier, nous présentons le modèle *Independent Cascades* (IC), qui est un modèle de diffusion classique, étudié et étendu dans de nombreux travaux.

Nous présentons ensuite une première contribution : une méthode d'apprentissage du modèle IC permettant d'ignorer les délais de diffusion, qui nous permet d'obtenir de meilleurs résultats qu'un modèle IC appris de façon classique.

Chapitre 2

Modélisation et prédiction de la diffusion d'information : état de l'art

Résumé Ce chapitre présente un état de l'art sur la prédiction de diffusion dans les réseaux sociaux. Nous étudions plusieurs approches correspondant à des contextes variés et à des tâches différentes. Nous nous intéressons également à quelques problématiques annexes liées à la prédiction de diffusion : la maximisation d'influence, l'identification de leaders d'opinion et la détection de source.

2.1 Introduction

Dans ce chapitre, nous présentons un état de l'art sur le sujet de la diffusion d'information dans les réseaux sociaux et sur plusieurs tâches associés. Nous commençons par présenter plusieurs modèles de diffusion d'informations. Ceux-ci peuvent être séparés en deux groupes, que nous nommerons « modèles à *faible* granularité » et « modèles à *forte* granularité ».

- Un *modèle à faible granularité* est une approche s'intéressant à une propriété globale de la diffusion la diffusion, comme par exemple le nombre d'utilisateurs infectés par une information donnée. Ce type de modèle est généralement basé sur des propriétés globales du réseau social étudié : taille du réseau, connectivité, distribution des degrés, etc.
- Un *modèle à forte granularité* s'intéresse pour sa part à la diffusion à l'échelle des utilisateurs. Ce type de modèle vise en général à simuler la diffusion au sein d'un graphe fixé.

Après avoir présenté ces travaux sur la modélisation de la diffusion d'information, nous nous intéressons à plusieurs problématiques liées.

- La *prédiction de diffusion* consiste à prédire le résultat de la diffusion d'une information à partir d'un état initial.

— La *maximisation d'influence* vise à trouver quels utilisateurs initialement infectés permettent de maximiser l'ampleur de la diffusion d'information.
— L'*identification des leaders d'opinion* cherche à retrouver les utilisateurs les plus influents d'un réseau social.
— Enfin, la *détection de source* désigne la tâche inverse de la prédiction de diffusion : retrouver, à partir de l'état final d'une diffusion, l'utilisateur dont elle est partie.

2.2 Généralités sur la diffusion

Avant d'aborder l'état de l'art, nous présentons de façon succincte le phénomène de diffusion d'information. Nous en profitons également pour définir quelques notations et termes utilisés tout au long de ce manuscrit.

2.2.1 Réseau Social

Soit $U = \{u_1, u_2, \ldots, u_N\}$ une *population* de N *utilisateurs* faisant partie d'un *réseau social*. Au sein de ce réseau social, ces utilisateurs sont reliés par un ensemble de liens E, qui constituent le *graphe social* $G = (U, E)$. Ces liens peuvent être orientés ou non, selon le réseau social considéré : les relations entre utilisateurs sont par exemple symétriques sur Facebook (liens d'amitié) mais asymétriques sur Twitter (abonnements). Nous notons Preds_i et Succs_i les prédécesseurs et successeurs de u_i dans G. La nature exacte des utilisateurs peut également dépendre du type de contexte étudié : il pourra s'agir de personnes physiques, de blogs ou sites web, par exemple.

2.2.2 Diffusion

Au cours du temps, diverses *informations* se diffusent au sein de la population U, principalement par le biais du *bouche-à-oreille*. Une information peut prendre beaucoup de formes : une vidéo ou article de blog partagés sur Facebook, un message sur Twitter retweeté par beaucoup d'utilisateurs, un comportement particulier adopté progressivement par la population (l'achat d'un produit à la mode, par exemple), etc...

Les utilisateurs atteints par une information donnée sont dit *infectés*, ou *contaminés*. Ces termes viennent du fait que plusieurs modèles issus de l'épidémiologie ont été appliqués à la diffusion d'information. Dans certains cas, on parlera aussi d'utilisateur *activé* ou ayant *adopté* une information (termes issus du marketing).

Lorsqu'une information se propage dans la population U, nous observons en général les temps d'infection des différents utilisateurs concernés. Nous nommons *épisode de diffusion*

D la séquence d'utilisateurs infectés, avec leurs temps d'infection associés :

$$D = \big((u_i, t_i^D), (u_j, t_j^D), (u_k, t_k^D), \dots\big)$$

L'exposant D des temps d'infection t_i^D sera parfois omis dans ce manuscrit lorsque le contexte ne laissera aucune ambiguïté. Dans la littérature, D est aussi nommé *séquence d'activation* ou *trace de diffusion*. Un épisode de diffusion peut correspondre à l'ensemble des « likes » recueillis par un article sur Facebook, où à l'ensemble des vues d'une vidéo sur Youtube. Il est important de noter que D indique seulement *qui* a été infecté par une information donnée et *quand*, mais pas *comment* ou *par qui*. Cette information manquante sera souvent source de difficultés. Nous considérons, sans perte de généralité, que le premier utilisateur est infecté à $t = 0$, et que les temps d'infection des utilisateurs suivants sont donc indiqués de façon relative à celui du premier utilisateur. De plus, nous considérons que les utilisateurs non infectés dans D vérifient $t_i^D = +\infty$

Nous notons U_t^D l'ensemble des utilisateurs infectés dans D avant le temps t, i.e :

$$U_t^D = \{u_i \in U | t_i^D < t\}$$

En particulier, U_∞^D désignera l'ensemble des utilisateurs infectés dans D, et sera parfois abrégé en U^D. Nous utiliserons également la notations $\bar{U}_t^D = U \setminus U_t^D$.

La notion d'épisode de diffusion définie ici est suffisamment abstraite pour s'appliquer à de nombreux contextes expérimentaux. Strictement parlant, n'importe quel corpus composé de triplets de la forme (*utilisateur*,*item*,*temps*) permet de définir des épisodes de diffusion. Suivant l'origine des données utilisées et la façon dont elles ont été extraites, la sémantique associée au concept d'*épisode de diffusion* sera donc variable, et celle associée au modèle appliqué aux données le sera donc aussi. De plus, dans certains cas, la définition des triplets n'est pas triviale. En particulier, retrouver dans un grand flux de messages (comme Twitter) les différents « sujets » discutés constitue déjà une tâche difficile, nommée *Topic Detection and Tracking (TDT)* dans la littérature. Ces considérations vont toutefois au delà du sujet étudié dans ce manuscrit. Comme dans la plupart des travaux sur la diffusion d'information, nous extrairons les épisodes de diffusion de façon relativement simple.

2.2.3 Graphe de Diffusion

Lorsque l'information « qui a contaminé qui » est connue, l'ensemble des utilisateurs infectés peut être représenté par un graphe. Ce graphe sera nommé *graphe de diffusion* ou, s'il est orienté, *cascade*. Dans la plupart des modèles, il s'agira d'un sous-graphe de G. De plus, beaucoup d'articles font l'hypothèse qu'un utilisateur infecté l'a été par un seul autre utilisateur. Dans ce cas, le graphe de diffusion sera un *arbre de diffusion*. Un exemple est donné en figure 2.1.

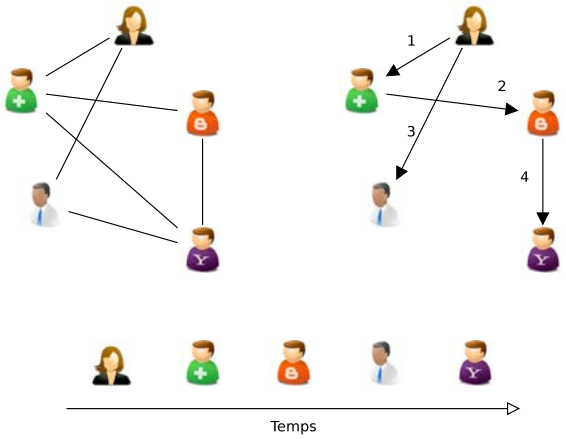


FIGURE 2.1 – Exemple de graphe de diffusion. À gauche, le graphe d'un réseau social. À droite, un graphe de diffusion représentant la diffusion d'une information. Les numéros indiquent dans quelle ordre les différentes contaminations ont lieu. En bas, l'épisode de diffusion (séquence d'utilisateurs infectés) correspondant.

2.3 Modèles de diffusion à faible granularité

Les modèles de diffusion à faible granularité visent à modéliser l'évolution, au cours du temps, d'une grandeur caractérisant la diffusion d'information. Le plus souvent, cette grandeur sera le nombre ou le pourcentage d'utilisateurs infectés.

2.3.1 Le modèle de Bass

Le premier modèle mathématique décrivant le phénomène du bouche-à-oreille a été proposé par Bass [Bass, 1969] pour expliquer la façon dont les consommateurs adoptent un produit donné. Dans ce modèle, un individu peut adopter un produit suite à l'influence soit de la publicité, soit d'autres personnes ayant déjà adopté le produit en question (on dira qu'il y a eu *contamination*).

La probabilité à tout instant pour un individu d'adopter le produit par le biais de la publicité est notée p, et la probabilité de contamination d'une personne par une autre est notée q. En notant $i(t)$ le *ratio* de consommateurs ayant adopté le produit au sein de la population au temps t (de façon cumulative), Bass propose l'équation différentielle :

$$\frac{\partial i}{\partial t}(t) = p \times (1 - i(t)) + q \times (i(t) \times (1 - i(t)))$$

Le premier terme de l'équation correspond à l'adoption du produit par le biais de la publicité, et le deuxième terme à l'effet du bouche-à-oreille.

La solution de cette équation, avec la condition initiale $i(0) = 0$, est :

$$i(t) = \frac{1 - e^{-(p+q).t}}{1 + \frac{q}{p}e^{-(p+q).t}}$$

Les valeurs de p et de q ont été mesurées à $p = 0.03$ et $q = 0.38$ en moyenne sur plusieurs centaines de courbes d'adoptions observées sur de vraies campagnes marketing [Mahajan et al., 1995], un résultat montrant l'importance de la diffusion inter-utilisateurs.

2.3.1.1 Extensions

Le modèle de Bass peut être appliqué à la diffusion d'information sur les réseaux sociaux, soit dans sa forme de base [Luu et al., 2012], soit dans une version étendue prenant en compte davantage de paramètres.

En particulier, le modèle se base sur la loi d'action de masse, c'est à dire que chaque utilisateur est susceptible d'influencer tous les autres. Dans [Luu et al., 2012], les auteurs observent qu'en pratique, la diffusion ne se fait pas dans un graphe complet, et que l'effet du bouche-à-oreille dépend donc de la distribution des degrés dans le graphe social. Ils proposent les extensions *Scale-free network Linear Influence Model* (SLIM) et *Exponential network Linear Influence Model* (ELIM), modélisant la diffusion dans une population où la répartition des degrés dans le graphe social suit une loi de puissance ou une loi exponentielle, respectivement. L'équation devient :

$$\frac{\partial i}{\partial t}(t) = p(1 - i(t)) + q\left(E_d(i(t))\right)$$

où $E_d(i(t))$ désigne le nombre moyen de voisins ayant adopté le produit pour les utilisateurs non infectés. Sa valeur dépend de la répartition des degrés et de $i(t)$.

Dans [Lerman and Hogg, 2010], les auteurs s'intéressent au site internet *Digg* et intègrent dans le modèle de Bass plusieurs paramètres très spécifiques à l'organisation de ce site, ce qui permet une modélisation plus précise : nombre moyens de visites du site, répartition des informations sur différentes pages, etc.

2.3.2 Modèles épidémiologiques

Des modèles suivant la même idée ont également été proposés pour modéliser et expliquer la diffusion d'une maladie au sein d'une population [Daley et al., 2001]. Dans ces modèles, chaque utilisateur se trouve dans un *état*. A chaque instant, il peut changer d'état, suivant des probabilités dépendant du modèle.

Le premier modèle de ce type a avoir été proposé est le *Susceptible-Infected-Recovered* (SIR) [Kermack and McKendrick, 1927] dans lequel chaque utilisateur peut se trouver dans un état parmi trois à chaque instant :

— *susceptible* : susceptible d'être contaminé par la maladie étudiée ;
— *infected* : infecté par cette maladie ;
— *recovered* : guéri, et immunisé contre cette maladie.

Les nombres d'utilisateurs susceptibles, infectés et guéris au temps t sont notés $S(t)$, $I(t)$ et $R(t)$, avec :

$$\forall t : S(t) + I(t) + R(t) = N$$

A chaque instant, chaque utilisateur infecté a une probabilité p de contaminer chaque utilisateur susceptible, et chaque utilisateur infecté a une probabilité r de guérir. L'évolution du système est donc régie par les équations suivantes :

$$\begin{cases} \frac{\partial S}{\partial t} = -p.SI \\ \frac{\partial I}{\partial t} = p.SI - r.I \\ \frac{\partial R}{\partial t} = r.I \end{cases}$$

Le calcul d'une solution exacte est complexe [Harko et al., 2014], mais plusieurs méthodes d'approximation existent [Keeling and Rohani, 2008, Harko et al., 2014].

De la même façon qu'avec le modèle de Bass, quelques travaux ont montré que ce modèle expliquait bien certains types d'épidémies. Des exemples concernant une épidémie de grippe et une épidémie de peste sont donnés dans [Brauer et al., 2001]. Ce modèle a également été utilisé pour étudier la diffusion d'informations sur des forums en ligne [Woo et al., 2011]. Le modèle SIR permet de calculer diverses propriétés épidémiologiques. En particulier, il est possible de montrer que si $\frac{p.S(0)}{r} > 1$, une épidémie a lieu : la valeur de I augmente jusqu'à un maximum, puis diminue jusqu'à 0. Sinon, la valeur de I diminue directement.

De nombreuses variations et extensions de ce modèle ont pu être proposées [Daley et al., 2001]. Par exemple, dans la cas d'une maladie dont il est impossible de guérir, la valeur de r est fixée à 0, ce qui conduit au modèle SI. Si la guérison de donne pas d'immunité, alors chaque utilisateur infecté aura une certaine probabilité de repasser en état *susceptible*, ce qui conduit au modèle SIS avec les équations :

$$\begin{cases} \frac{\partial S}{\partial t} = -p.SI + r.I \\ \frac{\partial I}{\partial t} = p.SI - r.I \end{cases}$$

Si un utilisateur guéri peut perdre son immunité avec une certaine probabilité s, il est possible de définir le modèle SIRS régi par les équations suivantes :

$$\begin{cases} \frac{\partial S}{\partial t} = -p.SI + s.R \\ \frac{\partial I}{\partial t} = p.SI - r.I \\ \frac{\partial R}{\partial t} = r.I - s.R \end{cases}$$

Toujours selon le même principe, il est également possible de considérer une maladie potentiellement mortelle, de prendre en compte des naissances ou des décès au sein de la population, de modéliser la transmission de l'immunité de la mère à l'enfant ou la vaccination d'une partie de la population, etc. [Brauer et al., 2001].

2.3.3 Apprentissage des influences globales des utilisateurs

La quantité de données rendues disponibles par le développement des réseaux sociaux a rendu possible la mise en place de modélisations basées sur des propriétés plus précises des utilisateurs, plutôt que sur des métadonnées définies à l'échelle du réseau.

Par exemple, le *Linear Influence Model* (LIM) a été proposé dans [Yang and Leskovec, 2010]. Dans cet article, les auteurs s'intéressent à l'évolution du nombre d'infections (ou « volume d'infection ») $I(t)$. Contrairement aux approches précédentes, cette modélisation est basée sur l'observation d'une sous-population $O \subset U$. Chaque utilisateur $u \in O$ est associé à une fonction d'influence I_u de façon à ce que $I_u(t)$ corresponde au nombre d'infections provoquées par u après un temps t. En notant t_i le temps d'infection de chaque utilisateur $u_i \in O$, le volume de diffusion I s'écrit alors :

$$I(t) = \sum_{\substack{u_i \in O \\ t_i < t}} I_{u_i}(t - t_i)$$

Autrement dit, le nombre total d'infections est égal à la somme des influences des utilisateurs de O, décalées en fonction de leurs temps d'infections. Les auteurs proposent une méthode simple pour apprendre les fonctions $I_u(t)$. Ils considèrent le cas où le temps s'écoule de façon discontinue (par pas de temps successifs), et apprennent chaque valeur de $I_u(t)$ directement, pour $t < T_{\max}$. Le problème d'apprentissage s'écrit alors comme un problème de minimisation pouvant être résolu efficacement.

2.4 Modèles de diffusion à forte granularité

Les modèles de diffusion à forte granularité sont des modèles basés sur le graphe du réseau social et visant à simuler, à l'échelle des utilisateurs, la diffusion d'information au sein de celui-ci. Nous en présentons plusieurs dans cette section, plus ou moins complexes

et reposant sur différentes hypothèses quant aux mécanismes régissant le diffusion d'information.

Nous présentons ici des modèles utilisés dans le cadre le diffusion d'information, mais ce type de modèle est généralement susceptible de modéliser de nombreux autres types de diffusion [Granovetter, 1978], où le comportement de chaque utilisateur est influencé par ceux des autres : adoption d'un nouveau produit, propagation de rumeurs ou de maladies, diffusion de décisions (se mettre en grève par exemple), etc. À l'origine, les deux premiers modèles présentés ici (IC et LT) étaient utilisés en sociologie pour expliquer certains comportement observés à l'échelle d'une population comme le résultat d'actions et d'interactions ayant lieu à l'échelle des individus.

2.4.1 Le modèle *Independent Cascades* (IC)

Le modèle « Independent Cascades » présenté ici a été défini par [Kempe et al., 2003] pour étudier le problème de la maximisation d'influence dans le cadre du marketing viral (dont nous parlons plus loin dans ce chapitre). Ce modèle a ensuite été étudié dans [Saito et al., 2008], et c'est celui que nous étudions dans plusieurs chapitres de ce manuscrit. Historiquement, le modèle IC défini par [Kempe et al., 2003] se base sur les travaux de [Goldenberg et al., 2001] et de [Granovetter, 1973], qui utilisaient des modèles similaires pour étudier l'impact du phénomène de bouche-à-oreille dans des campagnes publicitaires.

Dans le modèle IC, chaque lien (u_i, u_j) du graphe est associé à une probabilité de transmission $p_{i,j}$. La diffusion se déroule de façon itérative.

- À l'instant initial $t = 0$, un ensemble d'utilisateurs $U_I \subset U$ devient infecté.
- Lorsqu'un utilisateur devient infecté à un pas de temps t, il dispose d'une *unique* chance de contaminer chacun de ses successeurs $u_j \in \text{Succs}_i$ selon la probabilité $p_{i,j}$. Si la contamination a lieu, u_j devient infecté au temps $t + 1$. Ainsi, un utilisateur n'est contagieux que pendant un seul pas de temps.
- La diffusion se poursuit tant que de nouveaux utilisateurs deviennent infectés.

Un exemple de diffusion est donné en figure 2.2.

Le modèle IC est donc *centré sur l'émetteur* : ce sont les actions des émetteurs de contenu (transmission/non-transmission) qui sont modélisées. De plus, ce modèle fait une hypothèse d'indépendance : la probabilité qu'une transmission ait lieu sur un lien (u_i, u_j) particulier est toujours la même, et ne dépend pas des précédentes tentatives d'infections sur u_j.

Dans [Kempe et al., 2003], les probabilités de transmission sont considérées connues, et les auteurs s'intéressent uniquement au problème de la maximisation d'influence. Dans [Goldenberg et al., 2001], les auteurs utilisent un modèle assez proche, et considèrent que chaque lien peut être un lien *fort* (associé à une probabilité $p_{i,j} = p_F$) ou un lien *faible* (associé à une probabilité $p_{i,j} = p_f < p_F$). Ils étudient l'impact des liens faibles

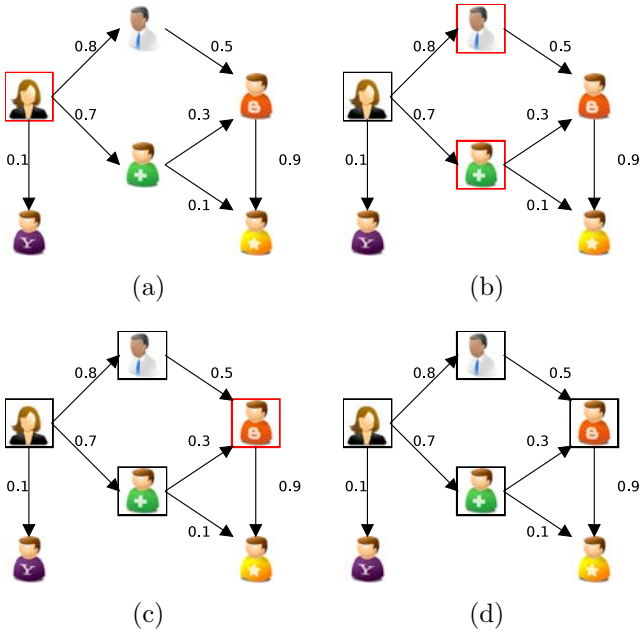


FIGURE 2.2 – Exemple de diffusion selon le modèle IC. Quatre itérations sont ici représentées. Les utilisateurs encadrés sont ceux ayant été contaminés, ceux encadrés en rouge sont les « nouveaux infectés », i.e. ceux infectés à l'itération courante et qui tentent donc d'infecter chaque voisin.

sur la diffusion d'un produit par bouche-à-oreille en simulant l'évolution du système pour différentes valeurs de p_f et p_F.

En pratique, les probabilités de transmission ne sont pas connues. Pour appliquer ce modèle à un réseau social réel, il est donc nécessaire d'*inférer* les valeurs de ces probabilités à partir d'un ensemble d'épisodes de diffusion.

Ce problème d'apprentissage a été étudié dans [Gruhl et al., 2004], qui proposait un algorithme de type EM considérant que chaque utilisateur infecté avait été contaminé par *exactement un* utilisateur. Une approche plus précise a ensuite été proposée par [Saito et al., 2008]. L'algorithme de cet article considère que chaque utilisateur infecté a été contaminé par *au moins un* autre utilisateur, ce qui correspondant mieux au modèle IC. La principale difficulté de cet apprentissage provient du fait qu'un utilisateur infecté dans un épisode de diffusion peut avoir été contaminé par n'importe lequel de ses voisins, et qu'il n'est pas possible de savoir lequel avec certitude. Par exemple, dans la figure 2.2, nous pouvons voir qu'à la troisième itération, lorsque l'utilisateur orange devient infecté, il peut avoir été contaminé soit par l'utilisateur blanc, soit par l'utilisateur vert. Si cette information était connue, l'estimation des probabilités de transmission serait triviale : il suffirait de diviser le nombre d'infections réussies par le nombre de tentatives d'infection de u_j par u_i pour estimer $p_{i,j}$. Ce n'est pas le cas, et l'apprentissage des probabilités devient donc un problème d'estimation à partir d'informations incomplètes, pour lequel

[Saito et al., 2008] propose un algorithme de type espérance-maximisation (EM). Nous reviendrons plus en détail sur cet algorithme dans le chapitre 3.

2.4.2 Le modèle *Linear Threshold* (LT)

Le modèle « linear threshold » présenté ici est également issu de [Kempe et al., 2003]. Une version plus simple de ce modèle avait d'abord été étudiée en sociologie par [Granovetter, 1978] pour analyser les effets de seuil dans le comportement des groupes. De nombreuses versions de ce modèle appliquées à diverses problématiques avait ensuite été proposées ([Kempe et al., 2003] cite une dizaines de travaux en exemple).

Dans le modèle LT de [Kempe et al., 2003], chaque lien (u_i, u_j) est associé à une valeur $w_{i,j}$ représentant l'influence de u_i sur u_j, avec la contrainte :

$$\forall u_j \in U : \sum_{u_i \in \text{Preds}_j} w_{i,j} \leq 1$$

A chaque fois qu'une information se diffuse dans G, chaque utilisateur u_j tire un seuil d'influence $s_j \in [0, 1]$ uniformément. Comme pour le modèle IC, la diffusion est simulée de façon itérative. À chaque pas de temps $t > 0$, chaque utilisateur u_j non-infecté le devient si et seulement si :

$$\sum_{\text{Preds}_j \cap U_t} w_{i,j} \geq s_j$$

où U_t désigne l'ensemble des utilisateurs infectés avant t. La diffusion continue tant que de nouveaux utilisateurs deviennent infectés. Un exemple de diffusion en donné en figure 2.3.

À l'inverse du modèle IC, le modèle LT est *centré sur le récepteur* : c'est le comportement du récepteur u_j qui est modélisé avec le seuil s_j. De plus, ce modèle fait une hypothèse *d'additivité de l'influence* : la propension d'un utilisateur à s'infecter croît avec son nombre de voisins infectés.

Comme dans le cas du modèle IC, [Kempe et al., 2003] considère que les poids sont connus et étudie uniquement le problème de la maximisation d'influence.

L'apprentissage des paramètres du modèle LT à partir d'un ensemble d'épisodes de diffusion observés a été étudié dans [Goyal et al., 2010]. Les poids $w_{i,j}$ y sont estimés en utilisant différentes heuristiques relativement simples, basées sur le nombre d'épisodes de diffusion auxquels les utilisateurs de chaque couple (u_i, u_j) participent. Les modèles LT ainsi appris sont testés sur des données réelles issues de Flikr [1], au moyen de courbes ROC, et obtiennent de bonnes performances.

1. `https://www.flickr.com/`

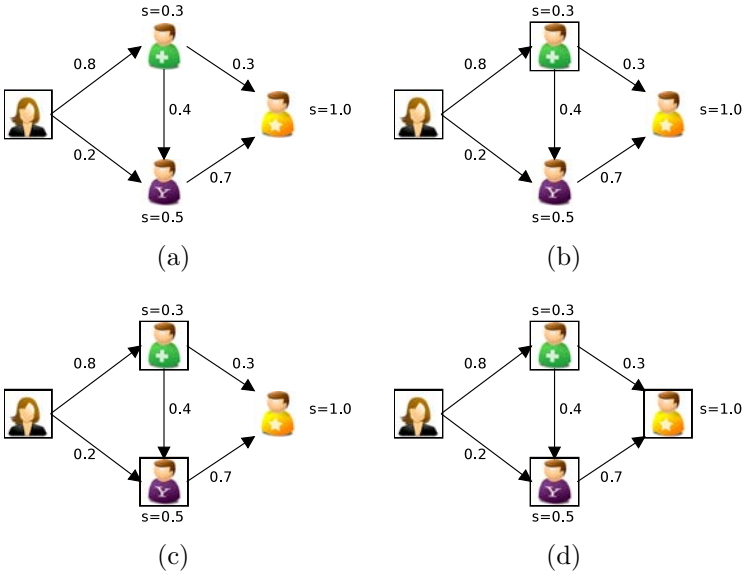


FIGURE 2.3 – Exemple de diffusion selon le modèle LT. Quatre itérations sont ici représentées. Un utilisateur devient infecté à partir du moment où la somme des poids des liens provenant d'utilisateurs infectés dépasse son seuil d'activation.

Une méthode plus précise pour apprendre les paramètres du modèle LT a été proposée dans [Vaswani and Duttachoudhury,]. Elle consiste à maximiser la vraisemblance des paramètres $W = (w_{i,j})_{(u_i,u_j)\in E}$ par rapport à un ensemble d'épisodes de diffusion D. Cette vraisemblance se calcule assez simplement en remarquant que la probabilité qu'un utilisateur u_j devienne infecté au temps t est égale à la probabilité que le seuil s_j tiré soit inférieur à la somme des poids des voisins infectés à ce moment, soit :

$$\begin{aligned} P_i^D(t) &= P\left(s_j \leq \sum_{u_i \in \text{Preds}_j \cap U_t^D} w_{i,j}\right) \\ &= \sum_{u_i \in \text{Preds}_j \cap U_t^D} w_{i,j} \end{aligned}$$

Le calcul ne fait donc pas intervenir les seuils tirés, uniquement les poids des relations. La vraisemblance vaut alors :

$$\mathcal{L}(W;\mathcal{D}) = \prod_{D\in\mathcal{D}} \prod_{t=0}^{T_{\max}-1} \left(\prod_{u_i \in (U_{t+1}^D \setminus U_t^D)} \left(P_i^D(t+1)\right) \prod_{u_i \in \bar{U}_{t+1}^D} \left(1 - P_i^D(t+1)\right) \right)$$

Cette vraisemblance est optimisée par [Vaswani and Duttachoudhury,] avec plusieurs techniques (gradient, espérance - maximisation, point intérieur). Cette méthode d'apprentissage est évaluée sur des épisodes de diffusion synthétiques générés avec le modèle LT. L'évaluation se fait en comparant les valeurs de W apprises aux vraies valeurs, et

sur des tâches de prédiction de diffusion. Les résultats indiquent que l'algorithme d'apprentissage parvient bien à retrouver les vraies valeurs de W. En revanche, la méthode d'apprentissage n'est pas testée sur des données réelles.

Comparaison : Les modèles IC et LT ont principalement été comparés dans le cadre des tâches de maximisation d'influence et d'identification de leaders d'opinion. En particulier, [Fushimi et al., 2008] étudie le comportement des deux modèles sur la tâche d'identification de leaders d'opinion, avec des graphes issus de réseaux sociaux en ligne. L'influence $I(u_i)$ d'un utilisateur y est définie comme l'espérance du nombre d'utilisateurs infectés à l'issue d'une diffusion commençant par cet utilisateur u_i et suivant un modèle diffusion donné. Ils observent que l'influence d'un utilisateur est d'avantage corrélée à son degré sortant avec le modèle LT qu'avec le modèle IC. Ils montrent également que la présence de communautés d'utilisateurs dans les graphes a un impact bien plus important sur les influences des utilisateurs au sens du modèle IC.

2.4.3 Généralisation de IC et LT

L'article [Kempe et al., 2003] a également proposé des versions « généralisées » des modèles IC et LT, permettant de lever l'hypothèse d'indépendance du modèle IC et celle d'additivité du modèle LT. Il est ensuite possible de montrer que toute instance du modèle IC généralisé est équivalente à une certaine instance du modèle LT généralisé, et vice-versa.

2.4.3.1 IC généralisé

Au lieu de considérer des probabilités de transmission $p_{i,j}$ constantes, le modèle IC généralisé considère que la transmission se fait avec une probabilité $p_i(j, X)$, où X est l'ensemble des utilisateurs ayant déjà tenté de contaminer u_j mais ayant échoué. L'hypothèse d'indépendance du modèle IC disparaît : la contamination de u_j par u_i peut dépendre des précédentes tentatives d'infections sur u_j. Typiquement, $p_i(j, X)$ pourra être croissante ou décroissante suivant $|X|$, pour modéliser différents comportements possibles. Le modèle IC normal correspond au cas où $p_i(j, X) = p_{i,j}$.

2.4.3.2 LT généralisé

Dans le modèle LT généralisé, la condition d'infection $\sum_{\text{Preds}_j \cap U_t} w_{i,j} \geq s_j$ est remplacée par une forme plus générale : $g_j(\text{Preds}_j \cap U_t) \geq s_j$, où $g_j : 2^U \to [0, 1]$ est une fonction

monotone avec $g_j(\emptyset) = 0$ associant à un ensemble d'utilisateurs [2] l'influence qu'ils exercent sur u_j. Le modèle LT normal correspond au cas où $g_j(X) = \sum_{u_i \in X} w_{i,j}$.

2.4.4 Modèles continus

Les modèles LT et IC sont des modèles itératifs dans lesquels le temps est discrétisé et la diffusion à lieu de façon synchrone, à chaque pas de temps. En pratique, l'information se propage de façon continue, et à des vitesses variables. Des versions continues des modèles IC et LT ont donc été proposées.

2.4.4.1 CTIC et CTLT

Une version continue du modèle IC, nommée *Continuous Time IC* (CTIC) a été définie dans [Saito et al., 2009].

- Dans cette version, chaque lien (u_i, u_j) du graphe est associé à un paramètre temporel $r_{i,j}$, en plus de sa probabilité de transmission $p_{i,j}$.
- Quand l'utilisateur u_i devient infecté au temps t_i, il tente de contaminer chaque successeur u_j, comme dans le modèle IC.
- Si cette contamination réussi, l'utilisateur u_j devient infecté au temps $t_i + d_{i,j}$, où $d_{i,j}$ est un délai tiré selon une loi exponentielle de paramètre $r_{i,j}$.

$$P(d_{i,j} = x) = \begin{cases} r_{i,j} e^{-x r_{i,j}} & \text{si } x >= 0 \\ 0 & \text{sinon} \end{cases}$$

Ainsi, la diffusion suivant ce lien prend en moyenne un temps $d_{i,j} = \frac{1}{r_{i,j}}$.

Le modèle IC classique peut être vu comme un cas particulier de CTIC dans lequel les délais de transmission valent toujours 1.

L'apprentissage des paramètres $p_{i,j}$ et $r_{i,j}$ de cette version a également été étudié par Saito dans [Saito et al., 2009]. Il propose un algorithme espérance-maximisation similaire à [Saito et al., 2008], prenant en compte le fait qu'un utilisateur observé dans une séquence d'activation peut avoir été contaminé par n'importe quel prédécesseur infecté avant lui, et pas uniquement par ceux infectés au pas de temps précédent.

Une extension similaire du modèle LT, CTLT, a été proposée dans [Saito et al., 2010b]. Cette fois-ci, chaque utilisateur u_i est associé à un paramètre temporel r_i. Lorsque l'influence exercée sur u_i par ses voisins au temps t dépasse son seuil d'activation s_i, u_i ne devient infecté qu'au temps $t + d_i$, après un délai tiré de la même façon que dans le modèle CTLT, selon une loi exponentielle de paramètre r_i.

2. Nous utiliserons dans ce manuscrit la notation 2^X pour désigner l'ensemble des parties d'un ensemble X. L'utilisation de la notation $\mathcal{P}(X)$ serait en effet source de confusion, car nous seront également amenés à manipuler des probabilités $P(X)$ dans divers contextes.

Notons que dans certains travaux, ces deux modèles sont également nommés AsIC et AsLT (pour *asynchronous* IC ou LT).

Comparaison : Les modèles CTIC et CTLT ont été comparés dans [Saito et al., 2010a, Saito et al., 2010c], avec des données réelles issues du site *Doblog*[3], dont sont extraits plusieurs ensembles d'apprentissages correspondant à des sujets différents. Seul le cas où les paramètres des modèles sont uniformes sur l'ensemble du réseau est étudié dans cet article. Les auteurs observent que le modèle CTIC explique mieux la diffusion de la plupart des types d'information présents dans le corpus. Seuls certains sujets précis, correspondant généralement à des épisodes de diffusion plus longs, sont mieux expliqués par un modèle CTLT.

2.4.4.2 Modèle continu de Leskovec et Gomez-Rodrigez : NetRate

Un modèle proche de CTIC a également été utilisé par Leskovec et Gomez-Rodriguez [Gomez-Rodriguez et al., 2011]. Celui-ci reprend l'idée de CTIC mais ne définit pas de probabilités de transmission $p_{i,j}$, uniquement un paramètre temporel $r_{i,j}$ sur chaque lien. Lorsqu'un utilisateur u_i devient infecté, il transmet l'information à chaque successeur u_j après un délai $d_{i,j}$, tiré selon une distribution de probabilité paramétrée par $r_{i,j}$ Cette distribution est généralement une loi exponentielle ou une loi de puissance [Gomez-Rodriguez et al., 2011], mais d'autres formes sont possibles [Farajtabar et al., 2015]. Avec une loi exponentielle, ce modèle devient équivalent à un modèle CTIC où toutes les probabilités de transmissions seraient égales à 1. Une faible propension à transmettre de l'information sera représentée dans ce modèle par une valeur $r_{i,j}$ très faible, au lieu d'une valeur de $p_{i,j}$ très faible.

Dans [Gomez Rodriguez et al., 2010], les auteurs considèrent que $r_{i,j}$ est toujours égal à une constante sur tous les liens du graphe. Dans [Gomez-Rodriguez et al., 2011], ils proposent une méthode pour apprendre les valeurs de $r_{i,j}$. Ne pas utiliser de probabilités de transmission $p_{i,j}$ leur permet notamment d'utiliser des méthodes d'optimisation continue plus simples qu'un algorithme EM. Ce modèle a ensuite été utilisé pour des tâches d'inférence de graphe (voir section 2.4.7) ou de détection de source (voir section 2.8).

2.4.5 Intégration du contenu et des attributs utilisateurs.

Dans les modèles vus jusqu'à présents, aucune distinction n'est faite a priori entre les différents utilisateurs et les différentes informations se diffusant. En pratique, le type d'information se diffusant ainsi que les profils des utilisateurs sont susceptibles d'avoir un

3. `http://www.doblog.com`

impact non-négligeable sur la diffusion. Par exemple, une information concernant la politique internationale ne se diffusera pas de la même façon et auprès des mêmes utilisateurs qu'une information concernant un résultat sportif. Divers travaux ont donc proposé des techniques permettant d'intégrer cette information dans des modèles similaires à IC ou LT. Ces techniques consistent en général à dériver les paramètres d'un modèle IC à partir des propriétés des utilisateurs ou du contenu de l'information. Dans cette section, τ_i désignera le profil de l'utilisateur u_i, et w_D le contenu de l'information se diffusant dans l'épisode D. La nature exacte de ces profils et du contenu dépendent du réseau social étudié.

2.4.5.1 Intégration des attributs utilisateurs

Dans [Saito et al., 2011], les auteurs considèrent pour chaque lien (u_i, u_j) du graphe le vecteur $x_{i,j}$ de même taille que les profils τ_i et τ_j, et dont chaque composante $x^a_{i,j}$ est définie par :

$$x^a_{i,j} = e^{|\tau^a_i - \tau^a_j|}$$

Ce vecteur est ensuite utilisé pour définir les valeurs des paramètres d'un modèle CTIC $p_{i,j}$ et $r_{i,j}$:

$$p_{i,j} = \frac{1}{1 + e^{-<\theta, x_{i,j}>}}$$

$$r_{i,j} = e^{-<\phi, x_{i,j}>}$$

où θ et ϕ sont deux vecteurs de paramètres de même taille que les vecteurs d'attributs. Les auteurs proposent un algorithme EM pour apprendre les valeurs de ces paramètres. Cependant, le modèle n'est testé que sur des données synthétiques, avec des attributs utilisateurs générés de façon artificielle sur des graphes issus données réelles.

Une idée similaire a été proposée par [Guille and Hacid, 2012] et testée sur des données réelles. Pour chaque liens (u_i, u_j), les auteurs définissent une douzaine de propriétés liées aux actions et interactions de ces deux utilisateurs. Diverses méthodes d'apprentissage automatique sont ensuite utilisées pour prédire, à partir de ces propriétés, si un contenu donné va se diffuser de l'utilisateur u_i à l'utilisateur u_j. Les auteurs évaluent leur approche sur des données réelles issues de Twitter et constatent entre autres que la propriété la plus importante pour prédire la diffusion de u_i vers u_j est leur nombre de voisins communs dans G.

2.4.5.2 Intégration du contenu et des attributs utilisateurs

Dans [Lagnier et al., 2013], les auteurs définissent un modèle centré sur le récepteur (comme le modèle LT) basé sur trois propriétés calculées au cours de la diffusion :

— la *similarité* entre le profil τ_i de l'utilisateur et le contenu diffusé w_D ;

— l'*activité* de u_i, c'est à dire sa propension générale à devenir infecté (calculée sur les épisodes d'apprentissage) ;
— l'*influence* de ses voisins sur cet utilisateur, i.e. le nombre de voisins infectés.

Ces propriétés permettent de calculer une probabilité d'infection de l'utilisateur u_i en appliquant une fonction logistique dont les paramètres sont appris sur les épisodes d'apprentissage. Ce modèle est testé sur des épisodes de diffusion réels, sur une tâche de prédiction de diffusion, et obtient de meilleurs résultats que les modèles classiques.

2.4.5.3 Intégration du contenu seul

Enfin, des versions des modèles IC et LT prenant en compte le contenu de l'information se diffusant ont été proposées dans [Barbieri et al., 2013b], TIC et TLT (pour *Topic-aware* IC et LT). Dans ces modèles, le contenu d'une information se propageant est représenté par une distribution de probabilité sur Q topics, i.e $w_D \in [0,1]^Q$ avec

$$\sum_{q=0}^{Q-1} w_D^q = 1$$

Ce vecteur peut notamment être obtenu au moyen d'une LDA (*Latent Dirichlet Allocation*).

Dans le modèle TIC, chaque lien (u_i, u_j) est associé à un ensemble de Q probabilités de transmissions $(p_{i,j}^q)_{q=0..Q-1}$. Quand une information de contenu w_D se propage, la probabilité de transmission d'un utilisateur u_i à u_j vaut alors :

$$p_{i,j}^D = \sum_{q=0}^{Q-1} w_D^q . p_{i,j}^q$$

De la même façon, dans le modèle TLT, chaque lien est associé à un ensemble de poids $(w_{i,j}^q)_{q=0..Q-1}$, et l'influence d'un utilisateur u_i sur un utilisateur u_j pour une information de contenu w_D est :

$$w_{i,j}^D = \sum_{q=0}^{Q-1} w_D^q . w_{i,j}^q$$

Des algorithmes de type EM sont proposés pour apprendre les paramètres de ces modèles. Ils obtiennent de meilleurs résultats que les versions « sans contenu » sur des corpus réels.

2.4.6 Modélisation de plusieurs diffusions

Tous les articles présentés jusqu'ici modélisent la diffusion d'une seule information à la fois. Plusieurs informations se diffusant en parallèle sont donc considérées de façons indépendantes. Quelques travaux se sont toutefois intéressés au cas où plusieurs diffusions simultanées pouvaient avoir une influence les unes sur les autres.

En particulier, [Myers and Leskovec, 2012] considère que la probabilité qu'un utilisateur u_i de Twitter exposé à une séquence d'informations $(I_1, I_2, I_3...)$ décide de *retweeter* l'information I_x dépend des k informations précédentes suivant la formule :

$$P_i(I_x|I_{x-1}, \ldots, I_{x-N}) = P_i(I_x) + \sum_{y=x-N}^{x-1} f(I_x, I_y) + \gamma_i$$

où $P_i(I_x)$ est la probabilité a priori que l'utilisateur u_i retweete l'information I_x, $f(I_x, I_y)$ correspond à l'influence (positive ou négative) de l'exposition à l'information I_y sur la probabilité de retweeter I_x, et γ_i est un biais propre à chaque utilisateur. L'influence de chaque information sur chaque autre (la fonction f) est calculée en partitionnant les informations en *clusters*, et en apprenant les influences de chaque cluster sur chaque autre. Les paramètres de ce modèle (infection a priori, biais et influences inter-clusters) sont appris en maximisant la vraisemblance d'un ensemble d'apprentissage d'environ 18000 diffusions simultanées. Les expériences effectuées montrent que l'inclusion de l'influence entres les différentes informations augmente les performances en prédiction de plus de 200%.

La diffusion de plusieurs informations a également été étudiée dans le cadre de la maximisation d'influence (voir section 2.6.3.1)

2.4.7 Inférence de graphe

Il arrive dans certaines applications que le graphe du réseau social au sein duquel l'information se diffuse soit caché ou inconnu. Pour appliquer un modèle de diffusion de type IC ou LT dans une telle situation, il est alors nécessaire d'*inférer* ce graphe à partir des épisodes de diffusion observés.

L'inférence de graphe consiste à prédire les liens existant entre les éléments d'un ensemble fixé. Dans le contexte de la diffusion d'information, le but est de retrouver le graphe social $G = (U, E)$ à partir d'un ensemble d'épisodes de diffusion observés $\mathcal{D}$ sur la population U. Nous présentons dans cette sous-section quelques travaux sur le sujet. Le principe général est toujours le même : faire l'hypothèse d'un certain modèle de diffusion, puis rechercher les liens les plus vraisemblables par rapport à $\mathcal{D}$ avec le modèle de diffusion en question.

2.4.7.1 NetInf

Dans ce premier article [Gomez Rodriguez et al., 2010], les auteurs se basent sur l'intuition suivante : plus un utilisateur u_j a tendance à être infecté peu de temps après un utilisateur u_i, plus l'existence d'un lien (u_i, u_j) est vraisemblable.

Les auteurs utilisent le modèle CTIC, en faisant l'hypothèse que tous les liens partagent la même probabilité de transmission $p_{i,j} = p$ et le même paramètre temporel $r_{i,j} = r$. Ils cherchent alors l'ensemble de k liens $\hat{E}$ de vraisemblance maximum par rapport à l'ensemble d'épisodes de diffusion observés $\mathcal{D}$ sous le modèle CTIC :

$$\hat{E} = \underset{|E| \leq k}{\arg\max} \sum_{D \in \mathcal{D}} \log P(D|E)$$

Limiter la taille de l'ensemble de liens du graphe à k est obligatoire, sinon une solution triviale consisterait à retrouver le graphe complet. Le problème d'optimisation est donc *combinatoire*, et les auteurs démontrent qu'il est NP-complet. Ils observent cependant que la fonction à optimiser $f(E) = \sum_{D \in \mathcal{D}} \log P(D|E)$ est sous-modulaire, et qu'il est donc possible d'obtenir une bonne solution en utilisant un algorithme glouton : partir de l'ensemble vide, et ajouter à chaque itération le lien maximisant le gain marginal (voir section 2.6.1).

2.4.7.2 NetRate

Dans [Gomez-Rodriguez et al., 2011], les auteur proposent d'utiliser le modèle continu présenté dans la section précédente pour l'inférence de lien (NetRate). Ils considèrent un graphe complet reliant tous les utilisateurs de U, et apprennent les paramètres temporels $\mathcal{R} = (r_{i,j})_{(u_i,u_j) \in U^2}$ sur tous les liens de ce graphe, à partir d'un ensemble d'épisodes de diffusion $\mathcal{D}$, en considérant le problème d'optimisation suivant :

$$\begin{cases} \text{minimiser}_{\mathcal{R}} & -\sum_{D \in \mathcal{D}} \log P(D|\mathcal{R}) \\ \text{s.c.} & \forall (u_i, u_j), r_{i,j} \geq 0 \end{cases}$$

Les liens prédits sont ceux dont le paramètre $r_{i,j}$ est strictement supérieur à 0. Ce modèle est testé sur plusieurs jeux de données réels et artificiels, et les auteurs observent une amélioration des performances par rapport à NetInf. Notons qu'il est possible de réduire considérablement la complexité de l'apprentissage en supprimant a priori tous les liens (u_i, u_j) pour lesquels il n'existe aucun exemple potentiel de diffusion dans $\mathcal{D}$, i.e aucun épisode D dans lequel les deux utilisateur sont infecté, avec u_i infecté avant u_j.

Cette approche fut ensuite améliorée dans [Daneshmand et al., 2014], où les auteurs étudient l'impact du nombre de cascades observées sur la qualité de la prédiction, et proposent d'ajouter une régularisation $\ell 1$ sur les valeurs de $\mathcal{R}$ pour améliorer la stabilité

du modèle. Une version améliorée de NetRate utilisant un noyau a aussi été proposée dans [Du et al., 2012]. Plus tard, [Gomez-Rodriguez et al., 2013] ont proposé une version plus générale où la probabilité d'infection d'un utilisateur est une fonction des temps d'infections de tous les utilisateurs précédents dans l'épisode de diffusion.

2.4.7.3 InfoPath

Enfin, une version dynamique du problème a été étudiée dans [Gomez Rodriguez et al., 2013]. En effet, un réseau social peut évoluer au cours du temps : des utilisateurs peuvent ajouter des contacts pour créer de nouveaux liens, ou en supprimer certains autres. De plus, un lien entre deux utilisateurs peut changer d'intensité : deux personnes peuvent par exemple rester amies sur Facebook mais se perdre de vue et ne plus échanger beaucoup d'information.

L'algorithme InfoPath est une extension de celui de NetRate. Au lieu d'observer un seul ensemble d'épisodes d'apprentissage $\mathcal{D}$, les auteurs observent une *séquence* d'ensemble d'épisodes $(\mathcal{D}^0, \mathcal{D}^1, \mathcal{D}^2 \dots)$. Suivant le contexte applicatif, ces ensembles peuvent être observés au rythme d'un par jour ou d'un par semaine, par exemple. Le but est d'étudier, à chaque fois qu'un nouvel ensemble est observé, l'évolution des paramètres de diffusion.

À chaque pas de temps T, les auteurs résolvent le problème de prédiction de liens selon NetRate sur l'ensemble des cascades observées jusqu'à présent $\mathcal{D}^0 \cup \mathcal{D}^1 \cup \dots \cup \mathcal{D}^T$, en ajoutant une pondération $w(t)$ pour donner plus d'importance aux cascades récentes :

$$\begin{cases} \text{minimiser}_{\mathcal{R}^T} & -\sum_{D \in \mathcal{D}^{t \leq T}} w(t) \log P(D|\mathcal{R}^T) \\ \text{s.c.} & \forall (u_i, u_j), r_{i,j}^T \geq 0 \end{cases}$$

Ce problème est optimisé, à chaque pas de temps, au moyen d'une descente de gradient stochastique. De plus, les valeurs de $\mathcal{R}^T$ sont initialisées avec celles de $\mathcal{R}^{T-1}$. De cette façon, les liens prédits évoluent progressivement avec le temps.

Cet algorithme est d'abord testé sur des données entièrement synthétiques, générées en suivant le modèle utilisé en prédiction, et obtient de bonnes performances. Malheureusement, les auteurs ne disposent pas d'un jeu de données étiquetées correspondant à ce contexte expérimental. Ils évaluent donc leur modèle de façon empirique, en observant l'évolution du graphe prédit au cours du temps.

2.5 Prédiction de diffusion

Dans ce manuscrit, nous nous intéressons principalement à la tâche de *prédiction de diffusion*. Le but est de *prédire* quels utilisateurs seront infectés par une information au

bout d'un certain temps $T_{\max}$ en connaissant uniquement l'état du réseau à un temps initial T_{init}, sans forcément expliquer *comment* ils le seront. Typiquement, T_{init} correspond à un temps de l'ordre de quelques heures (après la début de la diffusion) et $T_{\max}$ à un temps de l'ordre de quelques jours. En d'autres termes, il s'agit de prédire $U^D_{T_{\max}}$ à partir de $U^D_{T_{\text{init}}}$ et d'un ensemble r d'informations complémentaires (telles que le contenu de l'information se diffusant), en définissant une fonction de prédiction f telle que :

$$U^D_{T_{\max}} = f(U^D_{T_{\text{init}}}, r)$$

Plusieurs mesures ont été proposées pour l'évaluation des performances d'une telle fonction. Par exemple, [Najar et al., 2012] et [Lagnier et al., 2013] utilisent des mesures de précision ou de rappel. De leur coté, [Saito et al., 2010b] utilisent une divergence de Kullback-Leibler entre les probabilités finales d'infection prédites et celles observées afin d'évaluer la prédiction réalisée. Dans [Kondor and Lafferty, 2002], les auteurs mesurent un taux d'erreur. Enfin, [Barbieri et al., 2013b] visualise des courbes « taux de faux positifs - taux de vrais positifs ».

Toutes ces mesures traduisent des objectifs différents, et sont susceptibles de ne pas favoriser les mêmes modèles, comme nous le montrons dans le chapitre 5.

2.5.1 Application de modèles de diffusion dans le graphe

Les modèles de diffusion à forte granularité présentés en section 2.4 peuvent être utilisés dans un cadre prédictif : l'état du réseau à $T_{\max}$ peut être obtenu en simulant la diffusion à partir de son état à T_{init}.

2.5.1.1 Simulation du modèle IC

Ces modèles explicatifs, et IC en particulier, étant généralement des processus stochastiques, il est nécessaire d'utiliser une estimation de type « Monte-Carlo ». Cette méthode consiste simplement, étant donné l'ensemble $U^D_{T_{\text{init}}}$, à simuler plusieurs fois le processus de diffusion selon le modèle de diffusion considéré, et à compter le nombre d'instances dans lesquelles chaque utilisateur u_i est infecté pour en déduire la probabilité $P(u_i \in U^D_{T_{\max}} | U^D_{T_{\text{init}}})$. Dans le cas du modèle IC, cette procédure est équivalente à une percolation de liens [Bóta et al., 2013] :

- pour chaque lien (u_i, u_j) du graphe, conserver ce lien avec une probabilité $p_{i,j}$ ou le supprimer avec une probabilité $1 - p_{i,j}$;
- trouver dans le sous-graphe ainsi obtenu tous les utilisateurs pouvant être atteints depuis les utilisateurs initialement infectés.

Ce processus est équivalent à une simulation du modèle IC, et peut être répété pour obtenir la même estimation qu'avec la méthode précédente..

2.5.1.2 Heuristiques basées sur la distance dans le graphe

D'autres méthodes exploitent le fait que les réseaux sociaux sont des graphes parcimonieux et que les probabilités de transmission sont souvent assez faibles [Bóta et al., 2013], ce qui limite la « portée » de la diffusion. Cette observation permet d'appliquer des heuristiques basées sur la proximité entre les utilisateurs.

Par exemple, une méthode d'approximation pour le modèle IC a été proposée dans [Kimura and Saito, 2006]. Celle-ci est basée sur le calcul des plus courts chemins dans le graphe. En effet, en considérant que la longueur de chaque lien (u_i, u_j) est égale à $\log(1 - p_{i,j})$, les longueurs des chemins deviennent inversement proportionnelles à leurs probabilités. Considérer uniquement le plus court chemin d'un utilisateur initial u_x à un autre utilisateur u_y revient donc à approximer la probabilité d'infection finale de u_y avec la probabilité du chemin le plus *vraisemblable*.

Une autre méthode de ce genre a également été utilisée dans [Chen et al., 2010, Chen et al., 2009] pour le modèle LT, et a été adaptée au modèle IC dans [Bóta et al., 2013]. Elle consiste également à considérer uniquement les chemins les plus courts, c'est à dire les plus *vraisemblables*, en calculant la probabilité d'infection finale de chaque utilisateur uniquement partir de son voisinage dans le graphe, plutôt que sur le graphe entier.

2.5.1.3 Méthodes à noyaux

Noyau de modèles de diffusion Au lieu d'utiliser directement un modèle de diffusion à forte granularité, [Rosenfeld et al., 2016] proposent d'appliquer une méthode à noyau. La méthode à noyau proposée est appliquée à la prédiction du *nombre* d'utilisateurs infectés à $T_{\max}$ à partir de la liste des utilisateurs infectés à T_{init}, mais peut parfaitement être appliquée à la prédiction de l'état final de chaque utilisateur. Le but des auteurs est de prédire le nombre d'utilisateurs infectés à la fin d'une diffusion partant d'un ensemble U_I d'utilisateurs-sources, sous l'hypothèse d'un certain modèle de diffusion connu, en particulier IC ou LT. Cette valeur est notée $f(U_I, \theta) = \mathbb{E}[y|U_I, \theta]$, où $\mathbb{E}[y|U_I, \theta]$ est l'espérance du nombre y d'utilisateurs infectés à partir de U_I dans le modèle de diffusion considéré, et θ est l'ensemble des paramètres de ce modèle. Par exemple, si le modèle de diffusion est le modèle IC, θ sera l'ensemble des probabilités de transmission.

Cet ensemble θ doit être appris à partir d'un ensemble d'exemples prenant la forme de X couples (U_I^x, y^x). L'apprentissage des paramètres s'écrit alors :

$$\min_{\theta} \frac{1}{X} \sum_{(U_I^x, y^x)} (f(U_I^x, \theta) - y^x)^2$$

Cet apprentissage est complexe, mais les auteurs montrent qu'il est possible d'utiliser un noyau K, ce qui permet de réaliser l'optimisation sur un espace plus large contenant toutes

les fonctions $f(.,\theta)$ possibles :

$$\min_{\alpha} \frac{1}{X} \sum_{(U_I^x, y^x)} (g(U_I^x, \alpha) - y^x)^2$$

avec :

$$g(U_I, \alpha) = \sum_{U_I^x} \alpha_x K(U_I, U_I^x)$$

La définition du noyau K utilisé dépend du modèle de diffusion. Plusieurs noyaux sont décrits, dont ceux des modèles IC et LT. Cette approche est testée sur des réseaux artificiels et réels, et obtient de meilleurs résultats que diverses méthodes utilisant directement le graphe pour simuler la diffusion.

Noyau de diffusion définis sur les graphes Certains types de noyaux définis sur les graphes peuvent aussi être appliqués à la diffusion d'information. En effet, la diffusion d'information sur un graphe (au sens du modèle IC) peut être vue comme une série de marches aléatoires partant d'une source fixée (ou de plusieurs). Ce processus peut être représenté par un *noyau de diffusion* ou *noyau de chaleur* [Kondor and Lafferty, 2002] défini sur le graphe.

L'utilisation d'un tel noyau consiste à représenter l'infection d'un utilisateur u_i au temps t par une valeur continue $x_i(t)$, interprétée comme une quantité de chaleur. À chaque pas de temps, chaque utilisateur « chaud » (i.e. infecté) transmet une partie de sa chaleur à ses voisins plus « froids » (i.e non-infectés). En notant $x(t)$ le vecteur composé de l'ensemble des valeurs $x_i(t)$ pour $u_i \in U$, l'évolution du système suit :

$$x(t+1) - x(t) = \beta H x(t)$$

où β est un hyperparamètre et H est l'opposé de la matrice laplacienne du graphe :

$$H_{i,j} = \begin{cases} -\text{degré}(u_i) & \text{si } i = j \\ 1 & \text{si } (u_i, u_j) \in E \\ 0 & \text{sinon} \end{cases}$$

Il est alors possible de calculer une solution analytique :

$$x(t) = e^{t\beta H} x(0)$$

L'expression $e^{t\beta H}$ peut être développée en

$$e^{t\beta H} = I + t\beta H + \frac{t^2\beta^2}{2!} H^2 + \frac{t^3\beta^3}{3!} H^3 + ...$$

Intuitivement, $e^{t\beta H}$ est donc une matrice représentant le résultat moyen de plusieurs étapes de diffusion dans le graphe. Une fois cette expression posée, dans le cas où seule la valeur finale à $T_{\max}$ nous intéresse, l'estimation s'écrit donc :

$$x(T_{\max}) = Kx(T_{\text{init}}) \tag{2.1}$$

avec :

$$K = e^{(T_{\max} - T_{\text{init}})\beta H} \tag{2.2}$$

Le calcul exact du *noyau de chaleur* K nécessite de diagonaliser H :

$$H = T^{-1}DT$$
$$e^{\beta H} = T^{-1}e^{\beta D}T$$

La diagonalisation étant une opération complexe, [Kondor and Lafferty, 2002] étudie le cas de graphes particuliers sur lesquels des formules plus simples existent.

Ce noyau correspond à un modèle de diffusion simple, où chaque utilisateur transmet une portion β de sa chaleur à chacun de ses voisins. Plusieurs extensions, permettant de rajouter des poids sur les liens, de les orienter ou de rajouter des biais sur les utilisateurs ont été proposées dans [Ma et al., 2008]. Chaque extension conduit à une définition différente du noyau K.

2.5.1.4 Limites des approches basées sur la diffusion dans le graphe

L'utilisation de ces approches basées sur le graphe pose toutefois problème. En effet, celles-ci font implicitement l'hypothèse que l'information ne peut se diffuser *que* sur les liens de ce graphe. Cette hypothèse est en pratique assez forte [Yang and Leskovec, 2010], pour plusieurs raisons.

Tout d'abord, la multiplication des services en ligne fait que les utilisateurs font souvent partie de *plusieurs* réseaux sociaux parallèles. L'information devient ainsi susceptible de suivre des « chemins détournés », et de passer d'un utilisateur à un autre en suivant des liens ne faisant pas partie de l'unique réseau considéré.

De la même façon, la diffusion au sein du graphe n'est pas toujours le seul facteur expliquant l'infection des utilisateurs. Divers éléments propres au fonctionnement du service étudié peuvent avoir un impact sur cette diffusion. Sur Twitter par exemple, un utilisateur peut recevoir de l'information par le biais des personnes auxquelles il s'est abonné, mais aussi par le biais de la liste des « trending topics », qui regroupe l'ensemble des informations les plus populaires du moment. Il est toutefois possible, dans certains cas, de modéliser cette « influence extérieure » un ajoutant dans le graphe un utilisateur virtuel u_0, relié à tous les utilisateurs et considéré comme toujours infecté [Gomez Rodriguez et al., 2010].

D'autre part, le graphe explicite renseigné sur un réseau social n'est pas toujours le plus pertinent pour expliquer la diffusion [Huberman et al., 2008]. Ainsi, dans [Cha et al., 2010], nous apprenons que le nombre de « followers » d'un utilisateur donne assez peu d'information sur la nature de son influence dans Twitter, contrairement à son nombre de « retweets » et de « mentions ». Les auteurs observent ainsi que les comptes de journaux ou de chaînes d'informations génèrent généralement beaucoup de retweets, alors que les comptes de célébrités sont plus rarement retweetés mais génèrent plus de « mentions ». De plus, il a été montré que sur Twittter, les liens faibles (i.e. les liens reliant des utilisateurs ayant peu de connaissances communes) jouaient un rôle important dans la diffusion d'information [Zhao et al., 2010, Granovetter, 1973]. Utiliser directement le graphe explicite peut revenir à ignorer cette hétérogénéité dans la nature même de ces liens.

Enfin, il est possible que le graphe du réseau social soit inconnu, totalement ou partiellement. Cela peut être dû à différentes raisons. La liste des amis ou des contacts est parfois une information privée. Sur Facebook par exemple, un utilisateur peut décider de masquer cette liste. Sur Twitter, l'API offerte aux développeurs limite le nombre de requêtes possibles chaque heure, ce qui rend impossible l'extraction du graphe complet. L'effet de ces restrictions a d'ailleurs été étudié dans [Morstatter et al., 2013].

Tous ces éléments sont susceptibles de limiter la pertinence d'un modèle de diffusion à forte granularité. D'autres méthodes de prédiction ont donc été proposées.

2.5.2 Régression directe

Dans [Najar et al., 2012], les auteurs prédisent le vecteur $x(T_{\max})$ représentant l'état des utilisateurs du réseau au temps $T_{\max}$ à partir de $x(T_{\text{nit}})$ en utilisant l'apprentissage automatique. Différents modèles, en particulier une régression linéaire et une régression logistique, sont utilisés pour réaliser la prédiction, leurs paramètres étant appris par descente de gradient. Il est intéressant de remarquer que la prédiction avec un classifieur linéaire s'écrit :

$$x(T_{\max}) = \theta.x(T_{\text{init}})$$

où $\theta \in \mathbb{R}^{N \times N}$ la matrice des paramètres. On retrouve la même forme que l'équation 2.1. Cette approche revient donc en quelque sorte à « apprendre » un noyau de chaleur au lieu de le calculer à partir d'un graphe.

Les modèles appris sont évalués sur des épisodes de diffusion artificiels avec des mesures issues de la recherche d'information, et comparés aux modèles IC et LT. Les auteurs observent que leur modèle obtient des performances similaires aux modèles IC et LT lorsque le graphe est connu, et des performances bien meilleures que celles des modèles explicatifs lorsque le graphe utilisé pour générer les données n'est que partiellement connu. Cette robustesse est un avantage important des approches prédictives par rapport aux approches explicatives.

2.5.3 Recommandation

Le problèmes de recommandation a été très largement étudié ces dernières années, en particulier depuis la création du *Netflix Challenge*[4]. Le but de la recommandation est de *suggérer* à un utilisateur un ensemble d'*items* susceptibles de l'intéresser, au vu de son activité passée. Ce type de suggestion est par exemple visible sur Amazon : un utilisateur ayant acheté un smartphone se verra par la suite recommander divers accessoires pour celui-ci.

Les problèmes de recommandation et de prédiction de diffusion (à forte granularité) peuvent être vues comme deux facettes d'un même cadre plus large : associer des *utilisateurs* à des *items*.

— Dans le cadre de la recommandation, il s'agit principalement, étant donné un utilisateur, de trouver à quels items le relier pour les lui recommander.
— Dans le cadre de la diffusion, le but est inversé : à partir d'un item, l'objectif est de trouver quels utilisateurs vont être infectés.

Pour autant, les deux problématiques ne sont pas équivalentes : les systèmes de recommandation ne se placent pas dans un contexte séquentiel la plupart du temps et ne s'évaluent pas de la même façon que les modèles de diffusion. De plus, la prédiction de diffusion considère en général l'*influence* existant entre les utilisateurs, là où la recommandation s'intéresse aux *similarités* entres eux. Cependant, ces deux aspects (influence et similarité) peuvent être délicats à distinguer [Aral et al., 2009], et il apparaît donc que les deux problématiques peuvent se recouper [Zhang et al., 2007].

Dans le cadre de la prédiction de diffusion, il peut donc être pertinent de s'intéresser aux méthodes utilisées en recommandation.

Les méthodes les plus répandues aujourd'hui en recommandation sont celles dites de *filtrage collaboratif*, consistant à observer des similarités dans les comportements des utilisateurs et à prédire quels produits recommander à l'un d'eux en utilisant les comportements d'autres utilisateurs similaires. Citons notamment les travaux de [Koren et al., 2009], utilisant la factorisation matricielle. Le principe est le suivant : nous observons partiellement une matrice $M \in [0,5]^{N \times \text{nbItems}}$. Chaque ligne correspond à un utilisateur et chaque colonne à un produit. Chaque case de la matrice correspond à une note laissée par un utilisateur à un produit (traditionnellement entre 0 et 5 étoiles). Certaines cases de M ne sont pas observées, le but étant de prédire leurs valeurs. Pour cela, la factorisation matricielle consiste à observer que la matrice M « complétée » peut se factoriser ainsi :

$$M \approx R_U \times R_I$$

avec :

$$R_U \in \mathbb{R}^{N \times d}$$

4. `http://www.netflixprize.com`

$$R_I \in \mathbb{R}^{d \times \text{nbItems}}$$

Cette factorisation peut être obtenue en minimisant le coût :

$$\mathcal{L}(R_U, R_I) = \sum_{(i,j) \in \text{observées}} ||M^{i,j} - R_U^{i,\cdot}.R_I^{\cdot,j}||^2 + \lambda \left(||R_U^{i,\cdot}||^2 + ||R_I^{\cdot,j}||^2 \right)$$

La somme est calculée uniquement sur les composantes connues de la matrice M. Le second terme est un terme de régularisation. La matrice R_U peut ainsi être vue comme une projection des utilisateurs dans un espace à d dimensions (une ligne par utilisateur) et la matrice R_I comme une projection des produits dans le même espace (un produit par colonne). Une composante manquante $M^{i,j}$ sera prédite en utilisant $R_U^{i,\cdot}.R_I^{\cdot,j}$, c'est à dire la similarité dans l'espace latent entre l'utilisateur et le produit correspondants. L'avantage de cette formulation est que les utilisateurs ayant donné des notes similaires se verront attribuer des représentations proches. De la même façon, des produits bien notés par les mêmes utilisateurs seront également projetés à des emplacements similaires. Cette propriété permet de prédire de nouvelles relations entre les utilisateurs et les produits. Par exemples, si deux produits ont reçu de bonnes notes de la part des mêmes utilisateurs, les utilisateurs ayant noté un seul de ces deux produits se verront recommander l'autre.

Ce modèle s'étant montré particulièrement efficace, de nombreuses extensions ont été proposées, prenant en compte divers éléments supplémentaires comme les propriétés connues des produits ou celles des utilisateurs. L'utilisation de ce type d'approche dans le contexte d'une population reliée par un réseau social a également été étudiée [Ma et al., 2011, Jamali and Ester, 2010]. En particulier, dans [Jamali and Ester, 2010], les auteurs proposent le modèle SocialMF consistant à intégrer dans la factorisation matricielle un a priori représentant la *confiance* des utilisateurs les uns envers les autres. Cette confiance se *propage* dans le réseau de la même manière que de l'information : si l'utilisateur u_1 fait confiance aux notes données par l'utilisateur u_2 qui lui-même fait confiance à l'utilisateur u_3, alors u_1 fera vraisemblablement confiance à u_3. Dès lors, on peut considérer que la « confiance à u_3 » s'est propagée de u_2 à u_1.

Les liens existant entre les tâches de prédiction de diffusion et de recommandation, ainsi que l'efficacité des méthodes de factorisation matricielle, nous conduirons à utiliser des approches similaires d'*apprentissage de représentations* dans nos travaux (chapitres 5, 6 et 7).

2.5.4 Prédiction de Volume

Notons enfin que la prédiction de diffusion peut aussi se réaliser à *faible granularité*, c'est à dire uniquement en visant à prédire des propriétés générales d'une diffusion d'information, en particulier le volume.

2.5.4.1 Prédiction du taux d'adoption final

Dans beaucoup d'applications, comme la « prédiction de Buzz », le but peut être de prédire le nombre d'utilisateurs infectés à un horizon $T_{\max}$, noté $I(T_{\max})$, en connaissant sa valeur à un instant T_{init} mesurée assez tôt, ainsi que d'autres paramètres observés à T_{init}. Plusieurs approches assez simples pour résoudre ce problème existent

Régression simple La plus classique consiste à utiliser le fait que dans de nombreux cas, l'évolution de I vérifie :

$$\log(I(T_{\max})) \approx \alpha \times \log(I(T_{\text{init}})).$$

La prédiction $I(T_{\max})$ peut donc se faire en mesurant la valeur de α. Cette formule peut être appliquée à divers contextes, comme la popularité d'une vidéo sur Youtube ou le nombres de votes d'un article posté sur Digg [Szabo and Huberman, 2010].

Plus proche voisin Une méthode non-paramétrique a été étudiée dans [Chen et al., 2013]. Cet article vise à prédire les « trending topics » de Twitter. Le modèle proposé est basé sur une approche de type plus proche voisin : pour prédire si une information i va devenir un *trending topic* en observant seulement les premières valeurs de $I(t)$ pour $t < T_{\text{init}}$, ces valeurs sont comparées à celles observées sur un ensemble de séries temporelles d'apprentissage étiquetées en « trending/non-trending ». Si la série temporelle la plus proche de celle de i pour $t < T_{\text{init}}$ correspond à un *trending topic*, la prédiction est que i en sera également un. Le modèle est testé sur des données issues de Twitter. En testant plusieurs valeurs de T_{init}, le modèle parvient à un taux de vrais positifs de 95%, tout en détectant les *trending topics* avant que Twitter ne les désigne comme tels dans 79% des cas (un trending topics est détecté « avant Twitter » lorsque T_{init} est inférieur au temps à partir duquel le sujet est apparu dans la liste des tendances).

2.5.4.2 Intégration du Contenu et attributs utilisateurs

De la même façon que dans la sous-section 2.4.5, il est possible de définir des modèles basés sur le contenu de l'information diffusée ou sur certaines propriétés spécifiques des utilisateurs. Dans la cas de la prédiction du volume de diffusion, les propriétés prises en compte dépendent beaucoup de la nature du réseau social considérés. Pour cette raison, les articles proposant des approches basées sur le contenu sont souvent spécifiques à un site particulier.

Sur Twitter Dans [Tsur and Rappoport, 2012], les auteurs proposent un modèle permettant de prédire la popularité $I(T_{\max})$ d'un hashtag sur Twitter après un certain temps

$T_{\max}$. La fonction de prédiction utilisée est une fonction linéaire dont les paramètres sont appris au moyen d'une descente de gradient stochastique. Cette fonction linéaire est appliquée à une représentation vectorielle du hashtag prenant en compte de nombreuses caractéristiques de celui-ci :

Contenu du Hashtag : longueur, nombre de mots, projection du hashtag sur un dictionnaire, mots fréquemment associés, etc.

Topologie : nombre moyen et maximum de *followers* des utilisateurs ayant utilisé le hashtag avant T_{init}, nombre de retweets, etc.

Temporalité : différentes valeurs du nombre d'utilisateurs infectés $I(t)$ pour $t < T_{\text{init}}$

On remarque bien ici que le modèle est propre à Twitter et pourrait difficilement être utilisé tel quel sur un autre site. Le modèle est testé sur un large corpus de tweets. Il apparaît que les caractéristiques temporelles sont plus informatives que les celles liées au contenu, mais qu'un modèle prenant en compte toutes les caractéristiques décrites est meilleur.

Sur Facebook Un travail semblable a été effectué dans [Cheng et al., 2014] en collaboration avec Facebook, sur des données constituées de photos postées sur Facebook et diffusées par le biais de partages successifs sur une large population d'utilisateurs.

Au lieu de prédire $I(T_{\max})$ en fonction d'observations réalisées à T_{init}, un cadre prédictif plus général est défini. Les auteurs observent en effet que la répartition des tailles finales des épisodes de diffusion suit une loi de puissance d'exposant $\alpha \approx 2$. Ils posent donc comme objectif de prédire, après avoir observé les n premiers utilisateurs ayant partagé une photo donnée, si celle-ci va être partagée au moins $2n$ fois ou non. Il s'agit donc d'un problème de classification binaire. De plus, la valeur de n n'est pas fixée : le modèle doit être applicable à n'importe quel « point » de la diffusion, pour prédire si le nombre d'utilisateurs infectés va doubler ou non.

L'avantage de cette formulation est qu'elle rend le problème de classification *équilibré* : parmi les photos partagées au moins n fois, environ la moitié sera finalement partagée au moins $2n$ fois (loi de puissance d'exposant $\alpha \approx 2$). De la même façon que dans [Tsur and Rappoport, 2012], l'article utilise donc un modèle de classification (une régression logistique) appliqué à un vecteur de représentation calculé à partir des n premiers utilisateurs infectés, basé sur de nombreuses caractéristiques :

Contenu de la photo : Tags, mots utilisés dans la description, propriétés de l'image...

Propriétés de l'utilisateur-source : Age, nombre d'amis, activité sur Facebook...

Propriétés des utilisateurs ayant repartagé la photo : Age moyen, nombre d'amis...

Topologie : Nombre de liens dans le graphe de diffusion, taille du voisinage de ce graphe, profondeur...

Temporalité : temps écoulé depuis la diffusion de la photo, temps moyen entre les infections, etc.

Les premiers résultats obtenus en prédiction sont très proches de ceux de [Tsur and Rappoport, 2012] : la précision est de 79%, les caractéristiques temporelles sont les plus informatives et un classifieur prenant en compte toutes les caractéristiques est meilleur. De plus, les auteurs observent que l'importance des différentes caractéristiques évolue beaucoup avec la valeur de n. En particulier, plus n augmente, moins les caractéristiques de l'utilisateur initial et de la photo sont importantes dans la prédiction.

L'article va plus loin et propose également de prédire la *forme du graphe de diffusion*, représentée par l'indice de Wiener de ce graphe. L'indice de Wiener est défini comme la sommes des longueurs des plus courts chemins entres tous les sommets du graphe. Plus celui-ci est faible, plus le graphe est compact. En adoptant un critère de classification similaire, le modèle parvient également à prédire cet indice.

2.6 Maximisation d'influence

Nous avons vu dans la section précédente quelques méthodes permettant de prédire le volume final de diffusion d'une information. A partir de là, le but de la tâche de « maximisation d'influence » est de trouver comment agir sur un réseau social de façon à maximiser ce volume final. Dans la plupart des cas, *l'action* sur le réseau social consistera à sélectionner l'ensemble des utilisateurs initiaux à partir desquelles l'information se diffusera.

La principale application de ce problème est celle du marketing viral : un annonceur désire par exemple envoyer à un certain nombre d'utilisateurs d'un réseau un exemplaire gratuit d'un produit dont il veut faire la promotion. Son but est alors de trouver quels utilisateurs cibler de façon à déclencher une diffusion la plus large possible.

2.6.1 Modèles IC et LT

La formulation la plus courante du problème a été donnée par Kempe dans [Kempe et al., 2003]. Soit $\sigma : 2^U \to \mathbb{R}$ une fonction associant à un ensemble d'utilisateurs initiaux $U_I \subset U$ l'espérance du nombre d'utilisateurs infectés après une propagation d'information partant de U_I sous l'hypothèse d'un modèle de diffusion connu. Le problème de la maximisation d'influence revient alors à considérer la maximisation sous contraintes suivante :

$$\begin{cases} \max_{U_I} & \sigma(U_I) \\ \text{s.c.} & |U_I| \leq k \end{cases}$$

Dans l'article fondateur du problème de maximisation d'influence[Kempe et al., 2003], les auteurs considèrent le cas d'un modèle de diffusion IC ou LT dont tous les paramètres sont connus (graphe, probabilités de transmission ou poids). Il est montré que dans ce cas, le problème est NP-difficile. Cependant, les auteurs démontrent que la fonction σ est

sous-modulaire dans le cas des modèles de diffusion IC et LT, indépendamment de leurs paramètres, i.e :

$$\forall U'_I, \forall U_I \subseteq U'_I, \forall s \notin U'_I : \sigma(U_I \cup \{s\}) - \sigma(U_I) \geq \sigma(U'_I \cup \{s\}) - \sigma(U'_I)$$

Une propriété importante des fonctions sous-modulaires est qu'il est possible de trouver un ensemble U_I tel que $\sigma(U_I)$ soit une approximation à $\left(1 - \frac{1}{e}\right)$ près de la valeur maximale de σ au moyen d'un algorithme glouton :

- partir de l'ensemble $U_I = \emptyset$;
- ajouter à chaque itération l'utilisateur maximisant le gain marginal : $U_I \leftarrow U_I \cup \{\arg\max_U \sigma(U_I \cup \{u_i\})\}$;
- continuer jusqu'à atteindre $|U_I| = k$.

Cet algorithme glouton est testé sur des graphes réels, avec les modèles IC et LT, et comparé à différentes heuristiques permettant de choisir les utilisateurs initiaux : utilisation des degrés sortants, utilisation de la centralité de distance des utilisateurs ou tirage aléatoire. Chaque méthode est testée pour différentes valeurs de k, puis les modèles IC ou LT sont utilisés pour simuler la diffusion à partir des utilisateurs initiaux sélectionnés par chaque méthode. L'algorithme glouton proposé parvient toujours à infecter plus d'utilisateurs, quelques soient la valeur de k et le modèle de diffusion.

Une extension de ce travail, considérant des modèles plus généraux que IC et LT a ensuite été proposée dans [Kempe et al., 2005].

La principale limite de cette approche est celle du passage à l'échelle : chaque itération de l'algorithme glouton nécessite d'estimer $O(N)$ valeurs de σ, N étant le nombre d'utilisateurs. Cette estimation de σ repose sur une méthode de Monte-Carlo, le calcul exact serait $\#P$-difficile. Plusieurs optimisations ont donc été proposées [Chen et al., 2009, Chen et al., 2010]. En particulier, [Chen et al., 2010] propose également d'optimiser de façon gloutonne une fonction sous-modulaire, mais en basant le calcul de σ sur la recherche des plus courts chemins dans le graphe, qui peuvent être calculés une seule fois au moyen d'un algorithme de Dijkstra. Cette approche obtient des résultats très proches de ceux de [Kempe et al., 2003], tout en étant plus rapide de plusieurs ordres de grandeur.

2.6.2 Version temporelle

Le problème a également été étudié dans le cadre des approches continues de modélisation de la diffusion, comme celles décrites en section 2.4.4. Ainsi, dans [Gomez Rodriguez et al., 2012], les auteurs étudient le cas du modèle NetRate [Gomez-Rodriguez et al., 2011] pour rechercher les utilisateurs permettant de maximiser la diffusion. Rappelons que dans ce modèle de diffusion, chaque utilisateur infecté u_i contamine chacun de ses voisins u_j après un délai $d_{i,j}$ tiré selon une loi exponentielle de paramètre $r_{i,j}$. De la même façon que les articles précédents, les auteurs démontrent que la fonction σ découlant de ce modèle est

sous-modulaire, et proposent un algorithme glouton pour l'optimiser, ainsi que plusieurs techniques permettant d'en accélérer grandement le calcul.

L'algorithme est testé sur des graphes et des épisodes de diffusion artificiels et réels, et obtient de meilleurs résultats que les méthodes de [Kempe et al., 2003] et de [Chen et al., 2010] grâce à la prise en compte de la dimension temporelle, en particulier si l'horizon temporel considéré $T_{\max}$ est faible.

Une autre possibilité a été étudiée par [Ma et al., 2008]. Dans cet article, les auteurs utilisent un noyau de chaleur définie sur un graphe, similaire à l'approche présentée en section 2.5.3. Ce noyau permet aux auteurs de faciliter le calcul de σ, et de proposer plusieurs heuristiques pour sélectionner les utilisateurs initiaux. Ils étudient également le cas où ces utilisateurs initiaux peuvent devenir infectés à des temps différents.

2.6.3 Contextes de diffusion négative

Dans certaines applications, d'autres paramètres peuvent entrer en jeu dans le cadre du marketing viral. En particulier, divers éléments *négatifs* peuvent venir limiter la propagation de l'information.

2.6.3.1 Compétition entre plusieurs annonceurs.

Il arrive régulièrement que plusieurs annonceurs soient en compétition pour tenter de générer un « buzz » autour de leur marque. Ce fut par exemple le cas en 2013, lorsque les constructeurs Sony et Microsoft s'apprêtaient à sortir leurs nouvelles consoles de jeu [Mosca, 2013].

Ce cas est étudié dans [Bharathi et al., 2007], comme un problème de théorie des jeux. Une extension du modèle CTIC est définie, dans laquelle plusieurs informations se diffusent en parallèle, mais où chaque utilisateur n'est infecté que par la *première* information l'atteignant. Chaque joueur j_i sélectionne un ensemble de k utilisateurs à infecter au départ, avec pour objectif de maximiser sa propre fonction σ_i indiquant l'espérance du nombre d'utilisateurs finalement infectés par le produit vendu par j_i.

Il est montré que si un joueur j_i connaît les stratégies de tous les autres, σ_i devient une fonction sous-modulaire pouvant être maximisée avec l'algorithme glouton décrit précédemment. Une stratégie optimale pour le premier joueur est également donnée, mais seulement sur certains types de graphes.

2.6.3.2 Émergence d'opinions négatives.

Lorsqu'ils échangent des informations ou des idées, les utilisateurs ne sont pas toujours positifs. Un utilisateur n'ayant pas aimé un produit acheté en ligne pourra par exemple laisser une note défavorable à ce produit sur le site web du vendeur, et exprimer son mécontentement auprès de ses amis. Pour cette raison, une campagne de marketing virale peut parfois se retourner contre l'annonceur en générant un « Bad Buzz ».

La maximisation d'influence dans ce contexte a été étudiée dans [Chen et al., 2011]. Les auteurs définissent le modèle IC-N, une extension du modèle IC où chaque information se propageant est associée à un « facteur de qualité » q. Les utilisateurs de U_I ont chacun une probabilité q d'avoir une opinion positive, et une probabilité $1-q$ d'avoir une opinion négative. La diffusion se fait ensuite de la même façon que dans un modèle IC.

Lorsqu'un utilisateur est contaminé par un autre utilisateur avec une opinion positive, il adopte cette opinion positive avec une probabilité q, ou l'opinion négative avec une probabilité $1-q$. En revanche, lorsqu'un utilisateur est contaminé par un utilisateur ayant une opinion négative, il adopte directement l'opinion négative. Les opinions négatives se propagent donc mieux, ce qui est un résultat conforme à la réalité. L'algorithme proposé dans [Chen et al., 2010] est adapté à cette formulation.

Deux résultats particulièrement intéressants sont donnés.

- D'une part, l'impact de q sur la sélection de U_I peut être calculé en fonction du graphe.
- D'autre part, dans le cas d'une valeur de q faible, la sélection de U_I favorise les utilisateurs ayant un degré élevé. Ce résultat peut sembler contre-intuitif, mais peut s'expliquer ainsi : sélectionner les utilisateurs ayant un degré sortant élevé revient à favoriser les diffusions « courtes », où la majeure partie de la diffusion se fait entre les utilisateurs initiaux et leurs voisins, ce qui réduit la probabilité que beaucoup d'opinions négatives émergent : plus la distance entre une source et un utilisateur est grande, plus la probabilité que l'information devienne négative durant son trajet entre ces deux utilisateurs est élevée.

2.6.3.3 Présence de liens négatifs dans le graphe.

Enfin, il existe également des réseaux sociaux *signés*, contenant des relations utilisateurs *négatives*. Une relation négative peut indiquer qu'un utilisateur a une mauvaise opinion d'un autre, ou ne lui fait pas confiance. Dans ce cas, un avis positif sur un produit, partagé par une utilisateur u_i peut conduire à un avis négatif sur ce même produit chez un utilisateur u_j. L'article [Li et al., 2013] s'intéresse à ce cas.

La diffusion se fait selon un système de vote : à chaque pas de temps, chaque utilisateur adopte l'état (infecté / non-infecté) le plus représenté parmi ses voisins. Les auteurs

proposent des algorithmes pour la maximisation d'influence à court et long terme, et s'évaluent sur des graphes signés réels : slashdot et epinions.

2.7 Identification de leaders d'opinion

En parallèle de l'étude du problème de maximisation d'influence, d'autres travaux se sont attaqués à l'identification des utilisateurs les plus influents d'un réseau, en se basant exclusivement sur les propriétés du réseau social ou de ses utilisateurs, sans faire d'hypothèses sur le modèle de diffusion. Beaucoup d'articles dans ce domaine s'intéressent tout particulièrement à Twitter, car il s'agit d'un réseau très largement utilisé, où il est assez facile de récolter des données.

Dans le cadre de la maximisation d'influence, Kempe proposait une formulation rigoureuse du problème [Kempe et al., 2003]. Néanmoins, dans le cadre plus général de l'identification des leaders d'opinion, il n'existe pas de définition précise. Suivant le contexte et le réseau étudié, les termes « leaders d'opinion » ou « influenceurs » peuvent avoir des sens différents. Nous décrivons ici diverses propositions illustrant la variété des définitions possibles.

2.7.1 Approches Topologiques : mesures de centralité

Une approche intuitive pour la détection de leaders d'opinion serait de considérer tout simplement les degrés des utilisateurs dans le graphe social : il semble raisonnable de considérer qu'un utilisateur relié à beaucoup d'autres est un utilisateur important et influent.

Plusieurs articles [Cha et al., 2010, Kwak et al., 2010, Weng et al., 2010] ont étudié cette possibilité sur Twitter, où le nombre d'abonnés est couramment utilisé par les utilisateurs pour mesurer leur influence au sein du réseau. Leur conclusion est qu'il s'agit en vérité d'un assez mauvais indicateur de l'influence des utilisateurs. Selon [Weng et al., 2010] ceci peut s'expliquer par le fait que les liens d'un réseau social correspondent souvent à plusieurs sémantiques différentes qui se superposent. L'existence d'un lien peut indiquer l'influence d'un utilisateur sur un autre, mais peut aussi indiquer le fait qu'ils partagent certains centres d'intérêt. Tous les liens ne correspondent donc pas à des relations d'influence, et le degré des utilisateurs ne traduit donc pas forcément leur influence globale.

Sur Twitter, divers travaux proposent donc d'utiliser le nombre moyen de retweets des tweets de l'utilisateur, ou le nombre de « mentions[5] » désignant cet utilisateur chaque heure [Cha et al., 2010] pour évaluer son influence. Ces mesures semblent plus pertinentes, mais sont difficiles à évaluer en l'absence de vérité-terrain.

5. Terme utilisé sur Twitter pour désigner l'action consistant à mentionner le nom d'un autre utilisateur dans un Tweet. Une « mention » peut servir à répondre à un Tweet ou à prendre à parti un utilisateur.

Le fait que les degrés des utilisateurs ne représentent pas toujours l'importance de ceux-ci est un résultat connu dans le domaine de l'analyse des graphes. Par exemple, dans [Freeman, 1978], différentes mesures de *centralité* avait été comparées :

- mesures basées sur le degré ;
- mesures basées sur la proximité des utilisateurs, les utilisateurs centraux étant ceux proches de tous les autres ;
- mesures basées sur l'intermédiarité, les utilisateurs centraux étant ceux se trouvant souvent sur les plus courts chemins du graphe.

Il apparaît que ces mesures donnent des résultats assez différents. En particulier, les utilisateurs de centralités « moyennes » ont tendance à être très différents d'une mesure à l'autre.

Une problématique très similaire est le calcul de l'importance des pages internet dans un réseau hypertexte, notamment utilisée pour classer les résultats d'une recherche d'information. Dans ce contexte, l'algorithme PageRank [Page et al., 1999] a permis l'émergence de moteurs de recherche performants sur le Web. Il s'agit d'une mesure de centralité définie récursivement : l'importance d'une page est proportionnelle à l'importance des pages pointant vers elle. Dans le cas de Twitter, l'importance d'un utilisateur est proportionnelle à l'importance de ses abonnés. Intuitivement, le PageRank d'un sommet dans un graphe indique la probabilité qu'un agent se déplaçant aléatoirement dans ce graphe passe par ce sommet. À chaque pas, l'agent a une petite probabilité $1-d$ de se transporter directement à une page aléatoire au lieu de suivre un lien.

$$\mathrm{PageRank}(u_i) = \frac{1-d}{|U|} + d \sum_{u_j \in \mathrm{Succs}_i} \mathrm{PageRank}(u_j)$$

Notons bien que dans le cas de Twitter, l'ensemble Succs_i désigne les *followers* de l'utilisateur u_i, i.e. les ceux à qui u_i est susceptible de diffuser de l'information. Certains auteurs ont proposé d'utiliser le PageRank comme mesure de l'influence d'un utilisateur sur Twitter. Une première tentative se trouve dans [Kwak et al., 2010], les résultats étant évalués empiriquement. Dans [Weng et al., 2010], une version prenant en compte le fait que l'influence des utilisateurs dépend du sujet discuté est également proposée. Toutefois, il n'existe pas de vérité-terrain pour évaluer rigoureusement ces résultats, qui restent donc purement exploratoires.

Dans le même ordre d'idée, [Kitsak et al., 2010] propose d'utiliser la notion de $k-$noyau pour définir l'influence d'un utilisateur dans un graphe. Le $k-$noyau d'un graphe désigne sa plus grande composante connexe au sein de laquelle tous les sommets sont de degrés au moins k. Plus k est élevé, plus le $k-$noyau sera réduit. L'influence d'un utilisateur u_i est la valeur de k la plus élevé telle que $u_i \in k-$noyau. L'intuition sous-jacente est donc similaire à celle du PageRank : les utilisateurs les plus influents sont ceux appartenant à des $k-$noyau pour k assez élevé, c'est à dire ceux étant reliés à beaucoup d'utilisateurs eux mêmes reliés à beaucoup d'autres utilisateurs. Des tests utilisant des modèles de diffusion

de type SIS et SIR sont effectués et montrent que l'influence d'un utilisateur u_i ainsi calculée avec des $k-$ noyaux est un meilleur indicateur de la taille moyenne des cascades partant de cet utilisateur que son degré.

Enfin, une autre méthode de ce type, inspirée de l'algorithme *HITS* [Kleinberg, 1999] est décrite dans [Romero et al., 2011]. Dans cet article, les auteurs observent que sur Twitter, il est possible de définir l'influence d'un utilisateur u_i sur un de ses abonnés u_j en calculant le pourcentage $p_{i,j}$ de tweets de u_i retweetés par u_j. La difficulté est d'étendre cette notion d'influence *locale* (sur un utilisateur) à une notion d'influence *globale* (sur le réseau). Les auteurs proposent donc de définir *deux* valeurs pour chaque utilisateur : sa *passivité* et son *influence*.

— La passivité d'un utilisateur est sa tendance à ne *pas* être influencé par le contenu posté *par les utilisateurs influents* auxquels il est exposé.
— L'influence d'un utilisateur est sa capacité a influencer *les utilisateurs les plus passifs* (i.e. être retweeté).

Les deux valeurs sont donc définies de façon récursive. Plus précisément :

$$\begin{cases} \text{Influence}(u_i) = & \sum_{u_j \text{ abonné à } u_i} p_{i,j} \text{ Passivité}(u_j) \\ \text{Passivité}(u_i) = & \sum_{u_i \text{ abonné à } u_j} p_{j,i} \text{ Influence}(u_j) \end{cases}$$

Ces valeurs sont estimées itérativement, en étant successivement mises à jour en utilisant les valeurs de l'itération précédente. L'algorithme est notamment comparé à un algorithme de PageRank, et il est montré que la popularité des urls se diffusant sur Twitter est d'avantage corrélée à l'influence ainsi calculée des utilisateurs les partageant qu'à leurs scores de PageRanks.

2.7.2 Approches basées sur les propriétés des utilisateurs

Les approches topologiques ont souvent le défaut d'être coûteuses en termes de complexité algorithmique. D'autres méthodes, basées sur des propriétés locales des utilisateurs ont été proposées.

Une approche prédictive est présentée dans [Bakshy et al., 2011]. Les auteurs y étudient la diffusion d'hyperliens au sein de Twitter. Ils apprennent un arbre de régression visant à prédire l'influence d'un utilisateur (définie comme la taille moyenne des cascades débutant par celui-ci) en fonction de plusieurs propriétés de cet utilisateur : nombre de tweets, de followers, ancienneté, etc... Les tests effectués montrent que ce modèle prédit assez bien l'influence des utilisateurs, mais que la qualité de la prédiction souffre beaucoup du fait que les cascades longues soit très rares.

Un modèle similaire se trouve dans [Pal and Counts, 2011]. Chaque utilisateur y est représenté par un vecteur de caractéristiques basées sur son activité, son nombre de retweets, son utilisation des hashtags, etc. Toutes ces caractéristiques correspondent à des proprié-

tés susceptibles d'indiquer l'importance d'un utilisateur. L'identification des utilisateurs les plus influents se fait ensuite en observant les distributions des valeurs de ces caractéristiques au sein de la population, et en retenant les utilisateurs ayant des caractéristiques significativement plus élevées que la moyenne. Les résultats sont évalués en comparant les utilisateurs désignés par le modèle à ceux choisis par des expérimentateurs.

2.8 Détection de source

À mesure que l'utilisation des réseaux sociaux s'est développée, ceux-ci ont été de plus en plus utilisés pour diffuser des rumeurs, fausses informations ou des contenus volés ou piratés [Hooton, 2015]. Ce phénomène a motivé un certains nombre de travaux sur le problème de la *détection de source*. Il s'agit en fait du problème inverse de la prédiction de diffusion : le but est de retrouver l'utilisateur ayant partagé une information, la *source*, en observant le résultat de cette diffusion (typiquement, l'ensemble des utilisateurs infectés). Dans cette section, nous présentons différents travaux sur ce problème.

2.8.1 Mesure de centralité de rumeur

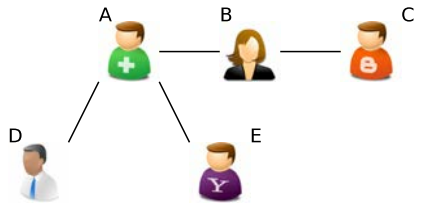
L'article fondateur de la détection de source dans le cadre de la diffusion d'information dans les réseaux sociaux date de 2010 [Shah and Zaman, 2010]. Cet article considère que le graphe de diffusion $G = (U, E)$ est connu et non-orienté. Les auteurs se basent sur un modèle de diffusion similaire à NetRate : au temps $t = 0$, un utilisateur-source u_s créé une information, et devient infecté. Tout utilisateur infecté u_i transmet l'information à chacun de ses voisins u_j après un temps $d_{i,j}$ tiré indépendamment pour chaque voisin selon une loi exponentielle de paramètre fixé pour tout le réseau.

L'ensemble U_T des utilisateurs infectés à un certain temps T est observé, et l'objectif est alors de retrouver lequel d'entre eux est l'utilisateur source. Notons bien que dans ce contexte, les *temps d'infections* de ces utilisateurs sont inconnus au moment de réaliser la prédiction. Les auteurs définissent un estimateur de type maximum de vraisemblance :

$$\hat{u}_s = \arg\max_{u_s \in U_T} P(U_T | u_s)$$

où $P(U_T|u_s)$ désigne la probabilité que l'ensemble des utilisateurs de U_T soient infectés au temps T sachant que l'utilisateur source est u_s, sous l'hypothèse du modèle de diffusion décrit plus haut. Malheureusement, le calcul de la valeur de $P(U_T|u_s)$ est complexe, car l'information partant de u_s peut avoir suivi différents chemins pour atteindre les utilisateurs de U_T.

Les auteurs s'intéressent donc d'abord au cas particulier où G est un arbre. Dans ce cas, le calcul est largement simplifié car il n'existe qu'un seul chemin possible entre n'importe



Utilisateur	ordres d'infections possibles	RC
a	abcde, abdce, adbce, abced, abdec, adbec, abecd, abedc, adebc, aebcd, aebdc, aedbc	12
b	badec, baedc, badce, baecd, bacde, baced, bcade, bcaed	8
c	cbade, cbaed	2
d	dabce, dabec, daebc	3
e	dabcd, dabdc, dadbc	3

FIGURE 2.4 – Exemple de l'utilisation de la mesure de *Rumor Centrality* (RC). À gauche, le sous-graphe des utilisateurs infectés. À droite, la liste pour chaque source potentielle des ordres d'infections possibles à partir de cette source. La RC d'une source potentielle est égale au nombre d'ordres possibles à partir de cette source.

quelle source potentielle u_s et n'importe quel utilisateur. Lorsque l'ensemble U_T est observé, *l'ordre exact* dans lequel les différents utilisateurs ont été infectés à partir d'une source possible $u_s \in U_T$ est inconnu, mais les liens du graphe G permettent toutefois de déduire un ordre partiel sur U_T pour cette source u_s. Dès lors, il est possible d'énumérer l'ensemble $\text{Ordres}(U_T, u_s)$ des ordres d'infections *possibles* de U_T à partir de u_s, i.e. tels que :

- u_s est infecté en premier ;
- aucun utilisateur n'est infecté avant qu'au moins un de ses prédécesseurs dans G ne le soit.

Une mesure RC baptisée « centralité de rumeur » est ensuite définie avec :

$$RC(U_T, u_s) = |\text{Ordres}(U_T, u_s)|$$

Il est montré que l'estimation de la source peut s'écrire :

$$\hat{u}_s = \arg\max_{u_s \in U_T} P(U_T|u_s) = \arg\max_{u_s \in U_T} RC(U_T, u_s)$$

Un exemple d'utilisation de la mesure RC est donné en figure 2.4. Les auteurs proposent un algorithme efficace, de type « passage de messages », pour calculer la valeur de RC. Cet algorithme est basé sur la relation suivante entre les centralités de deux utilisateurs u_i et u_j voisins dans le graphe (toujours dans le cas où G est un arbre) :

$$RC(U_T, u_i) = RC(U_T, u_j)\frac{SubTree_i^j}{N - SubTree_j^i}$$

où $SubTree_j^i$ désigne le nombre de nœuds dans le sous-arbre obtenu en partant de u_j et en s'éloignant de u_i. L'ensemble des centralités de rumeur peut donc être calculé de façon récursive, à partir de la centralité d'un nœud quelconque. Dans le cas général où G est un graphe quelconque, l'heuristique suivante est proposée.

— Pour chaque source possible $u_s \in U_T$, extraire un arbre $\mathcal{T}(u_s, G)$ en utilisant une exploration en largeur d'abord de G partant de u_s et limitée à U_T ;
— Calculer la $RC(U_T, u_s)$ dans cet arbre $\mathcal{T}(u_s, G)$

L'utilisation de l'arbre $\mathcal{T}(u_s, G)$ extrait avec une exploration en largeur d'abord est justifiée par le fait que la longueur du plus court chemin entre la source et n'importe quel autre utilisateur dans $\mathcal{T}(u_s, G)$ est égale à celle dans le graphe G. En d'autres termes, au lieu de considérer tous les chemins possibles pour calculer la valeur de $P(U_T|u_s)$, seuls les plus courts - et donc les plus vraisemblables - sont pris en compte.

Plusieurs résultats théoriques sont fournis, concernant la probabilité de détection sur différents types de graphe. Les auteurs montrent ainsi que si G est un arbre régulier de degré $d = 2$ (ou « graphe-ligne »), la probabilité de détection de la source tend vers 0. En revanche, si G est un arbre régulier de degré $d > 2$, la probabilité de détection est non-triviale. L'heuristique pour les graphes généraux est testée sur des épisodes de diffusions synthétiques générées sur des graphes réels, et comparée à une mesure de centralité de distance classique consistant à choisir la source minimisant la somme des distances aux utilisateurs infectés :

$$\hat{u}_s = \arg\min_{u_s \in U_T} \sum_{u_i \in U_T} \text{Dist}(u_s, u_i)$$

où $D(u_s, u_i)$ est la longueur du plus court chemin entre u_s et u_i. Les auteurs observent que leur approche obtient de meilleurs résultats, en terme de distance à la vraie source, que la mesure de centralité de distance. Ce travail a ensuite été poursuivi dans [Shah and Zaman, 2012], où des résultats théoriques sont donnés pour d'autres types de graphes.

2.8.2 Autres estimateurs

En parallèle, [Luo et al., 2015b] se sont intéressés au cas où l'information peut se diffuser selon des modèles de type SI, SIR, SIRI ou SIS (décrits en section 2.3.2) dans un graphe. À la place d'un estimateur de type vraisemblance maximale, les auteurs utilisent l'estimateur suivant, précédemment défini dans [Zhu and Ying, 2013] :

$$\hat{u}_s = \arg\max_{u_s \in U_T} \max_{tree \in \mathcal{T}(U_T)} P(tree|u_s)$$

où $\mathcal{T}(U_T)$ désigne l'ensemble des arbres couvrants de U_T, et $P(tree|u_s)$ est la probabilité que l'information partant de la sources u_s se diffuse en suivant l'arbre de diffusion $tree$ dans le graphe, sous l'hypothèse du modèle de diffusion considéré. Ainsi, les auteurs considère uniquement l'arbre de diffusion enraciné en u_s le plus vraisemblable pour chaque source potentielle u_s au lieu de calculer la valeur exacte de $P(U_T|u_s)$ en énumérant tous les arbres possibles. L'idée est donc la même que celle utilisée dans [Shah and Zaman, 2010] pour les graphes quelconques.

Dans ce cadre, les auteurs proposent d'utiliser un *centre de Jordan*. Celui-ci est défini par :

$$JC(U_T) = \arg\min_{u_s \in U_T} \max_{u_i \in (U_T)} \text{Dist}(u_s, u_i)$$

Il est montré que le *Centre de Jordan* de U_T constitue un estimateur « universel » de $\hat{u}_s$, c'est à dire s'appliquant aux différents modèles de diffusion possibles (SI, SIR, SIRI, etc...) Intuitivement, l'utilisation du centre de Jordan revient à sélectionner la source minimisant le nombre de pas nécessaires pour contaminer les utilisateurs de U_T. Ce centre présente l'avantage de pouvoir être calculé en temps $O(|U| \times |E|)$.

Les auteurs expérimentent cet estimateur sur des graphes réels avec des épisodes de diffusion artificiels, en se comparant à d'autres mesures de centralité, et observent de meilleurs résultats avec le centre de Jordan.

Plus tard, le problème a également été abordé dans [Dong et al., 2013]. Cet article étudie le cas où il existe un *a priori* sur les différentes sources possibles. Plusieurs résultats théoriques sont donnés, concernant l'impact du nombre de sources possibles a priori et du type de graphe considéré.

2.8.3 Contexte d'observation partielle

Tous les travaux précédents considèrent que l'état de *l'ensemble* des utilisateurs du réseau est observé à un temps T. Plus récemment, le cas où seuls les états d'une partie $O \subset U_T$ des utilisateurs sont observés a été étudié dans [Seo et al., 2012]. Les utilisateurs de O ainsi observés sont dits « monitorés »

Différentes méthodes pour sélectionner les utilisateurs à monitorer sont définies : centralité dans le graphe, degrés, maximisation de la distance entre moniteurs, etc... De plus, une heuristique basée sur quatre mesures différentes visant à retrouver la sources à partir de l'état (infecté/non-infecté) des utilisateurs monitorés est également proposée.

Les heuristiques de sélection d'utilisateurs sont testées sur des données réelles issues de Twitter. Les auteurs observent que la meilleure est celle consistant à choisir les utilisateurs de façon à ce que les distances entre eux dans G soient toujours supérieures à un certain seuil k. Cela revient à « éparpiller » au maximum les utilisateurs monitorés dans le graphe, ce qui constitue un résultat intuitif : éparpiller les utilisateurs permet de mieux couvrir le graphe et donc de maximiser la quantité d'information observée.

Prise en compte des temps d'infection Dans le contexte où seul l'état d'une partie des utilisateurs est observé, il devient intéressant d'étudier le cas où les temps d'infection de ces utilisateurs observés sont connus (le cas où tous les utilisateurs sont observés *avec* leurs temps d'infection est trivial, la source étant dans ce cas le premier utilisateur infecté).

Une première tentative se trouve dans [Pinto et al., 2012]. Soit D^O un épisode de diffusion « partiel » où seuls les états et les temps d'infections du sous-ensemble d'utilisateurs O sont observés. Les utilisateurs non observés sont notés $H = U \setminus O$, et nous avons donc : $D = D^O \cup D^H$. Les auteurs considèrent que la diffusion suit un modèle où chaque utilisateur infecté transmet l'information à chacun de ses successeurs après un délai tiré sur chaque lien selon une loi gaussienne de paramètres fixés. Il définissent ensuite un estimateur par maximum de vraisemblance :

$$\hat{u}_s = \arg\max_{u_s \in U} P(D^O | u_s)$$

où $P(D^O|u_s)$ est la probabilité d'observer l'épisode partiel D^O quand u_s est la source, sous l'hypothèse du modèle de diffusion décrit. Là encore, ce calcul est très complexe. En effet, le modèle de diffusion défini ne permet pas de calculer $P(D^O|u_s)$ directement : cette probabilité n'est définie que lorsque $O = U$, c'est à dire lorsque D est entièrement observé. Pour estimer cette probabilité dans le cas d'une observation partielle, il est donc nécessaire d'énumérer l'ensemble des observations manquantes possibles pour les utilisateurs cachés.

$$\hat{u}_s = \arg\max_{u_s \in U} \int_{D_x^H \in D^H \text{possibles}} \left(P(D^H \cup D^O | u_s) \prod_{u_i \in H} dt_j^{D^H} \right)$$

Le calcul de cette vraisemblance doit donc prendre en compte deux sources d'incertitude : celle concernant les états et les temps d'infection des utilisateurs non-observés et celle concernant les chemins suivis par l'information. Le calcul exact ne passant pas l'échelle, les auteurs adoptent une méthodologie similaire à celle de [Shah and Zaman, 2010], le premier article présenté dans cette section : ils commencent par étudier le cas où le graphe G est un arbre, ce qui supprime la complexité liée à l'énumération des chemins possibles. Dans ce cas, la vraisemblance d'une source peut-être calculée de façon exacte, avec une complexité linéaire. Dans le cas où G est un graphe quelconque, l'estimation de la vraisemblance d'une source se fait dans l'arbre $\mathcal{T}(u_s, G)$ extrait de G avec une recherche en largeur d'abord à partir de u_s. De la même façon que dans [Shah and Zaman, 2010], cela revient à considérer seulement l'arbre de diffusion le plus vraisemblable plutôt que tous les arbres possibles.

Enfin, l'article [Farajtabar et al., 2015] s'est placé dans un contexte similaire, mais en considérant cette fois que le graphe G est inconnu. Ce graphe est donc estimé à partir d'un ensemble d'épisodes de diffusion d'apprentissage $\mathcal{D}$ en utilisant une extension de l'algorithme NetRate décrite dans [Daneshmand et al., 2014]. La détection de source se fait toujours avec un estimateur de maximum de vraisemblance, suivant le modèle de diffusion NetRate [Gomez-Rodriguez et al., 2011]. De la même façon que dans l'article précédent [Pinto et al., 2012], cette vraisemblance est difficile à calculer à cause de la complexité liée à la présence d'utilisateurs dont l'état est inconnu. Toutefois, plutôt que de proposer une heuristique basée sur l'extraction de l'arbre de diffusion le plus vraisemblable, les auteurs proposent une méthode d'approximation basée sur l'intégration par échantillonnage

préférentiel. En effet, l'intégration peut être approximée par une somme :

$$\int_{D_x^H \in D^H \text{possibles}} \left(P(D^H \cup D^O | u_s) \prod_{u_i \in H} dt_j^{D^H} \right) \approx \frac{1}{\Gamma} \sum_{x=1}^{X} P(D_x^H \cup D^O | u_s)$$

où $(D_x^H)_{x=1..X}$ est un ensemble de valeurs possibles de la partie cachée D^H, tirées aléatoirement, et $\frac{1}{\Gamma}$ un terme de normalisation. Ainsi, au lieu de réaliser une intégration sur toutes les valeurs possibles de D^H, nous approximons cette intégrale en tirant seulement X valeurs possibles de D^H. La qualité de cette approximation augmente avec X. L'intégration par échantillonnage préférentiel consiste ensuite à favoriser les valeurs plus probables de D^H, qui ont un impact plus grand sur le calcul de la somme. Pour cela, l'échantillonnage préférentiel utilise les longueurs des plus courts chemins dans le graphe de façon à favoriser le tirage de temps d'infection « vraisemblables » pour les utilisateurs cachés.

Contrairement aux autres modèles présentés dans cette section, celui-ci est testé sur des données réelles, mais ne parvient à retrouver la source que dans le cas où *plusieurs* diffusions partant d'une même source sont observées, ce qui n'est pas réaliste dans beaucoup d'applications.

2.8.4 La détection de source comme problème adverse

Une formulation intéressante du problème de détection de source a été donnée par [Luo et al., 2015a]. Le contexte est ici celui d'un jeu opposant deux joueurs : une source diffusant des informations dans un réseau, et l'administrateur dudit réseau cherchant à identifier la source (pour la bannir du réseau parce qu'elle diffuse du contenu illégal, par exemple). L'objectif du joueur « source » est d'infecter un maximum d'utilisateurs (ce qui est associé à une récompense) sans être repérée (ce qui est associé à un coût). L'objectif du joueur « administrateur » est de retrouver cette source tout en inspectant un minimum d'utilisateurs, cette inspection ayant un coût. Les auteurs formulent une série d'hypothèses sur les mécanismes de diffusion et les actions possibles des joueurs, et étudient l'existence d'un équilibre de Nash dans ce contexte. Ils remarquent en particulier que si un équilibre de Nash existe, la stratégie optimale de l'administrateur est d'inspecter uniquement le centre de Jordan.

2.8.5 Détection de plusieurs sources

Dans la plupart des applications, une seule source est à l'origine de chaque rumeur. Certains travaux ont toutefois étudié le cas où *plusieurs* utilisateurs lancent une rumeur en même temps.

Dans [Lappas et al., 2010], les auteurs se placent dans le cadre du modèle IC et définissent le problème des *k-effectors*. Étant donné un *vecteur d'activation* a, i.e. un vecteur binaire

de taille N indiquant quels utilisateur ont été infectés par une information, le but est de retrouver un ensemble X d'utilisateurs minimisant le coût :

$$C(X) = \sum_{u_i \in U} |a(i) - \alpha(i, X)|$$

où $\alpha(i, X)$ désigne la probabilité que l'utilisateur u_i devienne infecté durant une diffusion démarrant par l'ensemble X. De la même façon que dans [Shah and Zaman, 2010], les auteurs montrent que le problème est difficile (NP-difficile, dans ce cas) et commencent par étudier le cas où G est un arbre, et proposent deux heuristiques. Si G est un graphe quelconque, ils proposent d'extraire un arbre couvrant et appliquent leurs heuristiques sur cet arbre. L'approche est testée sur des données de diffusion de mots-clés au sein d'une communauté de chercheurs.

L'utilisation des centres de Jordan pour la détection de source a également été étudiée dans le cas de la détection de *plusieurs* sources [Luo et al., 2015b].

Une autre approche se trouve dans [Prakash et al., 2012]. Les auteurs proposent ici une méthode permettant de prédire le nombre de sources, puis leurs identités. La méthode proposée, baptisée NETSLEUTH, est basée sur le principe de longueur de description minimale : le but des auteurs est de décrire parfaitement l'ensemble d'utilisateurs infectés U_T en utilisant le moins de bits possible.

Pour cela, ils considèrent que la diffusion se fait selon un modèle SI : à chaque pas de temps, chaque utilisateur infecté tente de contaminer chacun de ses voisins, avec une probabilité de succès égale à β. Pour encoder de la façon la plus efficace possible un épisode de diffusion, les auteurs utilisent la procédure suivante.

- Encoder le nombre de sources, avec un code favorisant les valeurs plus faibles (qui sont plus probables).
- Encoder l'identité des sources. Cela peut se faire efficacement en remarquant qu'une fois le nombre x de sources connu, il existe seulement $\binom{N}{x}$ ensembles de sources possibles. En définissant un ordre sur ces ensembles, cette information peut être encodée en $log_2\left(\binom{N}{x}\right)$ bits.
- Encoder le nombre d'itérations nécessaires pour infecter tous les utilisateurs.
- Pour chacune de ces itérations, encoder la liste des nouveaux utilisateurs infectés. Cette information peut être compressée en observant que seule une partie des utilisateurs peuvent être contaminés à chaque pas de temps (ceux dont au moins un voisin est déjà infecté), ce qui réduit grandement le nombre de bits nécessaires.

Les auteurs cherchent l'ensemble de sources minimisant le nombre de bits nécessaires pour décrire U_T suivant cette procédure, afin de trouver le nombre de sources et leurs identités. Cette minimisation étant complexe, une approximation utilisant un algorithme glouton est proposée. Celle-ci est basée sur l'extraction des vecteurs propres de la matrice laplacienne du graphe des utilisateurs infectés. Contrairement à beaucoup d'autres travaux présentés dans cette section, cet article n'est donc *pas* basé sur un maximum de vrai-

semblance. NetSleuth est testé sur des graphes artificiels et réels, avec des épisodes de diffusion synthétiques, et parvient bien à retrouver le *nombre* de sources ainsi que leurs identités.

2.9 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur le sujet de la diffusion d'information dans les réseaux sociaux. La diversité des tâches et des travaux nous montre que l'expression recouvre en pratique un champ applicatif assez vaste. Nous pouvons toutefois dégager plusieurs difficultés communes à la plupart des articles présentés. Dans cette thèse, nous serons nous aussi confrontés à ces problèmes.

La diffusion d'information est un phénomène *rare* : chaque jour, une énorme quantité d'information est générée sur internet. Toutefois, seule une infime partie de celle-ci devient particulièrement populaire. De nombreux travaux indiquent que la répartition des tailles des cascades suit une loi de puissance (en général de paramètre $\alpha \approx 2$ [Cheng et al., 2014]). Ce déséquilibre peut être source de difficultés. Le manque de longues cascades rend notamment les données rares, ce qui complique l'apprentissage des paramètres d'un modèle ou l'extraction des caractéristiques des utilisateurs. De plus, prédire un phénomène rare est toujours délicat.

La diffusion d'information est un phénomène *chaotique* : s'il est possible d'observer des régularités à un niveau de diffusion global, les comportements des utilisateurs et leurs interactions sont très variables et délicats à caractériser.

Les modèles ont une *complexité* calculatoire importante : avec le développement des réseaux sociaux en ligne, la taille de ceux-ci a largement augmenté. Pratiquement tous les modèles basés sur le graphe social se heurtent à des problèmes de passage à l'échelle. Il devient rapidement nécessaire de proposer des heuristiques pour remplacer un calcul exact dans le graphe, ou de définir des méthodes ne reposant pas sur ce graphe.

Les problématiques peuvent être *propres à chaque réseau* : la notion de « réseau social en ligne » est assez floue et désigne de très nombreux services web fonctionnant de façons différentes. Il est ainsi courant qu'un modèle soit défini pour un réseau particulier et ne s'adapte pas, ou mal, à un autre. De plus, les possibilités sont étroitement liées à la disponibilité des données : de nombreuses informations pertinentes ne sont pas accessibles par le biais des API offertes par les grands réseaux sociaux en ligne. Par exemple, si Twitter était très ouvert durant ses premières années, il l'est beaucoup moins aujourd'hui : il n'est plus possible de récupérer l'intégralité du trafic facilement.

Les tâches sont *mal définies* : pour les différentes tâches que nous avons présentées, il n'existe pas de définition formelle consensuelle (exception faite de la maxi-

misation d'influence). Les articles cités étudient des contextes expérimentaux variés et pas toujours compatibles entre eux. L'évaluation des performances reste également un problème ouvert. En particulier, dans un certain nombre de travaux sur la prédiction de diffusion ou la détection de sources présentés dans ce chapitre, l'évaluation est réalisée sur des épisodes de diffusion synthétiques générés selon un modèle connu dans un graphe réel, et non pas sur des épisodes issus de données réelles. Cela est susceptible de limiter la pertinence des résultats ainsi obtenus. De la même façon, sur la tâche de maximisation d'influence, les modèles sont évalués en *simulant* la diffusion dans un réseau social, et non pas en observant de vraies expériences de marketing viral dans un réseau.

Chapitre 3

Relaxation et régularisation du modèle IC

Résumé Ce chapitre présente une première contribution, publiée dans [Lamprier et al., 2015]. Nous proposons d'apprendre les paramètres du modèle IC selon la méthode de [Saito et al., 2008] mais en relaxant les contraintes sur des délais de transmission, afin d'obtenir un modèle plus robuste que [Saito et al., 2008], que nous testons sur des données réelles. Nous proposons également de régulariser les probabilités apprises afin de limiter l'effet du surapprentissage sur certains corpus.

3.1 Difficultés liées à l'apprentissage d'IC

Dans ce chapitre, nous nous intéressons au modèle *Independent Cascades*, qui est à la base de nombreux travaux présentés dans le chapitre 2. Sa capacité à expliquer de manière relativement réaliste de nombreux processus de diffusion, tout en conservant une certaine simplicité grâce à ses hypothèses d'indépendance, en fait en effet un des modèles les plus étudiés.

Nous avons vu dans le chapitre 2 que le modèle IC était un modèle génératif basé sur le graphe social : lorsqu'un utilisateur u_i devient infecté, il tente de contaminer chacun de ses voisins u_j avec une certaine probabilité de réussite $p_{i,j}$. L'apprentissage du modèle IC revient donc à apprendre une probabilité de transmission sur chaque lien du graphe social, à partir d'un ensemble $\mathcal{D}$ d'épisodes de diffusion observés. Nous noterons $\mathcal{P}$ cet ensemble de probabilités.

Une difficulté de l'apprentissage de $\mathcal{P}$ vient de la notion d'épisode de diffusion. Rappelons qu'un épisode de diffusion D désigne une séquence d'utilisateurs infectés par une même

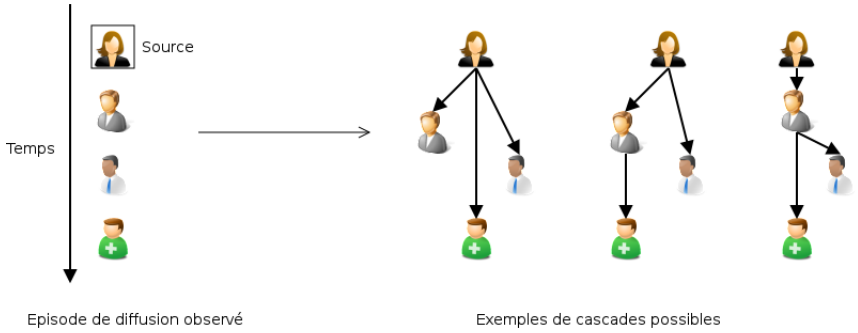


FIGURE 3.1 – Un exemple d'épisode de diffusion et de cascades possibles. Les structures de cascades représentent plusieurs façons dont l'information a pu se transmettre dans le réseau.

information, avec leurs temps d'infection associés :

$$D = \big((u_i, t_i^D), (u_j, t_j^D), (u_k, t_k^D), \dots\big)$$

Dans un épisode de diffusion, nous savons *quand* a été infecté chaque utilisateur, mais nous ne savons pas *par qui*. Un exemple se trouve en figure 3.1. Dans cet exemple, nous pouvons constater que plusieurs structures de diffusions, ou « cascades », sont susceptibles d'expliquer un même épisode de diffusion.

Si cette information manquante était connue, l'estimation des paramètres du modèles IC serait simple. En effet, il suffirait pour estimer chaque $p_{i,j}$ de compter le nombre d'épisodes de diffusion où l'utilisateur u_i a *contaminé* l'utilisateur u_j, et de diviser cette valeur par le nombre d'épisodes où u_i a *tenté* de contaminer u_j (voir section 3.2.1). Dans un épisode de diffusion D, nous pouvons savoir quelles *tentatives* de transmission ont eu lieu : chaque utilisateur, lorsqu'il devient infecté dans D, tente de contaminer chacun de ses voisins non infectés. En revanche, nous ne savons pas quelles tentatives ont réussi.

L'incertitude liée à cette information manquante peut être limitée en ajoutant certains a priori. En particulier, tous les modèles graphiques font l'hypothèse qu'un utilisateur infecté a forcément été contaminé par un de ses prédécesseurs déjà infectés. Il est également possible d'ajouter un a priori sur le délais de transmission, i.e. le temps mis par un utilisateur pour en contaminer un autre. Ces possibilités sont discutées dans les prochaines sous-sections.

3.1.1 Graphe du réseau social

Le modèle IC classique, comme beaucoup d'autres modèles présentés dans le chapitre 2, fait l'hypothèse que la diffusion ne peut avoir lieu que sur les liens du graphe du réseau social, supposé connu. Ainsi, un utilisateur infecté dans un épisode de diffusion ne peut

avoir été contaminé que par l'un de ses voisins précédemment infectés. Cela restreint les structures de cascades possibles pour un épisode de diffusion donné.

Toutefois, il est à noter que quand aucun graphe explicite n'est disponible, ou lorsque les relations connues ne représentent pas les canaux de diffusion étudiés (voir chapitre 2, section 2.5.1.4), les probabilités de diffusion peuvent être définies sur le graphe *complet* reliant tous les utilisateurs. Cela revient alors à apprendre les canaux de diffusion uniquement à partir des comportements observés, sans a priori sur le graphe de diffusion sous-jacent, de la même façon que dans [Gomez-Rodriguez et al., 2011]. C'est dans ce cadre que nous nous placerons dans ce chapitre et dans ce manuscrit.

3.1.2 Discrétisation du temps

Le modèle IC classique fait l'hypothèse que la contamination a lieu durant des pas de temps consécutifs : lorsqu'un utilisateur devient infecté au pas de temps t, il dispose d'une unique chance d'infecter chacun de ses voisins au pas de temps $t+1$. En conséquence, un utilisateur infecté au temps t dans un épisode de diffusion D ne peut avoir été contaminé que par un utilisateur ayant lui même été contaminé au temps $t-1$.

Cependant, dans les corpus issus de sites internet, le temps est renseigné sous forme d'un « timestamp », i.e. le nombre de secondes écoulées depuis le 1er janvier 1970. Or, il est bien évident que dans le cas de la diffusion d'information sur les réseaux sociaux, il est impossible qu'un utilisateur contamine un de ses voisins après une durée d'une seconde, qui est bien trop courte. Il est donc nécessaire de définir une « longueur de pas de temps » raisonnablement grande (par exemple, quelques minutes sur Twitter ou quelques heures sur Facebook).

Cela pose une difficulté majeure : comment choisir le bon « pas de temps » ? Avec un pas de temps trop grand, de nombreux utilisateurs peuvent être regroupés au sein d'une même itération du modèle, ce qui risque de masquer de très nombreuses relations entre eux. À l'inverse, un pas de temps trop court peut rendre impossible la modélisation d'épisodes où l'interval de temps entre deux infections est trop long.

La figure 3.2 illustre ce problème. Nous montrons comment un même épisode de diffusion peut être discrétisé de différentes manières. Sur cette figure, nous pouvons voir que les relations utilisateurs susceptibles d'être inférées à partir d'un épisode de diffusion dépendent beaucoup du pas de temps considéré. Par exemple, avec un pas de temps de une seconde, il est impossible que l'utilisateur vert ait été contaminé par l'utilisateur gris, alors que cela est possible avec un pas de temps de une ou deux minutes. Avec un pas fixé à dix minutes, toutes les infections ont lieu au temps initial et l'épisode de diffusion représenté à gauche n'apporte donc aucune information. Cet exemple simple illustre le fait que le *choix* d'une valeur de pas de temps constitue un a priori fort sur la dynamique de la diffusion, et qu'il

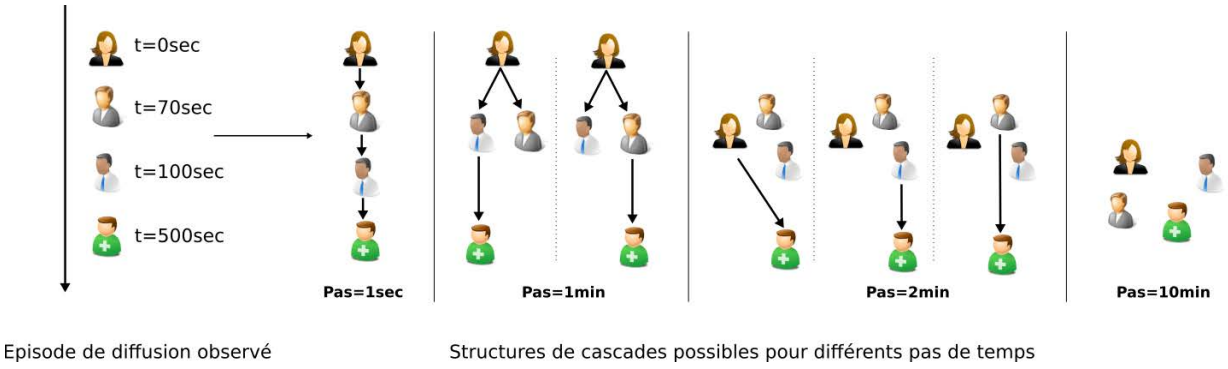


FIGURE 3.2 – Importance du choix du pas de temps

n'existe pas de « bonne » solution, a fortiori sur des ensembles de plusieurs dizaines de milliers d'épisodes de diffusion.

3.1.3 Modélisation du temps

Pour dépasser cette simple discrétisation du temps et les difficultés qu'elle soulève, plusieurs articles ont proposé de modéliser les délais d'infections, conjointement aux probabilités d'infection. C'est notamment le cas du modèle CTIC et du modèle continu de Leskovec (NetRate), décrits dans le chapitre 2 :

— le modèle CTIC est une extension du modèle IC qui considère que lorsqu'un utilisateur u_i en contamine un autre u_j (ce qui se produit avec une probabilité $p_{i,j}$), l'infection a lieu après un délai $d_{i,j}$ tiré selon une certaine loi de probabilité, de paramètre $r_{i,j}$ devant également être appris pour chaque lien ;
— le modèle NetRate considère que la probabilité de transmission de u_j par u_i dépend du temps : à chaque instant t, la probabilité que u_i contamine u_j dépend du temps écoulé depuis l'infection de u_i, suivant une certaine loi de probabilité de paramètre $r_{i,j}$. Ce modèle est en fait équivalent à un modèle CTIC dont les probabilités de transmission seraient toutes égales à 1.

Toutefois, les régularités sur les délais d'infection nous semblent difficiles à extraire d'épisodes de diffusion issus de données réelles. Estimer l'influence qu'ont les utilisateurs les uns sur les autres constitue déjà un problème difficile. Ajouter à ce problème l'extraction de régularités sur les délais d'infections à partir de données de diffusion très parcimonieuses complexifie encore la tâche. De plus, dans ces modèles, les délais d'infection observés dans les données d'apprentissage ont un impact non-négligeable sur les probabilités apprises. L'apprentissage des probabilités peut donc souffrir de la grande variance de ces délais.

3.2 Delay-Agnostic Independant Cascades (DAIC)

Face à ces difficultés, il peut apparaître viable de s'abstraire de cette dimension temporelle pour la modélisation de la diffusion. Cela nous a amené à une contribution introductive à ce travail de thèse, consistant en la proposition d'un algorithme d'apprentissage du modèle IC où nous considérons qu'un utilisateur $u_j \in D$ peut avoir été infecté par *n'importe lequel* de ses prédécesseurs déjà infectés, indépendamment de leurs temps d'infections. Cela revient de fait à considérer que les délais de transmission suivent une loi uniforme. Nous baptisons cet algorithme « Delay-Agnostic IC », ou *DAIC*.

Cette approche est également justifiée par le fait que la modélisation des délais de contamination, et donc la prédiction des temps d'infections, n'est pas essentielle dans de nombreuses applications. Par exemple, dans le cas du problème de *maximisation d'influence* présenté dans le chapitre 2, il est seulement nécessaire de prédire *quels* utilisateurs seront infectés, ou *combien*, mais pas *quand*.

3.2.1 Apprentissage

Dans notre modèle, comme dans le modèle IC, chaque utilisateur devenant infecté dispose d'une unique chance d'infecter chacun de ses voisins. Toutefois, nous considérons que cette contamination peut avoir lieu après un délai quelconque, et pas forcément au pas de temps suivant comme dans le modèle IC classique. Un utilisateur u_j infecté dans un épisode de diffusion D est donc susceptible d'avoir été contaminé par n'importe quel prédécesseur infecté avant lui.

Nous suivons ensuite la méthodologie définie dans [Saito et al., 2008] pour apprendre l'ensemble des probabilités de transmission $\mathcal{P}$. Soit u_j un utilisateur, et D un épisode de diffusion. L'ensemble $(U^D_{t_j} \cap \text{Preds}_j)$ est nommé « ensemble des *infecteurs potentiels* » de u_j. Cet ensemble correspond à l'ensemble des utilisateurs tentant de transmettre à u_j l'information considérée. Dans notre cas il s'agit donc de l'ensemble des prédécesseurs de u_j infectés avant lui. La probabilité d'infection de u_j dans un épisode D est donc la probabilité qu'*au moins un* des utilisateurs de $(U^D_{t_j} \cap \text{Preds}_j)$ transmette l'information à u_j. Cette probabilité est notée $P(u_j|U^D_{t_j}, \mathcal{P})$ et vaut :

$$P(u_j|U^D_{t_j}, \mathcal{P}) = 1 - \prod_{u_i \in (U^D_{t_j} \cap \text{Preds}_j)} (1 - p_{i,j}) \tag{3.1}$$

Rappelons ici que $p_{i,j}$ désigne la probabilité de transmission de u_i vers u_j.

La probabilité d'observer un épisode de diffusion D dépend alors de la probabilité d'observer :

— l'infection de chaque utilisateur $u_j \in U^D_\infty$, en connaissant l'ensemble des utilisateurs infectés avant lui $U^D_{t_j}$;

— la non-infection de chaque utilisateur $u_j \in \bar{U}_\infty^D$.

$$\begin{aligned} P = (D|\mathcal{P}) &= \prod_{u_j \in U_\infty^D} P(u_j|U_{t_j}^D, \mathcal{P}) \times \prod_{u_j \in \bar{U}_\infty^D} (1 - P(u_j|U_\infty^D, \mathcal{P})) \\ &= \prod_{u_j \in U_\infty^D} P(u_j|U_{t_j}^D, \mathcal{P}) \times \prod_{u_j \in \bar{U}_\infty^D} \prod_{u_i \in U_\infty^D} (1 - p_{i,j}) \end{aligned} \tag{3.2}$$

La log-vraisemblance d'un ensemble de paramètres $\mathcal{P}$ (les probabilités de transmission) par rapport à un ensemble $\mathcal{D}$ d'épisodes de diffusion observés est donc donnée par :

$$\begin{aligned} \mathcal{L}(\mathcal{P}; \mathcal{D}) &= \sum_{D \in \mathcal{D}} \log P(D|\mathcal{P}) \\ &= \sum_{D \in \mathcal{D}} \left(\sum_{u_j \in U_\infty^D} \log(P_j^D) + \sum_{u_j \in \bar{U}_\infty^D} \sum_{u_i \in U_\infty^D} \log(1 - p_{i,j}) \right) \end{aligned} \tag{3.3}$$

où P_j^D est une écriture simplifiée de $P(u_j|U_{t_j}^D, \mathcal{P})$. Le problème d'apprentissage des probabilités s'écrit alors :

$$\mathcal{P}^\star = \arg\max_{\mathcal{P}} \mathcal{L}(\mathcal{P}; \mathcal{D})$$

Malheureusement, l'optimisation de cette log-vraisemblance est difficile, à cause de la définition de P_j^D (équation 3.1). Toutefois, comme nous l'avons expliqué au début du chapitre, l'estimation de $\mathcal{P}$ serait largement facilitée si nous savions *qui a infecté qui* (ou plus exactement : *quelles tentatives de contamination ont réussi*). Cette information manquante correspond donc à un *facteur latent* du modèle. C'est précisément dans ce genre de situation qu'un algorithme d'espérance-maximisation est indiqué [Dempster et al., 1977]. Nous suivons donc la méthodologie de [Saito et al., 2008] pour optimiser $\mathcal{L}(\mathcal{P}; \mathcal{D})$.

Soit $\mathcal{X} = \left(X_{i \to j}^D\right)_{D \in \mathcal{D}, (u_i, u_j) \in E}$ l'information manquante, indiquant quelles tentatives de contamination ont réussi dans $\mathcal{D}$. Nous notons :

$$X_{i \to j}^D = \begin{cases} 1 & \text{si } u_i \text{ a réussi à contaminer } u_j \text{ dans } D \\ 0 & \text{sinon} \end{cases}$$

Si l'information $\mathcal{X}$ était connue, la log-vraisemblance d'un ensemble de paramètres $\mathcal{P}$ par rapport aux données complétées $(\mathcal{D}, \mathcal{X})$ serait égale à la log-vraisemblance des contaminations et des non-contaminations indiquées par $\mathcal{X}$:

$$\begin{aligned} \mathcal{L}\left(\mathcal{P}; (\mathcal{D}, \mathcal{X})\right) = &\sum_{D \in \mathcal{D}} \sum_{u_j \in U_\infty^D} \sum_{u_i \in (U_{t_j}^D \cap \mathrm{Preds}_j)} \left(X_{i \to j}^D \log(p_{i,j}) + \left(1 - X_{i \to j}^D\right) \log(1 - p_{i,j}) \right) \\ &+ \sum_{D \in \mathcal{D}} \sum_{u_j \in \bar{U}_\infty^D} \sum_{u_i \in U_\infty^D} \log(1 - p_{i,j}) \end{aligned} \tag{3.4}$$

En pratique, $\mathcal{X}$ est inconnu mais nous pouvons calculer l'*espérance* de $X^D_{i\to j}$ lorsque u_j est infecté dans un épisode D et que u_i fait partie de ses infecteurs potentiels, en nous basant sur une estimation courante de $\mathcal{P}$ notée $\hat{\mathcal{P}}$. Cela se fait en appliquant le théorème de Bayes, et en remarquant que l'espérance de $X^D_{i,j}$ est égale à la probabilité que sa valeur soit égale à 1 :

$$\begin{aligned}
&\mathbb{E}\left[X^D_{i\to j} | u_j \in U^D_\infty, u_i \in (U^D_{t_j} \cap \operatorname*{Preds}_j), \hat{\mathcal{P}}\right] \\
&= P(X^D_{i\to j} = 1 | u_j \in U^D_\infty, u_i \in (U^D_{t_j} \cap \operatorname*{Preds}_j), \hat{\mathcal{P}}) \\
&= \frac{P(u_j \in U^D_\infty | X^D_{i\to j} = 1, u_i \in (U^D_{t_j} \cap \operatorname{Preds}_j), \hat{\mathcal{P}}) \times P(X^D_{i\to j} = 1 | u_i \in (U^D_{t_j} \cap \operatorname{Preds}_j), \hat{\mathcal{P}})}{P(u_j \in U^D_\infty | u_i \in (U^D_{t_j} \cap \operatorname{Preds}_j), \hat{\mathcal{P}})} \\
&= \frac{1 \times P(X^D_{i\to j} = 1 | u_i \in (U^D_{t_j} \cap \operatorname{Preds}_j), \hat{\mathcal{P}})}{P(u_j \in U^D_\infty | u_i \in (U^D_{t_j} \cap \operatorname{Preds}_j), \hat{\mathcal{P}})} \\
&= \frac{\hat{p}_{i,j}}{\hat{P}^D_j}
\end{aligned}$$

Nous notons $\hat{P}^D_{i\to j}$ cette valeur. Rappelons bien ici que P^D_j désigne la probabilité qu'*au moins une* tentative de transmission vers u_j dans D ait réussi (équation 3.1), alors que $P^D_{i\to j}$ désigne la probabilité que *la* tentative de transmission depuis u_i vers u_j dans D ait réussi.

Nous pouvons alors calculer *l'espérance* de la vraisemblance d'un ensemble de paramètres $\mathcal{P}$ connaissant les données complétées $(\mathcal{D}, \mathcal{X})$ et une estimation courante $\hat{\mathcal{P}}$:

$$\begin{aligned}
\mathcal{Q}(\mathcal{P}|\hat{\mathcal{P}}) &= \mathbb{E}_{\mathcal{X}}\left[\mathcal{L}\left(\mathcal{P}; (\mathcal{D}, \mathcal{X})\right) | \hat{\mathcal{P}}\right] \\
&= \sum_{D\in\mathcal{D}} \left(\Phi^D(\mathcal{P}|\hat{\mathcal{P}}) + \sum_{u_j \in \bar{U}^D_\infty} \sum_{u_i \in U^D_\infty} \log(1 - p_{i,j}) \right)
\end{aligned} \tag{3.5}$$

avec :

$$\Phi^D(\mathcal{P}|\hat{\mathcal{P}}) = \sum_{u_j \in U^D_\infty} \sum_{u_i \in (U^D_{t_j} \cap \operatorname{Preds}_j)} \left(\hat{P}^D_{i\to j} \log(p_{i,j}) + \left(1 - \hat{P}^D_{i\to j}\right) \log(1 - p_{i,j}) \right) \tag{3.6}$$

Remarquons la similarité entre les équations 3.5 et 3.3. Dans [Dempster et al., 1977], il est montré que la suite $\mathcal{P}^{(n+1)} = \arg\max_{\mathcal{P}} \mathcal{Q}(\mathcal{P}|\mathcal{P}^{(n)})$ converge vers un maximum local quand n augmente.

Annuler la dérivée de $\mathcal{Q}(\mathcal{P}|\hat{\mathcal{P}})$ par rapport aux paramètres $\mathcal{P}$ nous permet de maximiser l'espérance $\mathcal{Q}$ à chaque itération de l'algorithme EM. Pour chaque lien $(u_i, u_j) \in E$, nous

obtenons la formule de mise à jour :

$$p_{i,j} \leftarrow \frac{\sum_{D \in \mathcal{D}^?_{i,j}} \frac{\hat{p}_{i,j}}{\hat{P}^D_j}}{|\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}|} \tag{3.7}$$

avec :

— $\mathcal{D}^?_{i,j}$: ensemble des épisodes de diffusion où il est **possible** que u_i ait contaminé u_j, c'est à dire où u_i est infecté avant u_j :

$$\mathcal{D}^?_{i,j} = \{D \in \mathcal{D} | (u_i \in U^D_\infty) \wedge (u_j \in U^D_\infty) \wedge (t^D_i < t^D_j)\}$$

— $\mathcal{D}^-_{i,j}$: ensemble des épisodes de diffusion où il est **impossible** que u_i ait réussi à contaminer u_j, c'est à dire où u_i est infecté et u_j ne l'est pas. Les épisodes de $\mathcal{D}^-_{i,j}$ sont appelés des « contre-exemples » pour le couple d'utilisateurs (u_i, u_j).

$$\mathcal{D}^-_{i,j} = \{D \in \mathcal{D} | (u_i \in U^D_\infty) \wedge (u_j \notin U^D_\infty)\}$$

— $\hat{P}^D_j$: l'estimation courante de P^D_j calculée selon l'équation 3.1 avec les valeurs courantes $\hat{p}_{i,j}$.

La démonstration de la formule 3.7 est donnée en annexe A, et l'algorithme 1 résume l'ensemble de la procédure d'apprentissage. Remarquons enfin que la formule de mise à jour 3.7 peut être comprise intuitivement ainsi : si nous connaissions $\mathcal{X}$, l'estimation de $\mathcal{P}$ aurait la forme :

$$p_{i,j} = \frac{\sum_{D \in \mathcal{D}^?_{i,j}} X^D_{i,j}}{|\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}|} \tag{3.8}$$

En d'autres termes, il suffirait de diviser le nombre de fois où u_i a réussi à transmettre une information à u_j par le nombre de fois où il a essayé de le faire. L'information $\mathcal{X}$ étant manquante, la formule 3.7 est une estimation de la valeur de la formule 3.8.

Par rapport à [Saito et al., 2008], l'estimation de $p_{i,j}$ est similaire mais se base sur bien plus d'exemples, car elle considère beaucoup plus de cas comme étant des possibilités d'infection. Cela nous permet d'obtenir un modèle plus réaliste et robuste, tout en évitant les difficultés liées à l'apprentissage des délais de diffusion.

Remarquons enfin que la formule de mise à jour 3.7 fait que $\hat{p}_{i,j} > 0 \implies p_{i,j} > 0$. Il en découle, par récurrence, que les valeurs de $\mathcal{P}$ apprises par l'algorithme 1 ne sont jamais nulles, car elles sont initialisées aléatoirement sur l'intervalle $]0, 1[$.

Algorithme 1 : Delay-Agnostic IC (DAIC)

Entrées :
U : Ensemble d'utilisateurs ;
$\mathcal{D}$: Ensemble d'épisodes de diffusion d'apprentissage ;
M : Nombre d'itérations
Sorties :
$\mathcal{P} = (p_{i,j})_{(u_i,u_j)\in U^2}$;

pour $(u_i, u_j) \in U^2$ **faire**
  $\hat{p}_{i,j} = 0$;
  si $|\mathcal{D}^?_{i,j}| > 0$ **alors**
    Initialiser $\hat{p}_{i,j}$ au hasard dans $]0, 1[$;
  fin
fin
$it \leftarrow 0$;
tant que $it < M$ **faire**
  pour (u_i, u_j) *tel que* $|\mathcal{D}^?_{i,j}| > 0$ **faire**
    $p_{i,j} \leftarrow \frac{\sum_{D\in\mathcal{D}^?_{i,j}} \frac{\hat{p}_{i,j}}{\hat{P}^D_j}}{|\mathcal{D}^?_{i,j}|+|\mathcal{D}^-_{i,j}|}$
  fin
  $\hat{\mathcal{P}} \leftarrow \mathcal{P}$
  $it \leftarrow it + 1$
fin
retourner $\hat{\mathcal{P}}$

3.3 Régularisation de l'apprentissage

3.3.1 Biais d'apprentissage

Dans l'algorithme d'apprentissage que nous avons présenté, les probabilités de transmission sont apprises en maximisant la vraisemblance de l'ensemble d'apprentissage, c'est à dire en cherchant les probabilités expliquant au mieux les épisodes de diffusion observés, en suivant la méthodologie définie dans [Saito et al., 2008].

Cela a pour conséquence d'introduire un biais dans l'apprentissage, lié à l'hétérogénéité des fréquences d'apparition des utilisateurs dans l'ensemble d'apprentissage. En effet, nous pouvons voir qu'avec la formule 3.7, des paires d'utilisateurs avec peu (ou pas) de « contre-exemples » dans l'ensemble d'apprentissage risquent de masquer les contaminations dans d'autres épisodes. Un « contre-exemple » pour un paramètre $p_{i,j}$ est un épisode de diffusion D dans lequel u_i est infecté et u_j ne l'est pas, ce qui signifie que u_i n'a pas réussi à contaminer u_j. L'ensemble $\mathcal{D}^-_{i,j}$ correspond à l'ensemble de ces contre-exemples.

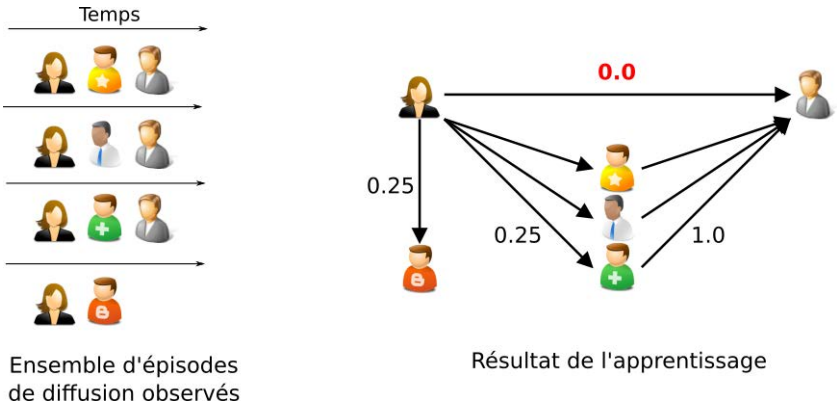


FIGURE 3.3 – Illustration du problème de biais d'apprentissage

.

Soit $p_{i,j}^{(n)}$ l'estimation de $p_{i,j}$ à la n-ième itération de l'algorithme EM. Nous avons la proposition suivante :

Proposition 1. *Pour tout lien* $(u_i, u_j) \in E$ *tel que* $|\mathcal{D}_{i,j}^-| > 0$, *s'il existe pour chaque épisode* $D \in \mathcal{D}_{i,j}^?$ *un utilisateur* $u_k \in U_{t_j}^D \cap Preds_j$ *tel que* $|\mathcal{D}_{k,j}^-| = 0$, *alors :*

$$\lim_{n \to +\infty} p_{i,j}^{(n)} = 0$$

La démonstration de cette proposition se trouve en annexe B.

Une illustration de la proposition 1 est donnée en figure 3.3. Dans cette figure, nous pouvons voir plusieurs exemples positifs entre l'utilisatrice noire et l'utilisateur gris (les trois premiers épisodes de diffusion) ainsi qu'un contre exemple de diffusion pour ce couple d'utilisateurs. Par contre, pour les couples d'utilisateurs jaune-gris, blanc-gris et vert-gris, il n'existe aucun contre-exemple de diffusion : chaque fois que le premier utilisateur est infecté, le second l'est aussi. Cet ensemble d'épisodes de diffusion conduit à l'apprentissage des probabilités représentées à droite. On constate que la probabilité de transmission de l'utilisatrice noire à l'utilisateur gris est nulle (ou, plus exactement, tend vers 0). L'algorithme d'apprentissage considère en fait que les infections de l'utilisateur gris dans les épisodes de diffusion observés sont parfaitement expliquées par des probabilités de transmissions à 1 pour les paires jaune-gris, blanc-gris et vert-gris, et de 0 (une valeur arbitrairement faible) pour noire-gris. Ainsi, des utilisateurs rares (jaune, blanc et gris) ont complètement masqué la relation entre noire et gris, pourtant bien plus présents dans l'ensemble d'apprentissage. Sur cet exemple théorique, cela n'est pas forcément gênant. Mais en pratique, les données extraites de corpus réels sont très bruitées, et les utilisateurs ont des comportements très chaotiques. Les utilisateurs rares peuvent donc correspondre à du bruit dans les données. Ce phénomène devient alors problématique, car il limite les capacités de généralisation du modèle.

Notons enfin que ce problème est également présent dans l'algorithme original de [Saito et al., 2008], mais de façon moins prononcée car celui-ci considère beaucoup moins d'infecteurs potentiels.

3.3.2 Maximum a posteriori

Pour résoudre ce problème, nous proposons d'ajouter un a priori sur les probabilités de transmission apprises. Le problème s'écrit alors sous la forme d'un maximum a posteriori :

$$\begin{aligned}\mathcal{P}^\star &= \arg\max_{\mathcal{P}} \prod_{D\in\mathcal{D}} P(U_\infty^D|\mathcal{P}) \prod_{p_{i,j}\in\mathcal{P}} f(p_{i,j}) \\ &= \arg\max_{\mathcal{P}} \mathcal{L}(\mathcal{P};\mathcal{D}) + \sum_{p_{i,j}\in\mathcal{P}} \log f(p_{i,j})\end{aligned} \tag{3.9}$$

où f est l'*a priori* appliqué aux probabilités de transmission. Plusieurs fonctions sont envisageables, nous proposons d'utiliser une loi exponentielle car celle-ci favorise les solutions parcimonieuses et les probabilités de transmission faibles. En effet, comme nous l'avons vu dans le chapitre 2, la diffusion est un phénomène *rare*. Il est donc peu vraisemblable d'avoir des probabilités de transmission élevées sur de nombreuses relations. Avec une distribution exponentielle $f(p_{i,j}) = \lambda e^{-\lambda p_{i,j}}$, le problème se simplifie facilement en :

$$\mathcal{P}^\star = \arg\max_{\mathcal{P}} \left(\mathcal{L}(\mathcal{P};\mathcal{D}) - \lambda \sum_{p_{i,j}\in\mathcal{P}} p_{i,j} \right) \tag{3.10}$$

Il s'agit donc d'une régularisation $\ell 1$ des probabilités apprises.

En reprenant la méthode décrite dans la section précédente, nous obtenons à chaque étape de maximisation de l'algorithme EM l'équation polynomiale suivante pour chaque paramètre $p_{i,j}$, que nous devons résoudre pour maximiser $\mathcal{Q}(\mathcal{P}|\hat{\mathcal{P}})$:

$$\lambda p_{i,j}^2 - \beta p_{i,j} + \gamma = 0 \tag{3.11}$$

avec :

$$\beta = |\mathcal{D}_{i,j}^?| + |\mathcal{D}_{i,j}^-| + \lambda \tag{3.12}$$

$$\gamma = \sum_{D\in\mathcal{D}_{i,j}^?} \frac{\hat{p}_{i,j}}{\hat{P}_j^D} \tag{3.13}$$

La démonstration de ce résultat est donnée en annexe C.

Ce polynôme permet de déduire la nouvelle formule de mise à jour des paramètres pour l'algorithme EM :

$$p_{i,j} \leftarrow \frac{\beta - \sqrt{\Delta}}{2\lambda} \tag{3.14}$$

La démonstration de de la validité de cette formule de mise à jour est donnée en annexe D.

L'utilisation d'un a priori de loi exponentielle pour les probabilités de transmission permet d'éviter que les relations peu observées convergent vers des probabilités trop élevées, ce qui limite le problème du biais : les utilisateurs rares pèsent en effet moins sur l'apprentissage du modèle. Néanmoins nous avons observé dans nos expériences préliminaires que les probabilités apprises étaient finalement trop faibles et conduisaient à des épisodes de diffusion prédits trop courts. Nous proposons donc comme heuristique, après l'apprentissage de la version régularisée, d'effectuer *une* itération de l'algorithme EM normal, afin d'obtenir des probabilités de transmission plus élevées tout en évitant les problèmes du biais d'apprentissage.

3.4 Expériences

Dans cette section, nous évaluons notre modèle, DAIC, en le comparant à diverses approches issues de l'état de l'art.

3.4.1 Modèles de Référence

Nous comparons notre approche DAIC à un modèle IC appris selon la précédure classique décrite dans [Saito et al., 2008], ainsi qu'aux modèles NetRate [Gomez-Rodriguez et al., 2011] et CTIC [Saito et al., 2009] décrits plus haut.

De plus, comme nous l'avons expliqué au début du chapitre, nous nous plaçons dans ce manuscrit dans le contexte d'un réseau social dont le graphe est inconnu ou inexistant. Toutefois, les modèles considérés ici (y compris DAIC) restent valides lorsqu'ils sont appliqués au graphe *complet*, reliant tous les utilisateurs entre eux. C'est ce que nous faisons dans ce chapitre et dans tout ce manuscrit. Cela rend l'apprentissage plus long, mais il est possible de l'accélérer on considérant uniquement les liens (u_i, u_j) telles que $|D^?_{i,j}| > 0$, les autres ayant nécessairement une probabilité de transmission de 0 à l'issue de l'apprentissage [Gomez-Rodriguez et al., 2011]. Cette propriété est prise en compte dans l'algorithme 1 en lignes 3 et 9.

3.4.2 Expériences sur des données synthétiques

Pour analyser les performances des différentes approches, nous commençons par effectuer des expériences sur des jeux de données artificiels.

3.4.2.1 Génération des corpus synthétiques

Notre but dans ces expériences est de comprendre comment les différents modèles se comportent par rapport à la variabilité des délais entre deux infections successives.

Nous commençons donc par générer des épisodes de diffusion artificiels sur un réseau invariant d'échelle de 100 utilisateurs, construit avec le modèle de Barabási-Albert [Albert and Barabási, 2002]. Les probabilités de transmission sont générées au hasard, uniformément sur l'intervalle $[0, 1]$. Ce graphe est utilisé pour générer des épisodes de diffusion mais n'est *pas* utilisé pendant l'apprentissage, pour respecter le contexte expérimental fixé. Nous générons des épisodes de diffusion sur ce graphe en tirant un ensemble de sources (de 1 à 3 utilisateurs) avant d'effectuer une simulation de diffusion en utilisant une variante du modèle IC : lorsqu'un utilisateur infecté au temps t contamine un de ses voisins, ce voisin devient infecté après un délai $\delta^D_{i,j}$, qui est tiré pour chaque paire d'utilisateur et chaque épisode de diffusion :

$$\delta^D_{i,j} = 1 + \gamma_{i,j} + \xi^D_{i,j} \tag{3.15}$$

Les délais $\gamma_{i,j}$ et $\xi^D_{i,j}$ sont tirés selon des lois exponentielles de moyennes μ et σ :

$$\gamma_{i,j} \sim \frac{1}{\mu} e^{-\frac{x}{\mu}} \qquad \xi^D_{i,j} \sim \frac{1}{\sigma} e^{-\frac{x}{\sigma}} \tag{3.16}$$

La valeur μ nous permet donc de contrôler la variance des délais de transmission entre les différentes paires d'utilisateurs, alors que la valeur σ nous permet de contrôler celle de ces délais d'un épisode de diffusion à l'autre. Si, durant la génération des données, un délai de transmission trop long est tiré (i.e. conduisant à une contamination après un horizon temporel fixé à 1000), la contamination correspondante est ignorée et n'est pas inclue dans l'épisode de diffusion généré.

Notons que nous avons également envisagé d'autres méthodes de construction du réseau social (comme utiliser un réseau réel) et de génération des épisodes de diffusion. Toutefois, nous n'avons pas observé de résultats significativement différents, car la principale différence entre les différents modèles étudiés est leur gestion de la dimension temporelle.

3.4.2.2 Évaluation

Nous comparons les probabilités de transmission $\mathcal{P}^\star$ apprises par les différents algorithmes à celles utilisées pour générer les données, notées $\mathcal{P}^t = (p^t_{i,j})_{(u_i,u_j)}$. Nous utilisons pour cela

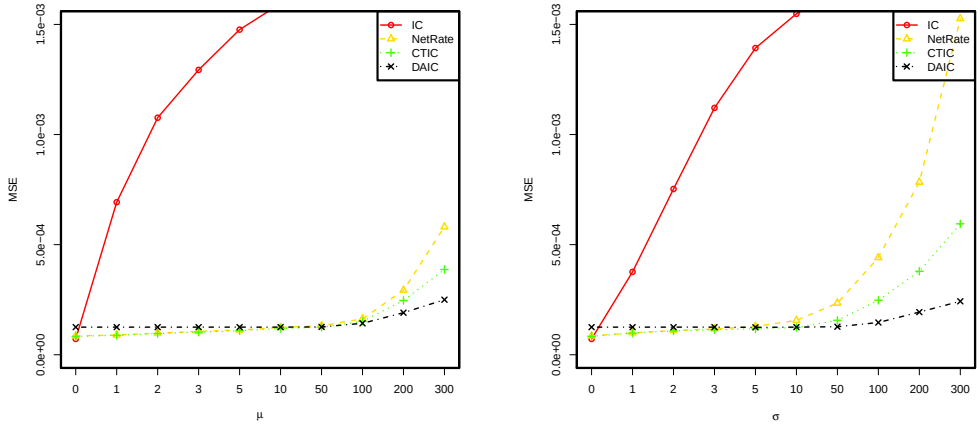


FIGURE 3.4 – MSE des probabilités de diffusion apprises $\mathcal{P}$ par rapport à $\mathcal{P}^\star$, pour différentes valeurs de μ et σ.

une mesure MSE (*Mean Squared Error*) calculée sur l'ensemble des probabilités :

$$MSE = \frac{1}{N \times (N-1)} \sum_{(u_i,u_j) \in U^2, u_i \neq u_j} (p^t_{i,j} - p^\star_{i,j})^2$$

3.4.2.3 Résultats

La figure 3.4 présente les scores de MSE obtenues par les modèles *IC*, *NetRate*, *CTIC* et *DAIC* sur les corpus artificiels. Dans la figure de gauche, nous étudions l'impact de la variance des délais de diffusion entre les paires (paramètre μ) pour une valeur de σ fixée à 10^{-5}. Une faible valeur de σ indique que les délais de diffusion sont stables d'un épisode de diffusion à l'autre. À l'inverse, sur la figure de droite, nous évaluons l'impact de la variance des délais de diffusion entre épisodes (paramètre σ) pour une valeur de μ fixée à 10^{-5}. Pour chaque configuration, les résultats sont moyennés sur 10 corpus de 1000 cascades.

Lorsque μ et σ tendent vers 0 (coin inférieur gauche de chaque figure), les délais de transmission tendent vers 1, c'est à dire qu'un utilisateur infecté contamine ses voisins au pas de temps suivant, ce qui correspond au cadre du modèle IC classique. Et effectivement, dans ce cas, nous pouvons voir que le modèle IC obtient de meilleures performances, car son algorithme d'apprentissage est justement restreint aux délais de diffusion de 1. Notre modèle, DAIC, considère qu'un utilisateur infecté peut avoir été contaminé par n'importe quel utilisateur infecté avant lui, ce qui ne correspond pas à cette configuration. Les modèles NetRate et CTIC sont plus souples que le modèle IC, mais considèrent tout de même que les délais de transmission plus courts sont plus vraisemblables, et obtiennent donc dans ce contexte une MSE meilleure que celle de DAIC.

En revanche, tout change lorsque la valeur de μ ou de σ augmente. La MSE du modèle IC monte très rapidement, car les délais de diffusion deviennent plus longs alors qu'IC est incapable de prendre en compte l'influence d'un utilisateur sur un autre pour un délai supérieur à 1. Les modèles NetRate, CTIC et DAIC ne sont pas touchés par ce problèmes et conservent une meilleure MSE.

Sur les courbes de gauche, nous pouvons voir que CTIC se comporte mieux que NetRate par rapport aux variations des délais entre les différents liens. CTIC considère en effet les délais et les probabilités de transmission de façons indépendantes, ce qui lui permet d'inférer de bonnes probabilités de transmission même pour des paires d'utilisateurs ayant de longs délais de transmission, contrairement à NetRate. Nous pouvons également voir, sur les courbes de droite, que CTIC est plus robuste que NetRate aux variations des délais entres les épisodes de diffusion.

Sur les deux ensembles de courbes, nous pouvons voir qu'à mesure que μ et σ augmentent, notre modèle DAIC obtient de meilleurs résultats que les modèles de référence. Il est plus robuste aux variations des délais puisqu'il ne les prend pas du tout en compte durant son apprentissage.

L'augmentation de la MSE pour les valeurs de μ et σ supérieures à 100 peut être en partie expliquée par le fait qu'à partir de ces valeurs, les épisodes de diffusion deviennent plus courts car certaines infections sont générées au delà de l'horizon temporel, et donc ignorées. Cela masque certaines contaminations et réduit la quantité de données disponibles pour l'apprentissage. Une autre explication possible est qu'avec des délais élevés, les délais $\delta_{i,j}^D$ peuvent devenir suffisamment longs pour qu'un utilisateurs infecté tard puisse avoir été contaminé par n'importe quel utilisateur précédent. Cela rend l'apprentissage plus difficile, mais correspond bien à l'hypothèse sur laquelle est basée notre modèle DAIC, qui obtient donc une meilleure MSE.

3.4.3 Expériences sur des données réelles

3.4.3.1 Tâche de prédiction de diffusion

Les modèles considérés dans cette section apprennent des probabilités de transmission et/ou des délais de transmission entre les utilisateurs d'un réseau social. En pratique, nous ne connaissons malheureusement pas les « vraies » valeurs de ces paramètres. Il nous est donc impossible d'évaluer ces modèles en comparant les valeurs apprises à celles d'une « vérité-terrain ».

Nous évaluons donc ces modèles sur une tâche de prédiction similaire à celles décrites dans le chapitre 2, section 2.5 : notre but est de retrouver l'ensemble des utilisateurs infectés dans un épisode de diffusion de test à partir des utilisateurs initiaux. Pour cela, les modèles appris sont utilisés en simulation pour prédire un ensemble d'utilisateurs finaux.

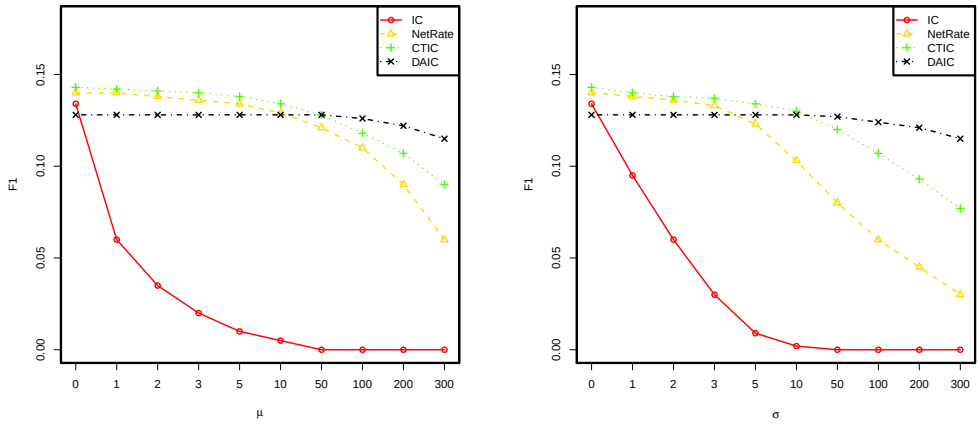


FIGURE 3.5 – Score $F1$ des différents modèles en prédiction de diffusion sur les données artificielles, pour différentes valeurs de μ et σ.

À chaque simulation, les résultats obtenus sont évalués avec une mesure de précision et une mesure de rappel. La précision est le taux d'utilisateurs prédits comme infectés faisant effectivement partie de U_∞^D. Le rappel est le taux d'utilisateur de U_∞^D infectés dans la simulation. Puis, chaque modèle est évalué à chaque simulation avec une mesure $F1$:

$$F1 = \frac{2 \times \text{Precision} \times \text{Rappel}}{\text{Precision} + \text{Rappel}}$$

Cette mesure est moyennée sur l'ensemble des simulations et l'ensemble des épisodes de diffusion de test.

Remarquons que dans le modèle NetRate, chaque utilisateur infecté finit forcément par contaminer l'ensemble de ses voisins, à mesure que le temps écoulé tend vers l'infini. La simulation de diffusion selon ce modèle est donc effectuée en posant un horizon temporel $T_{\max}$. Tout utilisateur infecté après cet horizon est considéré comme non infecté. Cet horizon temporel est égal à celui observé dans les épisodes de diffusion d'apprentissage.

La figure 3.5 présentent les scores $F1$ obtenus par les différents modèles en prédiction de diffusion, pour plusieurs valeurs de μ et σ de la même façon que dans la figure précédente. Les modèles sont appris sur des jeux de données de 1000 épisodes de diffusion, et testé sur des ensembles de test de la même taille. Nous pouvons constater que les différentes observations faites sur la figure 3.4 sont également applicables à la figure 3.5. Cela confirme l'idée selon laquelle l'évaluation des modèles en prédiction de diffusion permet de rendre compte de la qualité des probabilités de transmission apprises.

3.4.3.2 Corpus réels

Nous utilisons dans nos expériences cinq jeux de données issus de divers sites internet. Nous extrayons de chaque corpus des épisodes de diffusion, avec une méthode dépendant du fonctionnement du site utilisé.

Digg : Digg est un portail d'information en ligne [6]. Les utilisateurs de ce site peuvent y partager des articles où des vidéos issus de diverses sources, et attribuer des « digg » aux contenus qu'ils ont appréciés. Les différents contenus partagés apparaissent ensuite sur la page d'accueil ou sur d'autres pages de Digg, suivant leur popularité. Nous avons utilisé l'API de Digg pour récupérer l'historique *complet* du site sur une période d'un mois. Nous en avons extrait des épisodes de diffusion en considérant que chaque contenu partagé sur Digg correspondait à une information, et que chaque « digg » constituait une infection d'un utilisateur.

ICWSM : En 2009, à l'occasion de la conférence ICWSM (*International AAAI Conference on Weblogs and Social Media*), un corpus de 44 millions de posts de blogs avait été publié. Nous en extrayons des épisodes de diffusion en considérant des ensembles de posts se citant les uns les autres par le biais d'hyperliens. Les auteurs de chaque groupe de posts ainsi reliés sont considérés comme infectés par une même information.

Enron : Ce corpus est composé d'emails échangés par environ 150 personnes, principalement des managers d'Enron American Corporation. Nous considérons dans ce corpus que chaque adresse email correspond à un utilisateur. Nous formons des épisodes de diffusion en suivant la méthode décrite dans [Klimt and Yang, 2004], en considérant des séquences de messages formant des conversations. Ces conversations sont extraites en sélectionnant des messages contenant au moins deux mots en communs et dont l'expéditeur est le récepteur d'un message précédent dans la séquence.

Twitter : Ce corpus a été construit en utilisant l'API de Twitter. Nous avons commencé par récupérer une liste de 5000 utilisateurs ayant utilisé les hashtag *#obama*, *#romney* ou *#us2012*, durant la campagne présidentielle américaine de 2012. Puis, nous avons capturé l'intégralité de leurs messages sur une période de deux semaines. Des épisodes de diffusion ont ensuite été extraits en formant des séquences de tweets contenant un même hashtag. Les hashtags utilisés moins de cinq fois ont été ignorés.

Memetracker : Le corpus memetracker a été décrit dans [Leskovec et al., 2009]. Les épisodes représentent la diffusion de petites citations sur un large ensemble de sites d'information et de blogs durant la campagne présidentielle américaine de 2008.

Chaque corpus a été filtré pour garder une sous-population d'utilisateurs les plus actifs. La table 3.1 donne quelques statistiques sur les corpus utilisés : nombre d'utilisateurs, nombres d'épisodes de diffusion et taille moyenne de ces épisodes.

6. `www.digg.com`

	$\|U\|$	$\|D\|$	$\sum_{D\in\mathcal{D}} \frac{\|U^D_\infty\|}{\|\mathcal{D}\|}$
Digg	4587	20172	8.26
ICWSM	2270	20027	2.21
Enron	1557	1867	3.30
Twitter	4165	4815	22.54
Memetracker	30907	6724	20.21

TABLE 3.1 – Statistiques sur les corpus utilisés.

Les épisodes de diffusion extraits de chaque corpus sont repartis en un ensemble d'apprentissage, un ensemble de validation et un ensemble de test au moyen d'un tirage aléatoire sur l'ensemble des épisodes. La même procédure sera utilisée dans les chapitres suivants.

3.4.3.3 Résultats

	Digg	ICWSM	Enron	Twitter	Memetracker
IC	0.036	0.097	0.033	0.013	0.012
$NetRate$	0.102	0.358	0.105	0.027	0.048
$CTIC$	0.119	0.482	0.132	0.032	0.061
$DAIC_0$	**0.127**	**0.665**	**0.162**	0.026	**0.073**
$DAIC_5$	**0.128**	**0.665**	**0.164**	0.035	**0.087**
$DAIC_{10}$	**0.127**	**0.665**	**0.164**	**0.044**	**0.082**

TABLE 3.2 – Mesure F1 obtenues par les modèles sur les corpus réels. Les scores en gras sont significativement meilleurs que ceux de CTIC (selon un test de Student à 99%).

La table 3.2 indique les résultats obtenus par les modèles. Chaque résultat indique le score $F1$ moyen obtenu sur 1000 épisodes de diffusion de test. Nous utilisons la notation $DAIC_\lambda$ pour désigner notre modèle, λ étant le paramètre de la loi exponentielle utilisée comme a priori dans l'équation 3.10.

Pour apprendre le modèle IC de façon classique, nous devons définir un pas de temps, ce qui est délicat avec des épisode de diffusion réels (cf. chapitre 3). Après quelques expériences préliminaires, nous avons décidé d'utiliser comme pas de temps le délai moyen entre deux infections successives dans les épisodes de diffusion de l'ensemble d'apprentissage. Cette heuristique nous permet d'obtenir un nombre raisonnable d'exemples positifs pour chaque paire d'utilisateurs. Toutefois, les résultats obtenus par IC, présentés dans la table 3.2, montrent que ce modèle ne parvient pas à apprendre des probabilités de transmission pertinentes et obtient un score $F1$ bien inférieur à celui des autres approches. Ce score s'approche même de 0 sur Twitter et Memetracker, ce qui indique que les délais de transmission sur ces corpus sont trop variables pour trouver un pas de temps acceptable.

À l'exception du corpus Twitter, notre approche DAIC obtient des résultats significativement meilleurs que les autres. Cela confirme notre idée selon laquelle l'infection d'un utilisateur peut être expliquée par n'importe quel autre utilisateur infecté, et que les délais de transmission suivent une loi proche de la loi uniforme.

Le modèle CTIC, en faisant l'hypothèse que les délais plus courts sont plus vraisemblables, concentre son apprentissage sur certains liens et perd en capacité de prédiction. De plus, les utilisateurs apparaissant rarement dans $\mathcal{D}$ ont un impact négatif sur la pertinence des probabilités apprises.

Bien que notre approche ne puisse pas prédire les temps d'infection comme NetRate et CTIC, elle nous permet de mieux identifier les principaux liens empruntés par l'information. En considérant uniquement *l'ordre* dans lequel les utilisateurs ont été infectés, sans chercher à modéliser les temps d'infection, notre approche se concentre sur la prédiction de l'information de *qui infecte qui*, sans favoriser une source plutôt qu'une autre sur la base de son temps d'infection.

Sur le corpus Twitter, la possibilité pour un utilisateur infecté d'avoir été contaminé par n'importe quelle autre semble mener au problème de biais d'apprentissage décrit en section 3.3.1. En effet, sur ce corpus, beaucoup d'épisodes de diffusion contiennent des utilisateurs rares, ce qui conduit à une faible capacité de généralisation. Utiliser un a priori, comme proposé dans la formule 3.10, nous permet toutefois d'améliorer grandement les résultats de DAIC sur ce corpus. Plus généralement, sur les corpus contenant des épisodes de diffusion assez longs (Twitter et Memetracker), l'impact des utilisateurs rares est plus élevé, et c'est sur ces corpus que l'utilisation de la régularisation permet d'améliorer nos résultats.

Notons enfin que la valeur optimale de λ dépend du corpus considéré : 10 sur Twitter au lieu de 5 sur Memetracker. En pratique, cette valeur peut être fixée au moyen d'un processus de validation croisée.

3.5 Conclusion

L'apprentissage de modèles explicatifs de diffusion est une tâche difficile, en particulier avec des données issues de réseaux sociaux en ligne, particulièrement bruitées, parcimonieuses et partielles. Nous avons vu dans le chapitre 2 divers modèles explicatifs pour la diffusion d'information reposant sur différentes hypothèses. Malheureusement, des hypothèses trop fortes ou une modélisation trop fine peuvent se heurter à la qualité des données d'apprentissage disponibles. En particulier, nous avons présenté au début de ce chapitre les problèmes liés à la prise en compte du temps.

Dans ce chapitre, nous avons présenté deux contributions :

— nous avons proposé une version relaxée de l'algorithme d'apprentissage du modèle IC, qui considère l'ordre des infections plutôt que les temps exacts ;
— nous avons également décrit une méthode de régularisation permettant d'obtenir un modèle plus robuste, qui s'est révélée utile sur les jeux de données les plus grands et les plus bruités.

Les résultats obtenus sur des données artificielles et réelles ont montré la pertinence de cette approche, et ont confirmé notre idée selon laquelle la modélisation des délais de transmission complexifiait l'apprentissage des probabilités. Ces délais de transmission, en plus d'être très variables, ne sont pas toujours importants dans certaines applications. Ne pas les modéliser ne sera donc généralement pas considéré comme une limite d'un modèle.

Deuxième partie

Apprentissage de représentations pour la diffusion

Dans cette partie, nous proposons d'appliquer des méthodes d'apprentissage de représentations à la modélisation et à la prédiction de diffusion d'information.

Nous commençons par décrire la méthodes d'apprentissage de représentations et par présenter quelques travaux l'utilisant pour diverses tâches. Nous appliquons ensuite l'apprentissage de représentations à trois problèmes différents :

— l'apprentissage des probabilités de transmission du modèles IC ;
— la définition d'un modèle discriminant modélisant la diffusion d'information comme de la diffusion de chaleur ;
— le problème de la *détection de source*.

L'utilisation de l'apprentissage de représentations nous permet notamment de définir des modèles plus compacts et rapides.

Chapitre 4

Applications de l'apprentissage de représentations

Résumé Nous présentons dans ce chapitre quelques travaux utilisant l'apprentissage de représentations dans des contextes applicatifs variés. Ce rapide tour d'horizon nous permet de constater la diversité des applications possibles de ce type d'approche, ainsi que d'en dégager les principes généraux.

4.1 Introduction

L'apprentissage de représentations (*Representation Learning* ou RL) s'est largement développé au cours des dernières années, et a permis d'obtenir des résultats prometteurs sur diverses tâches mettant en jeu des dépendances relationnelles complexes [Bengio et al., 2013]. Le principe général des approches reposant sur l'apprentissage de représentations est le suivant :

> Projeter des éléments quelconques dans un *espace de représentation* (généralement $\mathbb{R}^d$) de façon à ce que les distances ou les similarités entre ces éléments *dans l'espace* modélisent une ou plusieurs relations existant entre eux *en dehors de cet espace*.

Les projections des éléments dans $\mathbb{R}^d$ sont nommées « représentations distribuées ». Ce type d'approche présente plusieurs avantages :

- définir une représentation *compacte* des données, ce qui peut être important pour certaines applications où l'utilisation des données brutes serait trop coûteuse ;
- *régulariser* les relations entre les éléments, des éléments similaires ou similaires aux même autres éléments étant projetés à des emplacements proches, ce qui peut permettre de découvrir certaines relations implicites.

Dans ce chapitre, nous présentons quelques applications de cette approche afin d'en dégager les grandes lignes. Nous présentons d'abords plusieurs articles étudiant la projection de

structures relationnelles simples ou complexes, puis nous nous intéressons à la prédiction de séquences.

4.2 Projection de structures relationnelles simples

4.2.1 Positionnement multidimensionnel

Historiquement, l'une des plus anciennes applications de l'apprentissage de représentations est celle du *positionnement multidimensionnel* (*Multidimensional Scaling* ou *MDS*) [Kruskal, 1964]. Étant donnée une matrice $M \in \mathbb{R}^{N \times N}$ décrivant les dissimilarités ou les distances entre N éléments, le but du MDS est de construire N représentations des éléments dans un espace multidimensionnel de façon à ce que les dissimilarités ou distances indiquées dans M soient le mieux respectées possible dans l'espace de représentation. Cela se fait en minimisant une fonction de coût nommée *stress* :

$$\text{Stress}_M(z_0, z_1, z_2....) = \sqrt{\sum_{i \neq j = 0,...,N-1} (M^{i,j} - ||z_i - z_j||)^2}$$

Ici, z_i désigne la projection du n-ième item. Cette mesure favorise donc les projections z telles que les distances entre les utilisateurs soient *proches* (au sens de la norme euclidienne) des distances indiquées dans M.

L'algorithme proposé pour minimiser le stress est une procédure itérative consistant à initialiser les z_i au hasard puis à les déplacer, à chaque itération, dans la direction du gradient du stress.

L'approche est validée sur divers ensembles de données réelles ou synthétiques. Les résultats sont évalués en observant la valeur finale de stress atteinte, ou en visualisant sur une courbes les couples de valeurs $(M^{i,j}, ||z_i - z_j||^2)$ pour tous les (i, j). Les auteurs effectuent également des expériences en « reconstruction ». Ils génèrent aléatoirement des valeurs de z_i, puis calculent à partir de celles-ci une matrice M, avec $M^{i,j} = ||z_i - z_j||$. Ils appliquent ensuite l'algorithme d'apprentissage à cette matrice M, et observent que leur méthode parvient à retrouver correctement les z_i originaux.

4.2.2 Graphe orienté

Le problème de projection d'un graphe orienté a été étudié dans [Chen et al., 2007]. Les auteurs posent la fonction de coût suivante :

$$\sum_{u_i} \left(\text{PageRank}(u_i) \sum_{u_j \in \text{Succs}(u_i)} p_{i,j} ||z_i - z_j||^2 \right) \tag{4.1}$$

$p_{i,j}$ désigne ici le poids du lien reliant l'utilisateur u_i à l'utilisateur u_j dans le graphe, et z_i est la projection de l'utilisateur u_i dans l'espace de représentation. Le PageRank (décrit en section 2.7.1) permet de calculer l'importance d'un utilisateur dans ce graphe. Cette fonction de coût exprime donc le fait que deux utilisateurs reliés dans le graphe devraient être plus proches l'un de l'autre dans l'espace de représentation, en particulier si u_i est un utilisateur « central » dans le graphe.

La méthode proposée pour optimiser cette fonction consiste à extraire les vecteurs propres d'une matrice calculée à partir de la matrice d'adjacence du graphe et du PageRank des utilisateurs. Cette méthode est testée sur un graphe de pages web extraites des sites internets de trois universités américaines, reliées par des liens hypertextes. Les pages d'un même site ont tendance à être plus densément reliées entre elles. Ce graphe est projeté dans un espace à deux dimensions et les auteurs observent que les points sont séparés en trois groupes correspondant effectivement aux trois universités. Ce résultat empirique est ensuite confirmé en effectuant des expériences en classification dans l'espace de représentation. Les performances en classification étant bonnes, les auteurs concluent que les proximités calculées dans l'espace de représentation sont pertinentes.

4.3 Projection de structures relationnelles complexes

L'apprentissage de représentations a également été utilisé pour projeter des éléments reliés entre eux par des relations explicites plus complexes : multigraphes, graphes hétérogènes, réseaux urbains, etc... Ces projections peuvent permettre de capturer certaines relations implicites ou de visualiser plus facilement un ensemble d'éléments liés.

4.3.1 Filtrage collaboratif

Nous avons parlé dans la sous-section 2.5.3 du *filtrage collaboratif*. Celui-ci considère une matrice M contenant les notes données par des utilisateurs à des produits. Cette matrice peut également être vue comme la matrice d'adjacence d'un graphe biparti ,valué, reliant des utilisateurs à des produits. Pour prédire de nouvelles relations, la méthode de factorisation matricielle consistant à factoriser $M \approx R_U \times R_I$ est une forme d'apprentissage de représentations où les utilisateurs et les produits sont projetés dans le même espace, de façon à ce qu'un utilisateur soit similaire aux produits auxquels il a attribué une bonne note.

Ces projections permettent ensuite de prédire la note qui serait attribuée par un utilisateur à un produit donné, et donc de trouver d'autres items susceptibles de l'intéresser.

4.3.2 Base de connaissances

L'article [Bordes et al., 2011] utilise l'apprentissage de représentations pour projeter des bases de connaissances complexes : WordNet (une grande collection de mots reliés par de nombreuses relations sémantiques) et Freebase (une vaste base structurée de connaissances variées). Ces bases prennent la forme d'ensembles de triplets du type *(entité1,relation,entité2)*. Par exemple, WordNet peut contenir le triplet ("voiture","instance_de","véhicule") et Freebase un triplet comme ("Marylin_Monroe","profession","actrice").

Les auteurs proposent une méthode de descente de gradient stochastique permettant d'apprendre non seulement une projection z_i pour chaque entité i, mais aussi une paire de matrices (R_1^r, R_2^r) pour chaque type de relation r, de façon à ce que la mesure :

$$S_r(z_i, z_j) = ||z_i.R_1^r - z_j.R_2^r||_1 \tag{4.2}$$

soit plus élevée pour les paires d'entités (z_i, z_j) pour lesquelles le triplet (z_i, r, z_j) existe dans la base considérée. Ainsi, chaque relation est associée à une certaine transformation de l'espace de représentation. Notons que le fait que d'utiliser *deux* matrices par relation permet de définir des mesures S_r asymétriques correspondant à des relations orientées. Le modèle est testé en répartissant les triplets en un ensemble d'apprentissage et un ensemble de test, et en observant la capacité du modèle appris à retrouver le troisième élément des triplets de l'ensemble de test à partir des deux autres.

Cette problématique a donné lieu à de nombreuses variantes de cette méthode, basées sur différentes représentations des relations. Par exemple, [Bordes et al., 2013] propose une autre façon de définir $S_r(z_i, z_j)$. Chaque relation r est cette fois-ci associée à une représentation ω_r située dans le même espace de représentation que les entités, et l'expression de S_r devient :

$$S_r(z_i, z_j) = ||(z_i + \omega_r) - z_j||_1$$

Chaque relation correspond donc à une *translation* des représentations des entités. Ce principe a ensuite été utilisé dans [Lin et al., 2015b] et [Wang et al., 2014]. Récemment, [Lin et al., 2015a] a proposé une méthode permettant de prendre le compte le fait que certaines relations peuvent s'additionner pour en obtenir de nouvelles. Par exemple, la relation indiquant dans quelle *ville* est née une personnalité peut s'ajouter à celle indiquant dans quel pays se trouve une ville pour obtenir la relation indiquant dans quel *pays* est née une personnalité.

4.3.3 Graphe hétérogène

Le même genre de méthode a été utilisée dans [Jacob et al., 2014] pour le problème de l'étiquetage de nœuds dans un graphe hétérogène. La tâche d'étiquetage de nœuds consiste à classifier les nœuds d'un graphe sur la base d'un ensemble de nœuds déjà étiquetés. Un

graphe hétérogène est un graphe contenant différentes *types* de nœuds, chaque type étant associé à un ensemble d'étiquettes possibles. Dans des graphes homogènes, ce problème est souvent résolu par des méthodes de « propagation d'étiquettes », qui consistent à poser comme contrainte qu'un nœud devrait avoir la même étiquette que ses voisins, ce qui n'est pas applicable directement dans un graphe hétérogène puisque les voisins d'un nœud n'ont pas forcément le même type et donc pas forcément des étiquettes « valides » Les auteurs proposent d'apprendre des représentations des nœuds en posant qu'un nœud doit avoir une représentation proche de celles de ses voisins, ce qui est représenté par un coût de la forme :

$$\sum_{i,j} w_{i,j} ||z_i - z_j||^2 \tag{4.3}$$

où $w_{i,j}$ est le poids du lien reliant le nœud i au nœud j, ou 0 si le lien n'existe pas. De plus, chaque étiquette k est associée à un classifieur f_{θ^k}. Les paramètres θ de ces classifieurs et les projections z des nœuds sont appris en même temps, en minimisant la somme d'un coût de classification et du coût de projection indiqué plus haut. Le coût de classification empêche que tous les éléments soient projetés au même point.

Le modèle est testé sur un corpus issu de DBLP, contenant des articles scientifiques et leurs auteurs. Les auteurs sont étiquetés avec leurs domaines de recherche, et les articles sont étiquetés avec les conférences dans lesquelles ils sont parus. Un corpus Flickr, contenant des images et des utilisateurs, est également utilisé. Chaque utilisateur est relié à ses photos et à ses amis (les relations sont donc elles-mêmes hétérogènes). Les photos sont étiquetées par des *tag* renseignés sur le site et les utilisateurs sont étiquetées par les *groupes* auxquels ils appartiennent. Le modèle appliqué à ces deux corpus obtient de meilleurs résultats (en classification) que les méthodes consistant à transformer le graphe hétérogène en plusieurs graphes homogènes (un par type de nœud) afin d'appliquer une propagation d'étiquettes classique sur chacun de ces graphes.

4.3.4 Images annotées

Enfin, l'apprentissage de représentations a aussi été appliqué à l'annotation d'images [Weston et al., 2011]. Les images sont décrites sous la forme de grands vecteurs de caractéristiques visuelles extraites des images. Les auteurs apprennent en même temps les paramètres θ d'un projecteur linéaire f_θ de ces images dans un espace de représentation de faible dimension, ainsi qu'une représentation z_i dans ce même espace pour chaque annotation (ou *tag*) i possible. En prédiction, le score d'un tag i pour une image donnée s'écrit :

$$s_i(\text{img}) = f_\theta(\text{img}).z_i \tag{4.4}$$

Ces paramètres sont appris en minimisant un coût d'ordonnancement, avec un algorithme stochastique.

Le modèle proposé est testé sur deux corpus d'images annotées issus du net, dont *ImageNet*, en étant comparé à des classifieurs multi-étiquettes classiques (un plus proche voisin et un séparateur à vaste marge). Leur méthode obtient de meilleurs résultats que les classifieurs classiques.

Ce modèle a ensuite été étendu dans [Gong et al., 2014], où les auteurs considèrent le cas où en plus des images et des tags, ils disposent également d'un a priori sur la *sémantique* de ces images et de ces tags. Ils proposent de projeter cette nouvelle information dans le même espace latent avec une fonction d'apprentissage similaire à [Weston et al., 2011] pour améliorer les capacités de généralisation du modèle.

Plus généralement, la problématique d'annotation d'images a donné lieu à une très abondante littérature ces deux dernières années, reposant souvent sur l'apprentissage de représentation : annotation par des mots-clefs, annotation par des phrases, génération de description à partir d'une image, etc.

4.4 Modélisation de séquences

L'apprentissage de représentations a également été appliqué à des données séquentielles, c'est à dire à des ensembles d'éléments ordonnées dans le temps. Dans ce contexte, l'apprentissage de représentations ne vise plus seulement à modéliser des relations *statiques* entres les éléments d'un ensemble, mais aussi des relations *temporelles*.

4.4.1 Modèles de langage

Un modèle de langage est un modèle définissant la probabilité d'observer une *phrase* quelconque dans un langage donné. Le modèle de langage le plus courant est celui dit des « *N-grammes* » consistant à définir la probabilité d'une phrase $S = (m_0, m_1, m_2...)$ comme la probabilité jointe de chaque *mot* de cette phrase conditionnellement aux N mots précédents :

$$P(S) = \prod_{m_i \in S} p(m_i | m_{i-1}, m_{i-2}, \ldots, m_{i-N+1})$$

L'ensemble des mots possibles est noté M. La distribution conditionnelle p est apprise statistiquement sur des corpus de plusieurs millions de phrases. Bien que cette méthode donne de bons résultats, sa complexité pose problème, le modèle devenant rapidement trop complexe à mesure que N augmente. De plus, la quantité de données nécessaires à l'apprentissage devient vite bien trop importante. En effet, le nombre de groupes de mots possibles est égal à $|M|^N$, et augmente donc de façon exponentielle avec N.

L'application de l'apprentissage de représentations aux modèles de langage consiste à définir la probabilité conditionnelle $p(m_i | m_{i-1}, m_{i-2}, \ldots, m_{i-N+1})$ comme une fonction

des représentations des mots concernés. Cela réduit grandement la complexité spatiale du problème, et permet d'utiliser une valeur de N plus grande. De plus, l'utilisation d'un espace de représentation améliore la capacité de généralisation du modèle. En effet, les mots similaires ont tendance à être projetés proches les uns des autres, et sont donc ensuite prédits dans les mêmes contextes. Ainsi, le modèle est capable d'associer une probabilité non-nulle à un mot m_i étant donné un contexte $m_{i-1}, m_{i-2}, \ldots, m_{i-N+1}$ même si aucun exemple d'apprentissage n'existe pour la suite de mots $m_i, m_{i-1}, m_{i-2}, \ldots, m_{i-N+1}$

Par exemple, dans [Bengio et al., 2006], les auteurs posent :

$$\begin{aligned} p(m_i|m_{i-1}, m_{i-2}, \ldots, m_{i-N+1}) &= f(i, m_{i-1}, m_{i-2}, \ldots, m_{i-N+1}) \\ &= g_\theta(i, z_{i-1}, z_{i-2}, \ldots, z_{i-N+1}) \end{aligned}$$

où z_i désigne la projection du mot m_i et g_θ est une fonction consistant à :

- concaténer les représentations distribuées des mots $z_{i-1}, z_{i-2}, \ldots, z_{i-N+1}$ en un seul grand vecteur d'entrée ;
- appliquer ce vecteur d'entrée à un réseau de neurones classique à une couche cachée utilisant l'ensemble de poids θ, donnant sur sa couche de sortie une valeur par mot du dictionnaire ;
- appliquer une fonction de type « softmax » sur les valeurs de sortie pour obtenir une distribution de probabilités, et renvoyer celle du i-ème mot.

Les représentations des mots et les paramètres θ du réseau de neurones sont appris en même temps, en maximisant la vraisemblance d'un grand corpus de textes (plus de 10 millions de mots) au moyen d'une montée de gradient stochastique parallélisée. L'approche obtient des résultats bien meilleurs qu'un modèle N-grammes classique.

Word2Vec Cette idée fut ensuite reprise dans les modèles *Word2Vec* [Mikolov et al., 2013a]. Le même principe général y est utilisé dans des contextes différents.

- D'une part, les auteurs proposent de prédire un mot quelconque d'un texte à partir des N mots le précédant et des N mots le suivant. Pour cela, les représentations de ces $2N$ mots sont moyennées, et la « représentation moyenne » obtenue est utilisée pour prédire le mot courant à partir d'un classifieur log-linéaire. Ce modèle est nommée *Continuous bag-of-words model* (CBOW)
- D'autre part, les auteurs s'intéressent au problème inverse : prédire les N mots suivants et les N mots précédents à partir d'un mot quelconque d'un texte. Ce modèle est baptisé *Skip-gram*.

Les auteurs remarquent que certaines relations syntaxiques et sémantiques se retrouvent dans l'espace de représentation appris. On observe par exemple que $z_{\text{France}} \approx z_{\text{Grèce}} - z_{\text{Athènes}} + z_{\text{Paris}}$. La même principe fonctionne pour d'autres relations, comme par exemple "frère-sœur" et "père-mère" ou "penser-pensant" et "lire-lisant". Les auteurs s'évaluent donc en particulier sur un ensemble de test constitué de quadruplets de mots comme ceux décrits

ci-dessus, et obtiennent une précision de 61% avec CBOW et 56% avec skip-gram, contre 47% pour un modèle de langage neuronal classique.

Cette correspondance entre d'une part les relations sémantiques ou syntaxiques existant entre les mots et d'autre part les relations algébriques existant entre leurs représentations constitue la principale force de l'apprentissage de représentations.

Les modèles de langage ainsi appris peuvent ensuite servir à des tâches comme la reconnaissance de parole [Schwenk, 2007, Graves et al., 2013] ou la traduction automatique [Cho et al., 2014].

4.4.2 Autres séquences

Les modèles de langage décrits dans la sous-section précédente peuvent en pratique être appliqués à n'importe quelles données prenant la forme de séquences de *symboles*. Plusieurs travaux ont ainsi utilisé des approches d'apprentissage de représentations appliquées à d'autres types de données séquentielles.

4.4.2.1 Prédiction de Playlists

Par exemple, dans [Chen et al., 2012], les auteurs étudient le problème de *génération de listes de lecture* sur les sites de streaming de musique. Le but est de générer automatiquement, à partir d'un point de départ (une chanson particulière), une séquence de pistes similaire aux listes de lecture créées par les utilisateurs. L'idée générale est la même que celle des modèles de langage, à savoir que la probabilité d'une séquence de musiques $S = (m_0, m_1, m_2...)$ vaut :

$$P(S) = \prod_{i>0} p(m_i|m_{i-1})$$

Au lieu d'utiliser un réseau de neurones, la probabilité p est définie avec une fonction softmax appliquées aux distances entre les représentations des musiques.

$$p(m_i|m_{i-1}) = \frac{\exp(||z_i - z_{i-1}||^2)}{\sum_{m_j \in M} \exp(||z_j - z_{i-1}||^2)}$$

M désigne ici l'ensemble des musiques. Les représentations de celles-ci sont apprises en maximisant la vraisemblance d'un ensemble de listes de lecture d'apprentissage. Les auteurs proposent également plusieurs extensions permettant d'intégrer certains données comme la popularité ou le genre musical des musiques. Ils visualisent les projections apprises sur un espace à deux dimensions et observent notamment que les chansons d'un même artiste ont tendance à être regroupées, ce qui est un résultat non-trivial puisque cette information n'est pas utilisée pendant l'apprentissage.

4.4.2.2 Recommandation séquentielle

Les modèles de prédiction de séquences décrits dans cette section ont aussi été appliqués à la recommandation dans un contexte séquentiel où l'utilisateur accède à des items les uns après les autres, la recommandation consistant à prédire le prochain item visité. Un exemple se trouve dans [Guàrdia-Sebaoun et al., 2015] : dans cet article, le formalisme de Word2Vec est utilisé pour apprendre des représentations distribuées des items à partir des séquences d'items vus par les utilisateurs. Celles-ci sont ensuite utilisées pour apprendre des représentations des utilisateurs, indiquant comment ceux-ci se « déplacent » d'un item à l'autre au sein de l'espace de représentation.

4.5 Conclusion

Dans ce chapitre, nous avons présenté de façon succincte quelques travaux appliquant des méthodes d'apprentissage de représentations à des tâches très variées. Cependant, et malgré la diversité des tâches, nous pouvons dégager plusieurs principes généraux.

- Le but est de projeter des données symboliques dans un espace de représentation continu $\mathbb{R}^d$. Ces projections peuvent être apprises *ad hoc* ou être issues d'une transformation depuis un autre espace. Les données projetées peuvent être homogènes ou hétérogènes (i.e tous les éléments peuvent être de même nature ou pas).
- Une ou plusieurs mesures définies sur $\mathbb{R}^d$ permettent de modéliser certaines relations existant entre les éléments projetés. Ces mesures peuvent être définies *a priori* (une distance ou une similarité) ou bien être elles-mêmes apprises (mesures paramétrées).
- L'apprentissage se fait généralement de façon stochastique, sur la base d'un ensemble d'exemples élémentaires prenant la forme de tuples d'éléments reliés par une certaine relation. La fonction de coût est généralement une somme sur l'ensemble de ces tuples (remarquons à ce propos la similitude entre les equations 4.1, 4.2, 4.3 et 4.4). L'apprentissage stochastique permet en outre d'utiliser de grands volumes de données.
- L'utilisation d'un espace de représentation permet de compresser l'information contenue dans ces exemples, et de prédire de nouvelles relations entres les éléments.

Dans ce manuscrit, nous appliquons une approche similaire à celles décrites ici à des problématiques liées à la diffusion d'information dans les réseaux sociaux. En effets, les épisodes de diffusion constituent des données séquentielles, et les réseaux sociaux en ligne peuvent être vus comme des graphes hétérogènes.

Projeter les utilisateurs dans un espace de représentation permet de retrouver diverses propriétés connues des réseaux sociaux : présences de communautés, faible diamètre, transitivité des relations, etc... En particulier, la transitivité des relations (« les amis de mes amis sont mes amis ») est naturellement traduite par l'inégalité triangulaire de la distance.

L'apprentissage de représentations nous permettra également de proposer des modèles plus rapides et d'étudier facilement des cas où le graphe du réseau social est inconnu.

Chapitre 5

Apprentissage de représentations pour le modèle IC

Résumé Ce chapitre décrit notre seconde contribution, publiée dans [Bourigault et al., 2016b]. Nous appliquons une méthode d'apprentissage de représentations à l'apprentissage des probabilités de transmission du modèle IC : les utilisateurs sont projetés dans un espace $\mathbb{R}^d$, et leurs représentations sont utilisées pour définir les probabilités de transmissions. Cette méthode nous permet d'obtenir des résultats légèrement supérieurs à ceux du chapitre 3.

5.1 Limites de l'apprentissage explicite des probabilités de transmission

Dans le chapitre 3, nous avons proposé une méthode d'apprentissage des paramètres du modèle IC, et nous nous sommes intéressés au cas où le graphe du réseau social était inconnu. Pour utiliser un modèle IC dans ce contexte, il est possible de réaliser une inférence de graphe (décrite dans la section 2.4.7 dans le chapitre 2) ou de considérer le graphe complet, éventuellement limité aux paires d'utilisateurs pour lesquelles au moins un exemple positif existe dans l'ensemble d'apprentissage. C'est ce que nous avons fait. Malheureusement, considérer le graphe complet du réseau implique une grande complexité spatiale, le modèle IC apprenant un paramètre indépendant pour *chaque* lien du graphe. De plus, utiliser le graphe complet revient à ignorer de nombreuses propriétés spécifiques des relations entre utilisateurs dans les réseaux sociaux : distribution des degrés en loi de puissance, faible diamètre, etc... [Mislove et al., 2007].

Une propriété importante de ces réseaux sociaux est la présence de *communautés*, que l'on peut définir comme des groupes d'utilisateurs similaires, plus densément connectés les uns aux autres et interagissant d'avantage entre eux. Récemment, [Barbieri et al., 2013a] a

montré que dans le cadre de la diffusion d'information, les communautés pouvaient non seulement être formées d'utilisateurs interagissant beaucoup les uns avec les autres (on parle de communautés *cohésives*), mais aussi d'utilisateurs interagissant avec les mêmes autres groupes d'utilisateurs (on parle alors de communautés *bimodales*). L'impact de ces communautés sur la diffusion d'information a été étudié dans [Barbieri et al., 2013a, Yang et al., 2014] :

— S'il y a diffusion entre les utilisateurs a et b d'une part, et entre les utilisateurs b et c d'autre part, alors il est vraisemblable qu'il y ait diffusion entre les utilisateurs a et c (communautés *cohésives*)

— S'il y a diffusion entre les utilisateurs a et c, entre les utilisateurs a et d et entre les utilisateurs b et c, alors il est vraisemblable qu'il y ait de l'influence entre les utilisateurs b et d (communautés *bimodales*)

Une illustration de ces principes est donnée en figure 5.1.

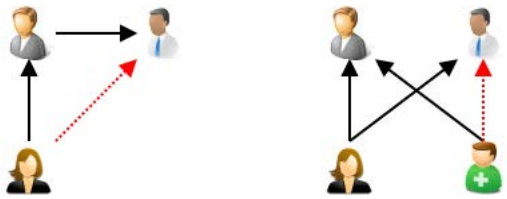


FIGURE 5.1 – Régularités sur les relations utilisateurs. Lorsque les liens en noir existent, les liens rouges sont plus vraisemblables.

Dans le modèle DAIC du chapitre 3 avec un graphe complet (ou dans l'apprentissage classique du modèle IC [Saito et al., 2008]), les probabilités de transmission sont estimées sans prendre en compte ces propriétés et leurs impact sur les probabilités de diffusion. Cela peut conduire à des structures de graphe de diffusion irréalistes, à causes de probabilités de transmission ne suivant pas les principes évoqués plus haut.

5.2 Projection du modèle IC

5.2.1 Apprentissage de Représentations

Pour résoudre les deux problèmes liés à l'apprentissage des probabilités de transmission (complexité spatiale et prise en compte des régularités des relations utilisateurs), nous proposons d'utiliser une approche basée sur l'*apprentissage de représentations*, pour modéliser les relations entre utilisateurs. Le principe est donc de projeter les utilisateurs dans un espace $\mathbb{R}^d$ de façon à ce que les distances entre eux permettent de modéliser les probabilités de transmission. La figure 5.2 illustre cette approche. L'utilisation d'un espace de représentation réduit considérablement le nombre de paramètres à apprendre et prend naturellement en compte les deux propriétés décrites plus haut. En particulier, l'expres-

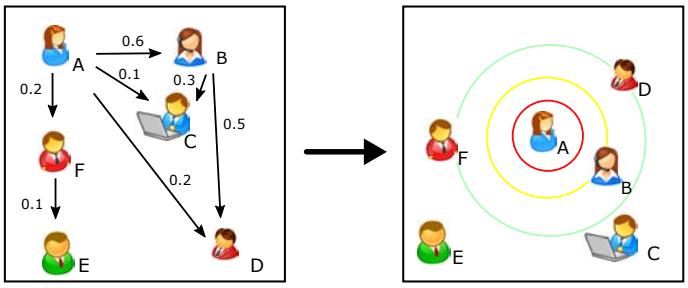


FIGURE 5.2 – Passage d'un graphe de diffusion à un espace de représentation vectoriel. À gauche, les valeurs associées aux liens représentent les probabilités de transmission entre utilisateurs. À droite, ces probabilités de transmission sont calculées en fonction de la distance séparant les utilisateurs. Les cercles réprésentent des lignes de niveau d'équiprobabilité depuis l'utilisatrice A.

sion « les amis de mes amis sont mes amis » est implicitement modélisée par l'inégalité triangulaire dans l'espace $\mathbb{R}^d$.

Notre problème diffère des approches de prédiction de séquences présentées dans le chapitre 4 par le fait que l'information se diffuse de manière arborescente, et non selon une séquence bien définie comme c'est le cas dans tous les domaines où des techniques de ce genre ont été employées. Cela complexifie le problème car, comme nous l'avons vu dans le chapitre 3, nous ne savons pas par qui chaque utilisateur d'un épisode de diffusion a été infecté.

Notre méthode peut se rapprocher de certaines approches probabilistes, en particulier celle de [Barbieri et al., 2013a]. Dans cet article, chaque utilisateur est associé à un ensemble de variables latentes traduisant son appartenance à des communautés. Ces variables latentes permettent de générer des liens et des épisodes de diffusion entre les utilisateurs. Ce modèle suppose toutefois la connaissance du graphe.

Une autre méthode de ce type, présentée dans le chapitre 2, se trouve dans [Guille and Hacid, 2012]. Là aussi, la diffusion d'un utilisateur à un autre dépend d'un ensemble de propriétés de ces utilisateurs. Ces variables sont toutefois mesurées et non pas apprises.

5.2.2 Formulation

Au lieu d'apprendre explicitement l'ensemble des probabilités de transmission $\mathcal{P}$, chaque utilisateur u_i est associé à deux représentations latentes z_i et ω_i dans l'espace $\mathbb{R}^d$. La projection z_i modélise le comportement de u_i en tant qu'*émetteur* de contenu, et ω_i son comportement en tant que *récepteur*. Les ensembles de projections sont notés $\mathcal{Z} = (z_i)_{u_i \in U}$ et $\Omega = (\omega_i)_{u_i \in U}$.

Nous proposons de définir les probabilités de transmission $p_{i,j}$ du modèle IC selon une fonction $f : \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$ de ces projections :

$$p_{i,j} = f(z_i, \omega_j) \tag{5.1}$$

Apprendre deux projections par utilisateur nous permet ainsi d'obtenir des probabilités de transmission asymétriques.

La fonction f peut être définie de multiples façons. Nous proposons de considérer la fonction logistique suivante :

$$f(z_i, \omega_j) = \frac{1}{1 + \exp(z_i^{(0)} + \omega_j^{(0)} + \sum_{x=1}^{d-1} (z_i^{(x)} - \omega_j^{(x)})^2)} \tag{5.2}$$

où $z^{(x)}$ est la x-ème composante du vecteur z. Le choix d'une fonction logistique permet de définir des probabilités décroissantes en fonction des distances séparant les représentations émetteur $\mathcal{Z}$ et récepteur Ω dans l'espace de projection. D'autre part, de par sa forme en S, l'utilisation de cette fonction implique un impact plus important des variations survenant sur les distances modérées, tombant dans la partie de plus forte pente de la fonction. Cela permet de focaliser l'attention sur les influences moins évidentes lors de l'apprentissage. À noter également que la fonction ainsi définie considère la première composante de chaque représentation comme une valeur de biais : $z_i^{(0)}$ et $\omega_j^{(0)}$ modélisent respectivement la tendance générale de u_i à transmettre de l'information et la tendance générale de u_j à devenir infecté.

Une fois cette fonction posée, nous pouvons reprendre le développement du modèle IC de la même façon que dans le chapitre 3, en remplaçant $p_{i,j}$ par $f(z_i, \omega_j)$ et sans nous limiter au graphe des « exemples positifs » (c'est à dire aux couples (u_i, u_j) tels que $|\mathcal{D}^?_{i,j}| > 0$). Ainsi, la probabilité d'un épisode de diffusion D s'écrit :

$$P(D|Z, \Omega) = \prod_{u_j \in U^D_\infty} P(u_j | U^D_{t_j}, Z, \Omega) \times \prod_{u_j \in \bar{U}^D_\infty} \prod_{u_i \in U^D_\infty} (1 - f(z_i, \omega_j))$$

avec :

$$P^D_j = P(u_j | U^D_{t_i}, \mathcal{Z}, \Omega) = 1 - \prod_{u_i \in U^D_{t_i}} (1 - f(z_i, \omega_j)) \tag{5.3}$$

La vraisemblance des ensembles de projections $\mathcal{Z}, \Omega$ par rapport à un ensemble d'épisodes de diffusion s'écrit donc :

$$\begin{aligned} \mathcal{L}(\mathcal{Z}, \Omega; \mathcal{D}) &= \sum_{D \in \mathcal{D}} \log P(D|\mathcal{Z}, \Omega) \\ &= \sum_{D \in \mathcal{D}} \left(\sum_{u_j \in U^D_\infty} \log(P^D_j) + \sum_{u_j \in \bar{U}^D_\infty} \sum_{u_i \in U^D_\infty} \log(1 - f(z_i, \omega_j)) \right) \end{aligned} \tag{5.4}$$

L'apprentissage des projections utilisateurs à partir d'un ensemble d'épisodes de diffusion D prend donc la forme :

$$\mathcal{Z}^\star, \Omega^\star = \arg\max_{\mathcal{Z},\Omega} \mathcal{L}(\mathcal{Z}, \Omega; \mathcal{D})$$

À partir de ces équation, nous pouvons reprendre le développement de l'algorithme EM de la même façon que dans le chapitre 3, page 72. Nous arrivons à la fonction d'espérance $\mathcal{Q}(\mathcal{Z}, \Omega | \hat{\mathcal{Z}}, \hat{\Omega})$ de la forme :

$$\mathcal{Q}(\mathcal{Z}, \Omega | \hat{\mathcal{Z}}, \hat{\Omega}) = \sum_{D \in \mathcal{D}} \left(\Phi^D(\mathcal{Z}, \Omega | \hat{\mathcal{Z}}, \hat{\Omega}) + \sum_{u_j \in \bar{U}^D_\infty} \sum_{u_i \in U^D_\infty} \log(1 - f(z_i, \omega_j)) \right) \tag{5.5}$$

avec :

$$\Phi^D(\mathcal{Z}, \Omega | \hat{\mathcal{Z}}, \hat{\Omega}) = \sum_{u_j \in U^D_\infty} \sum_{u_i \in (U^D_{t_j} \cap \mathrm{Preds}_j)} \left(\hat{P}^D_{i \to j} \log(f(z_i, \omega_j)) + \left(1 - \hat{P}^D_{i \to j}\right) \log(1 - f(z_i, \omega_j)) \right)$$

et :

$$\hat{P}^D_{i \to j} = \frac{f(\hat{z}_i, \hat{\omega}_j)}{\hat{P}^D_j}$$

L'utilisation d'un espace de représentation fait que les différentes probabilités de transmission ne sont plus libres : les contraintes géométriques rendent leurs valeurs interdépendantes. Maximiser $\mathcal{Q}(.|\hat{\mathcal{Z}}, \hat{\Omega})$ ne peut alors plus se décomposer en un ensemble de sous-problèmes convexes comme cela peut être le cas avec un modèle DAIC, et l'étape de maximisation ne possède donc pas de solution analytique comme c'était le cas dans le chapitre 3, avec l'équation 3.7. Néanmoins, il est possible de définir une procédure de montée de gradient stochastique convergeant vers un bon maximum local.

L'algorithme 2 détaille la procédure utilisée pour apprendre $\mathcal{Z}$ et Ω. Il s'agit d'un algorithme EM, similaire à celui du chapitre 3, mais où l'étape de maximisation est remplacée par un pas de gradient stochastique visant à augmenter la valeur de $\mathcal{Q}$. Une itération se déroule ainsi :

1. Ligne 7 : tirage uniforme d'un épisode de diffusion D et d'un utilisateur u_j n'étant pas la source de D ;
2. Lignes 9 à 12 : si u_j fait partie de D, calcul des estimations courantes $\hat{P}^D_j$ et $\hat{p}_{i,j}$ pour chaque utilisateur u_i infecté avant u_j (selon les formules 5.3 et 5.1 en utilisant les valeurs courantes de z_i et ω_j) ;
3. Lignes 13 à 22 : mise à jour des valeurs de $\mathcal{Z}$ et Ω avec un pas de gradient pour augmenter la valeur de $Q(\mathcal{Z}, \Omega | \hat{\mathcal{Z}}, \hat{\Omega})$. Si u_j est infecté dans D, cette mise à jour fait intervenir la dérivée de $\Phi^D(\mathcal{Z}, \Omega | \hat{\mathcal{Z}}, \hat{\Omega})$ (lignes 16 à 19). Sinon, elle fait intervenir la dérivée de $\log(1 - f(z_i, \omega_j))$ (lignes 19 à 21). Le pas d'apprentissage ϵ est fixé à 10^{-4}.

Algorithme 2 : Apprentissage du modèle IC projeté

Entrées :

U : l'ensemble des utilisateurs ; $\mathcal{D}$: l'ensemble des épisodes de diffusion;
d : le nombre de dimensions ; ϵ : le pas d'apprentissage;
$freq$: la fréquence des tests de convergence;

Sorties :

$\mathcal{Z} = \{\forall u_i \in U : z_i \in R^d\}$; $\Omega = \{\forall u_i \in U : \omega_i \in R^d\}$;

$nbProbas \leftarrow \sum_{D\in\mathcal{D}} \sum_{u_i \in U_\infty^D} |\overline{U}^D_{t_i+1}|$;
pour $u_i \in U$ **faire**
  Tirage uniforme de $z_i \in [-1,1]^d$; Tirage uniforme de $\omega_i \in [-1,1]^d$;
fin
$oldL \leftarrow -\infty$; $it \leftarrow 0$;
tant que *true* **faire**
  Tirage uniforme de $D \in \mathcal{D}$ et $u_j \in \bar{U}_1^D$;
  $\beta \leftarrow |\mathcal{D}| \times |\ \bar{U}_1^D\ | \times \frac{1}{nbProbas}$;
  si $t_j^D < \infty$ **alors**
    $\hat{p}_{i,j} \leftarrow f(z_i, \omega_j)$;
    $\hat{P}_j^D \leftarrow 1 - \prod_{u_i \in (U_{t_j}^D \cap \mathrm{Preds}_j)} (1 - f(z_i, \omega_j))$
  fin
  pour $u_i \in U_{t_j}^D$ **faire**
    $\xi_i^+ \leftarrow \dfrac{\partial \log f(z_i,\omega_j)}{\partial z_i}$; $\xi_i^- \leftarrow \dfrac{\partial \log(1 - f(z_i,\omega_j))}{\partial z_i}$;
    $\xi_j^+ \leftarrow \dfrac{\partial \log f(z_i,\omega_j)}{\partial \omega_j}$; $\xi_j^- \leftarrow \dfrac{\partial \log(1 - f(z_i,\omega_j))}{\partial \omega_j}$;
    si $t_j^D < \infty$ **alors**
      $z_i \leftarrow z_i + \beta \times \epsilon \times \big(\frac{\hat{p}_{i,j}}{\hat{P}_j^D}\xi_i^+ + (1 - \frac{\hat{p}_{i,j}}{\hat{P}_j^D})\xi_i^-\big)$;
      $\omega_j \leftarrow \omega_j + \beta \times \epsilon \times \big(\frac{\hat{p}_{i,j}}{\hat{P}_j^D}\xi_j^+ + (1 - \frac{\hat{p}_{i,j}}{\hat{P}_j^D})\xi_j^-\big)$;
    sinon
      $z_i \leftarrow z_i + \beta \times \epsilon \times \xi_i^-$; $\omega_j \leftarrow \omega_j + \beta \times \epsilon \times \xi_j^-$;
    fin
  fin
  $it \leftarrow it + 1$;
  si $it \ mod \ freq = 0$ **alors**
    $L \leftarrow$ Calcul de la log-vraisemblance selon (5.4) avec Z et Ω
    si $L \leq oldL$ **alors**
      retourner $(\mathcal{Z}, \Omega)$;
    fin
    $oldL \leftarrow L$;
  fin
fin

4. Lignes 20 à 30 : test de convergence toutes les *freq* itérations (1000000 dans notre cas). Le processus s'arrête si la log-vraisemblance sur $\mathcal{D}$ n'a pas suffisamment augmenté depuis le dernier test.

Notons enfin qu'à cause du tirage effectué à la ligne 7, les différents couples (u_i, u_j) ne sont pas associées aux épisodes de D dans les mêmes proportions que dans la formule 5.4 : les paires d'utilisateurs apparaissant dans les épisodes plus courts sont tirées plus souvent.

Dans notre procédure de tirage aléatoire, la probabilité de considérer les relations vers un utilisateur u_j dans un épisode D est égale à :

$$\frac{1}{|\mathcal{D}| \times |\overline{U}_1^D|}$$

Or, nous voudrions que chaque relation de transmission soit considérée dans les mêmes proportions que dans $\mathcal{Q}(\mathcal{Z}, \Omega|\hat{\mathcal{Z}}, \hat{\Omega})$, c'est à dire tirée avec une probabilité :

$$\frac{1}{\sum_{D\in\mathcal{D}} \sum_{u_i \in U_\infty^D} |\overline{U}_{t_i+1}^D|}$$

dont le dénominateur est noté *nbProbas* (ligne 1) dans l'algorithme. Cette valeur correspond au nombre total de relations considérée dans la formule $\mathcal{Q}$. Nous calculons donc, en ligne 8, un poids β permettant de corriger ce biais.

$$\beta = \frac{|\mathcal{D}| \times |\ \bar{U}_1^D\ |}{\sum_{D\in\mathcal{D}} \sum_{u_i \in U_\infty^D} |\overline{U}_{t_i+1}^D|}$$

Cette valeur β est utilisée lors de la mise à jour des paramètres (lignes 16 et 21).

5.3 Expériences

5.3.1 Corpus et modèles de référence

Dans nos expériences, les jeux de données suivants ont été utilisés :

Lastfm : corpus issu d'un site d'écoute de musique en streaming, collecté pendant un an par [Celma, 2010]. Chaque épisode regroupe les événements d'écoute d'un morceau.

Irvine : corpus présenté dans [Opsahl and Panzarasa, 2009] regroupant les participations d'étudiants de l'université d'Irvine à des forums en ligne. Chaque épisode regroupe l'ensemble des participations à un fil de discussion particulier.

Corpus	$\|U\|$	$\|E\|$	Densité	$\|\mathcal{D}\|$ Appr.	$\|\mathcal{D}\|$ Test	Taille Episode Moy.
Irvine	847	74871	0.1	433	49	14.6
Icwsm	2270	4775	0.001	19027	1000	2.22
Memetracker	498	229073	0.9	10000	1000	2.17
Digg	3295	689416	0.06	17000	1000	2.43
Twitter	2841	884832	0.09	10000	1000	20.5
LastFm	986	708159	0.72	10000	1000	7.25

TABLE 5.1 – Quelques statistiques sur les jeux de données.

Twitter, Memetracker, ICWSM et Digg : présentés dans le chapitre 3, page 83.

La table 5.1 donne quelques statistiques sur les jeux de données utilisés : nombre d'utilisateurs, nombre de liens et densité du graphe, nombre d'épisodes en apprentissage et en test et enfin taille moyenne des épisodes de diffusion. Comme dans le chapitre précédent, le graphe utilisé pour apprendre les modèles est le graphe complet privé des arêtes correspondant à des paires d'utilisateurs sans exemples positifs $\mathcal{D}^{?}_{i,j}$.

Nous comparons notre modèle à un modèle DAIC appris en utilisant notre algorithme présenté dans le chapitre 3, ainsi qu'aux modèles NetRate et CTIC déjà présentés dans le même chapitre.

5.3.2 Évaluation

De la même façon que dans le chapitre 3, nous ne connaissons pas les *vraies* probabilités de transmissions à comparer aux probabilités définies par les différents modèles. Nous évaluons donc les modèles de façon indirecte, en prédisant à partir d'une ou plusieurs sources U_1^D la probabilité pour chaque utilisateur de devenir infecté, en simulant un grand nombre de fois la diffusion avec les probabilités apprise. Comme nous l'avons expliqué en conclusion du chapitre 2, il n'existe pas de définition formelle couramment admise du problème de « prédiction de diffusion ». Dans ce chapitre, nous comparons donc les probabilités d'infection finales prédites à U_∞^D sur un ensemble d'épisodes de diffusion de test en utilisant plusieurs mesures différentes afin d'observer leurs comportements et comparer les modèles selon plusieurs logiques différentes.

MSE : les probabilités d'infection prédites sont comparées au vraies valeurs (0 ou 1) avec une mesure d'erreur quadratique.

Log-Vraisemblance : la log-vraisemblance de l'ensemble des épisodes de diffusion de test selon les différents modèles. Les probabilités sont projetées sur l'intervalle $[10^{-5}, 1 - 10^{-5}]$ pour éviter de calculer $\log(0)$

F1 : la mesure classique $F1$, définie comme dans le chapitre précédent.

MAP : les utilisateurs sont classés par ordre décroissant de probabilité d'infection, et la liste est évaluée par Mean-Average-Precision. La précision moyenne calculée sur un épisode de diffusion D est définie comme :

$$P_{\text{moy}} = \frac{1}{|U_\infty^D|} \sum_{k=0}^{N-1} (P_k \times \text{rel}(k))$$

où P_k est la précision au rang k (le taux d'utilisateurs effectivement infectés dans D parmi les k utilisateurs les mieux classés) et $\text{rel}(k)$ vaut 1 si le k-ième utilisateur de la liste fait partie des utilisateurs infectés dans D, 0 sinon. La MAP est la moyenne de la précision moyenne ainsi calculée sur l'ensemble des épisodes de test.

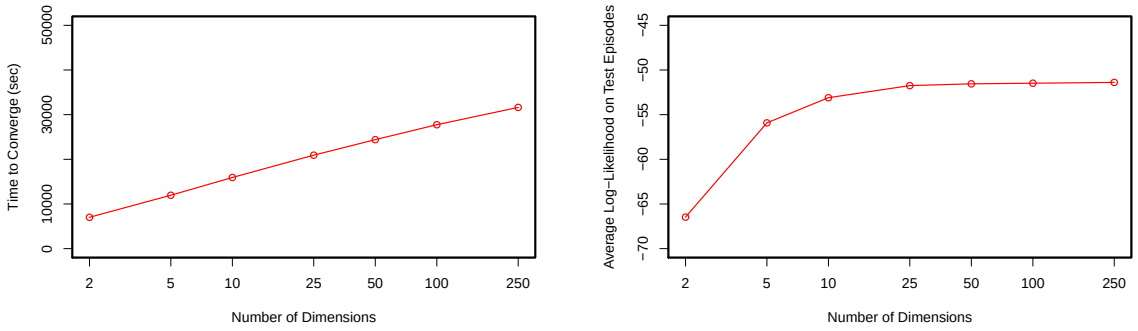


FIGURE 5.3 – Durée (en secondes) de l'apprentissage de notre modèle jusqu'à convergence, et log-vraisemblance obtenue en test, sur le jeu de données Digg.

Avant de comparer notre modèle à ceux présentés ci-dessus, nous étudions l'impact du nombre de dimensions sur la durée de l'apprentissage (sur un ordinateur de bureau équipé d'un processeur Intel(R) Core(TM) i7 CPU 950@3.07GHz) et sur les performances du modèle. Nous ne reportons en figure 5.3 que les résultats obtenus sur Digg, mais les tendances observées sur les autres jeux de données sont similaires. Nous remarquons que la durée de l'apprentissage augmente logarithmiquement avec le nombre de dimensions, mais que la qualité du modèle augmente peu au delà de 25 dimensions. Dans la suite, nous utilisons donc un espace de dimension $d = 25$ pour l'apprentissage.

5.3.3 Résultats

Les résultats sont présentés en table 5.2. Nous pouvons tout d'abord remarquer que notre modèle (IC Proj) obtient des résultats toujours *au moins aussi bons* que ceux du modèle DAIC. La comparaison entre les résultats de DAIC et ceux de NetRate et CTIC conduit pour sa part à des conclusions similaires à celles du chapitre 3, le modèle DAIC étant presque toujours meilleur. Ces résultats montrent que nous parvenons à calculer

Corpus	Modèle	MSE	LogVrai.	MAP	F1	nbParams
Irvine	DAIC	15,31	-960,5	0,079	0,020	74871
	NetRate	15,42	-892,13	0,078	0,019	74871
	CTIC	15,29	-771,42*	**0,080**	0,020	149742
	IC Proj	**14,53***	**-532,5***	0,079	**0,025***	42350
ICWSM	DAIC	0,2	-8,3	0,77	**0,651**	4775
	NetRate	0,23	-9,01	0,72	0,357	4775
	CTIC	0,22	-8,46	0,76	0,482	9550
	IC Proj	**0,19**	**-6,14***	**0,78**	**0,651**	113500
MemeTracker	DAIC	32,62	-795,85	0,22	0,0585	229073
	NetRate	34,55	-850,48	0,17	0,0442	229073
	CTIC	33,27	-802,52	0,22	0,0551	458146
	IC Proj	**32,15**	**-791,3**	**0,23**	**0,0632***	24900
Digg	DAIC	2,1	-69,5	0,411	**0,201**	689416
	NetRate	1,95	-64,01*	0,409	0,199	689416
	CTIC	1,92*	-64,18*	0,413	**0,201**	1378832
	IC Proj	**1,79***	**-51,75***	**0,434***	0,198	164750
Twitter	DAIC	6,70	-412,75	0,047	0,012	884832
	NetRate	6,91	-428,78	0,039	0,011	884832
	CTIC	6,72	-401,56	0,049	0,012	1769664
	IC Proj	**5,47***	**-223,15***	**0,056***	**0,013**	142050
LastFM	DAIC	12,13	-409,5	0,132	0,026	708159
	NetRate	13,91	-413,02	0,112	0,022	708159
	CTIC	12,12	-409,3	0,128	0,025	1416318
	IC Proj	**11,62***	**-405**	**0,151***	**0,027**	49300

TABLE 5.2 – Résultats obtenus sur les différents corpus. Les valeurs marquées d'un astérisque sont significativement meilleures que celles obtenues par DAIC (Test-t de Student 95%), et celles en gras indiquent le meilleur résultat obtenu sur chaque corpus.

correctement les probabilités de transmission dans un espace vectoriel. Il est important de remarquer que cela se fait le plus souvent avec un nombre de paramètres (colonne de droite) largement inférieur au modèle DAIC, sauf sur le corpus ICWSM, dont le graphe des exemples positifs est très creux.

Il est intéressant de remarquer que les mesures se comportent de façons assez différentes :

1. La log-vraisemblance des épisodes de test est significativement meilleure avec notre IC projeté sur Irvine, ICWSM, Digg et Twitter. Cela peut se comprendre en observant la table 5.1 : on remarque que ces corpus sont ceux dont le graphe a une densité assez faible (de 0.001 à 0.1 contre 0.72 et 0.9 pour les corpus LastFM et Memetracker). Rappelons que le « graphe » d'un corpus désigne ici le graphe construit à partir de $\mathcal{D}$ et contenant tous les liens (u_i, u_j) tels qu'il existe au moins un épisode où u_j apparaît après u_i, et uniquement ceux-ci. Dès lors, sur ces corpus, le modèle DAIC classique ne peut apprendre qu'un nombre limité de probabilités de transmissions. Il arrive donc fréquemment que certaines infections observées dans l'ensemble de test ne puissent pas être expliquées correctement par ce modèle DAIC, ce qui conduit à une vraisemblance plus faible. Notre approche, en revanche, est capable d'inférer des probabilités de transmission pertinentes pour les paires d'utilisateurs n'ayant aucun exemple positif grâce à l'utilisation d'un espace de représentation (par exemple, en suivant l'un des principes illustrés en figure 5.1). Cela explique que IC projeté obtienne une meilleure log-vraisemblance sur les épisodes de test de ces corpus.
2. La MSE de notre modèle est significativement meilleure sur Irvine, Digg, Twitter et LastFM. Remarquons qu'il s'agit de jeux de données sur lesquels la diffusion est en fait délicate à caractériser : le fait que deux utilisateurs interagissent avec le même item ne veux pas forcément dire qu'il y a eu diffusion de l'un à l'autre, il peut s'agir uniquement d'une corrélation sur leurs centres d'intérêt. Distinguer les relations de *corrélation* des relations d'*influence* est une tâche difficile, les deux concepts étant étroitement liés [Anagnostopoulos et al., 2008].
3. La mesure MAP se comporte d'une façon similaire à la MSE, et indique que IC projeté est meilleur que DAIC sur pratiquement les mêmes corpus. En effet, la mesure MAP favorise les prédictions telles que les utilisateurs de U_∞^D aient des probabilités plus élevées que ceux de $\bar{U}_\infty^D$. La MSE récompensant les probabilités d'infections finales proches de 1 pour les utilisateurs de U_∞^D et 0 pour les autres, il est logique qu'un modèle avec une meilleure MSE ait également tendance à obtenir une meilleure MAP.
4. La mesure F1, en revanche, n'indique pas que le modèle IC Projeté est significativement meilleur sur les mêmes corpus que la MAP. Les deux mesures sont pourtant des fonctions de la précision et du rappel, mais la mesure MAP accorde plus d'importance à la précision dans les premiers rangs. Il est toutefois intéressant de remarquer que les corpus sur lesquels IC projeté obtient un score MAP

significativement meilleur sont ceux des réseaux sociaux fonctionnant en « hub », c'est à dire où un utilisateur se connectant sur le site considéré voit d'abord une liste des informations les plus populaires du moment, éventuellement pondérées par ses centres d'intérêt (tubes sur LastFM, "trends" sur Twitter, informations "hot" sur Digg). Certains utilisateurs vont alors systématiquement interagir avec certains contenus ainsi mis en avant. Une explication possible de la meilleure MAP obtenue par IC projeté sur ces corpus est donc que l'utilisation d'un biais $\omega_j^{(0)}$ dans la formule 5.2 permet de modéliser ce comportement, et de prédire un score plus élevé pour ces utilisateurs ayant tendance à suivre les informations populaires et à être plus souvent infectés, ce qui permet d'obtenir une meilleure MAP. Une autre explication possible de ces résultats est que sur ces corpus, la présence de hubs fait que la diffusion a lieu de façon plus « globale » : plus une information est populaire, plus elle a de chances d'infecter de nouveaux utilisateurs à travers tout le réseau. Ce phénomène est mieux représenté par notre modèle, où chaque utilisateur a une chance d'infecter tous les autres dans l'espace de représentation

Le premier point évoqué ci-dessus peut être vérifié en calculant le coefficient de Jaccard entre l'ensemble des liens du graphe des exemples positifs extrait des épisodes d'apprentissage et celui du graphe extrait des épisodes de test. Rappelons que le coefficient de Jaccard mesure la similarité entre deux ensembles et est égal au rapport entre la taille de leur intersection et la taille de leur union. Les valeurs obtenues sont indiquées en table 5.3. Nous pouvons voir que les valeurs du coefficient sont bien plus faibles sur Irvine, ICWSM, Digg et Twitter, ce qui signifie que beaucoup d'interactions utilisateurs dans les épisodes de test ne sont jamais observées dans les épisodes d'apprentissage. Cela confirme notre analyse.

Corpus	Coefficient de Jaccard
Irvine	0.05
ICWSM	0.17
MemeTracker	0.77
Digg	0.26
Twitter	0.37
LastFM	0.71

Table 5.3 – Coefficients de Jaccard entre les liens du graphe des exemples posififs des épisodes d'apprentissage et les liens du graphe des exemples posififs des épisodes de test.

Globalement, les résultats présentés ici indiquent que le modèle IC projeté, où les probabilités dépendent des représentations des utilisateurs, fonctionne aussi bien que le modèle DAIC où les probabilités sont apprises séparément. De plus, ce modèle IC projeté parvient à obtenir des résultats significativement meilleurs dans certains cas, tout en apprenant beaucoup moins de paramètres. Ces résultats nous apprennent aussi que la diffusion d'information est un phénomène complexe et protéiforme. Les jeux de données collectés ici sont de tailles et de densités très différentes, et chacun possède son propre « type » de

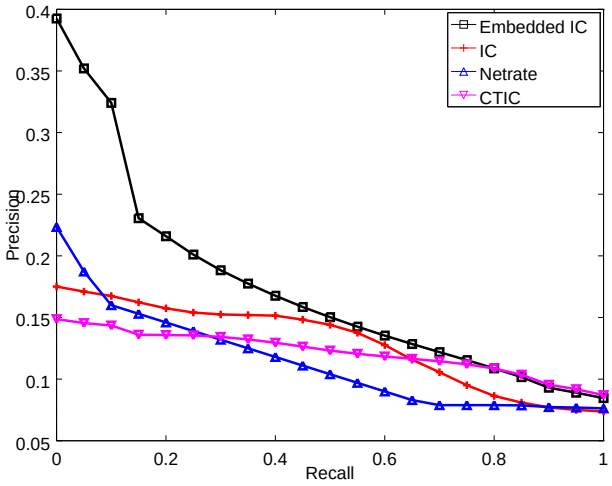


FIGURE 5.4 – Précision-Rappel de la détection de relations sur Memetracker.

diffusion (retweets, discussion, recommandation). Les différentes mesures utilisées correspondent également à des principes d'évaluation différents, et n'ont pas donné les mêmes résultats sur tous les corpus.

Néanmoins, d'une façon générale, les résultats mettent en avant les propriétés avantageuses de cette approche : des résultats meilleurs avec moins de paramètres à apprendre, une meilleure capacité de généralisation et une capacité à modéliser différents types de diffusion.

5.3.4 Détection de Relations d'Influence

Pour compléter les expérimentations menées, nous évaluons notre modèle sur sa capacité à retrouver un ensemble de relations de diffusion connues à partir d'un ensemble $\mathcal{D}$ d'épisodes de diffusion observés. De la même façon que dans [Gomez-Rodriguez et al., 2011, Gomez Rodriguez et al., 2010], l'idée est d'utiliser les hyperliens liant des messages du corpus memetracker comme l'ensemble de relations à identifier. À chaque fois qu'un utilisateur de ce corpus poste un message contenant un hyperlien vers un autre message, nous créons un lien de l'auteur du second message vers l'auteur du premier. Les liens du graphe ainsi construit correspondent donc de façon *certaine* à des liens de diffusion.

Pour chaque paire (u_i, u_j), la probabilité $p_{i,j}$ apprise selon chacun des modèles est interprétée comme la probabilité d'existence d'un lien entre ces deux utilisateurs. Nous trions les probabilités ainsi apprises par chaque modèle et évaluons les résultats avec une courbe de Précision-Rappel. Les résultats sont présentés en figure 5.4. Nous indiquons aussi ceux obtenus par les modèles CTIC et NetRate.

Cette expérience illustre la meilleure capacité de notre modèle à identifier des liens d'influence pertinents à partir d'épisodes de diffusion, sa courbe de precision en fonction du rappel se situant au dessus de celle des autres modèles. De plus, le fait de projeter les utilisateurs dans un espace de représentation pour définir les probabilités de transmission permet au modèle de découvrir des relations sans exemple de diffusion dans les épisodes observés, et donc d'obtenir de meilleurs résultats sur cette tâche de prédiction de liens. Enfin, dans cette approche, les contraintes géométriques permettent aussi d'éviter le problème du biais d'apprentissage décrit dans le chapitre 3, section 3.3.1. En effet, les valeurs des probabilités de transmission ne sont pas indépendantes et ne peuvent pas prendre de valeurs extrêmes (proches de 0 ou 1) sans avoir d'effets sur le reste des représentations.

5.4 Conclusion

Dans ce chapitre, nous avons présenté une version « projetée » du modèle IC, les probabilités de transmission étant définies de façon indirecte à partir des distances séparant les utilisateurs dans un espace de représentation. Cette approche permet d'une part de régulariser les relations en limitant l'impact du bruit et du surapprentissage, et d'autre part d'inférer des probabilités de transmission entre des utilisateurs pour lesquels aucun exemple positif n'existe. Cette capacité de généralisation est importante, les données issues de réseaux sociaux étant souvent bruitées et parcimonieuses.

Les résultats obtenus en prédiction de liens ont montré la capacité de notre modèle à repérer les liens les plus pertinents. Les résultats en prédiction de diffusion ont montré que la version projetée du modèle IC fonctionnait mieux que la version DAIC, voire bien mieux dans certains contextes, tout en nécessitant d'apprendre moins de paramètres. Les différences dans le comportement des mesures d'évaluations nous ont également montré que la diffusion était un phénomène complexe, délicat à caractériser et à évaluer, et étroitement lié au fonctionnement du site sur lequel il a lieu.

Chapitre 6

Modélisation par diffusion de chaleur

Résumé Ce chapitre décrit une troisième contribution, publiée dans [Bourigault et al., 2014]. Nous utilisons l'apprentissage de représentations pour projeter les utilisateurs dans un espace $\mathbb{R}^d$ de façon à ce que la diffusion d'information puisse cette fois ci être modélisée comme un processus de diffusion de chaleur au sein même de cet espace. Le modèle obtenu est donc *continu*, et non plus *itératif*. Il obtient des résultats similaires à ceux du chapitre précédent tout en étant plus rapide en inférence. Nous présentons également une extension permettant de prendre en compte le *contenu* de l'information se diffusant.

6.1 Introduction

6.1.1 Modèles explicatifs ou prédictifs

Dans les chapitres 3 et 5, nous avons étudiés des modèles *explicatifs*, visant à inférer précisément comment l'information se diffuse d'un utilisateur à l'autre. À chaque fois, nous avons évalué les performances de nos modèles et des modèles de références sur la tâche de *prédiction de diffusion* : notre but était de retrouver, à partir d'un ensemble d'utilisateurs initiaux, quels seraient les utilisateurs finalement infectés, l'évaluation se faisant avec des mesures issues de la recherche de documents (F1 et MAP). Cette méthode d'évaluation était rendue nécessaire par l'absence de vérité terrain sur les déroulement précis de la diffusion d'un utilisateur à l'autre.

Il est important de remarquer que nous avons utilisé un modèle *explicatif* et *génératif* pour une tâche de *prédiction*. Le modèle IC repose sur une simulation du processus de diffusion lui-même, et permet donc non-seulement de prédire *quels* utilisateurs seront infectés, mais aussi *par qui*. Durant l'étape d'apprentissage, c'est bien cette propriété qui est apprise : retrouver quels sont les chemins empruntés par l'information (sous la forme de probabilités de transmission). Nous avons d'ailleurs pu évaluer la capacité de

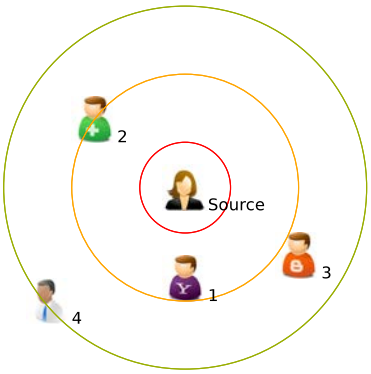


FIGURE 6.1 – Exemple de diffusion d'information continue dans un espace de représentation. L'utilisateur source diffuse de la *chaleur* dans l'espace. Les cercles colorés représentent des lignes de niveau de chaleur, et les indices des utilisateurs *l'ordre* dans lequel ils deviennent infectés.

notre approche à identifier les liens de diffusion dans le chapitre 5 (figure 5.4). Cette information apprise est ensuite utilisée pour prédire quels utilisateurs seront infectés. La tâche de prédiction de diffusion est donc résolue par l'intermédiaire d'une tâche « annexe ». Dans l'optique d'une problématique purement prédictive, une autre approche est toutefois possible. Vladimir Vapnik [Vapnik, 2013] écrivait ainsi :

> « Pour résoudre un problème donné, évitez de résoudre un problème plus général en tant qu'étape intermédiaire »

Dans ce chapitre, nous proposons donc une nouvelle approche du problème, également basée sur l'apprentissage de représentations, mais n'utilisant plus de modèle explicatif.

6.1.2 Diffusion de chaleur

Notre idée est de projeter les utilisateurs du réseau dans un espace continu de façon à ce que la diffusion d'information dans le réseau social puisse être vue comme un processus de *diffusion de chaleur* dans cet espace : plus un utilisateur est *chaud* à un instant t, plus il est infecté ou a de chances d'être infecté à cet instant, d'une façon similaire à [Kondor and Lafferty, 2002]. Une illustration du principe est donnée en figure 6.1.

Comme dans le chapitre précédent, l'utilisation d'un espace de représentation continu nous permet de définir un modèle relativement compact, et de régulariser les relations entre utilisateurs. De plus, le problème d'apprentissage des représentations des utilisateurs devient un problème d'optimisation continue pouvant être résolu par une descente de gradient classique. Enfin, l'utilisation du modèle en inférence est bien plus simple. Le calcul peut être effectué rapidement, sans avoir à réaliser des simulations comme dans la partie précédente. Nous proposons également une extension de ce modèle permettant d'intégrer le contenu de l'information se diffusant.

6.2 Modèle

6.2.1 Noyaux

6.2.1.1 Noyaux de diffusion de chaleur

Considérons l'espace de représentation $\mathbb{R}^d$. Un processus de diffusion de chaleur dans cet espace est défini par une fonction $f : \mathbb{R}^d \times \mathbb{R}^+ \to \mathbb{R}$ où $f(x,t)$ représente la température en un point x au temps t. Cette fonction est caractérisée par *l'équation de la chaleur* :

$$\begin{cases} \frac{\partial f}{\partial t} - \Delta f = 0 \\ f(x,0) = f_0(x) \end{cases} \tag{6.1}$$

où $f_0(x)$ représente l'état initial et Δ est l'opérateur Laplacien (la dérivée seconde par rapport à l'espace) . Nous définissons ensuite $K : \mathbb{R}^+ \times \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ tel que $K(t,y,x)$ corresponde à la température au point x à l'instant t avec une source initiale située en y, c'est à dire avec les conditions initiales suivantes :

$$K(0,y,x) = \delta(y-x) = f_0(x) \tag{6.2}$$

où δ est la fonction Dirac (valant 1 en 0 et 0 partout ailleurs). Dans un espace euclidien de dimension d, l'équation différentielle 6.1 avec les conditions initiales 6.2 possède une solution appelée *noyau de chaleur* :

$$K(t,y,x) = (4\pi t)^{-\frac{d}{2}} e^{-\frac{||y-x||^2}{4t}} \tag{6.3}$$

6.2.1.2 Définition d'un noyau pour la diffusion d'information

L'idée est d'utiliser le phénomène physique de diffusion de chaleur pour traiter la tâche de prédiction de diffusion d'information sur les réseaux sociaux. En conservant la notation $\mathcal{Z} = (z_i)_{u_i \in U}$ pour désigner l'ensemble des projections des utilisateurs, il est possible de réécrire le noyau de diffusion défini par la formule 6.3 comme une fonction $K_{\mathcal{Z}}(t, s_D, u_i)$ retournant un score de contamination pour l'utilisateur u_i à l'instant t d'une diffusion ayant été initiée par l'utilisateur s_D :

$$K_{\mathcal{Z}}(t, s_D, u_i) = (4\pi t)^{-\frac{d}{2}} e^{-\frac{||z_{s_D} - z_i||^2}{4t}} \tag{6.4}$$

Il s'agit alors de déterminer les valeurs optimales de $\mathcal{Z}$ à partir de $\mathcal{D}$. Pour cela, nous définissons une fonction de coût :

$$\mathcal{L}(\mathcal{Z}; \mathcal{D}) = \sum_{D \in \mathcal{D}} l(K_{\mathcal{Z}}(., s_D, .), D) \tag{6.5}$$

où $l(K_{\mathcal{Z}}(.,s_D,.),D)$ mesure, pour une source s_D, un coût entre la prédiction du noyau $K_{\mathcal{Z}}$ et l'épisode D. Ce coût sera présenté dans la sous-section suivante. Enfin, le problème final consiste alors à trouver $\mathcal{Z}^*$ tel que :

$$\mathcal{Z}^* = argmin_{\mathcal{Z}}\ \mathcal{L}(\mathcal{Z};\mathcal{D}) \tag{6.6}$$

6.2.1.3 Fonction de coût

Comme nous l'avons dit dans l'introduction, notre but est de prédire les utilisateurs infectés directement, sans passer par l'estimation de l'information manquante « quelles contaminations ont eu lieu ». Nous proposons de réaliser l'apprentissage des représentations des utilisateurs en contraignant uniquement le noyau à contaminer les utilisateurs *dans le même ordre* que dans les épisodes d'apprentissage, sans passer par la modélisation de la dynamique temporelle qui serait un problème plus complexe à résoudre, et sans chercher à identifier les transmissions ayant eu lieu. Nous définissons pour cela deux contraintes sur chaque épisode de diffusion D de l'ensemble d'apprentissage $\mathcal{D}$:

— pour tout couple d'utilisateurs (u_i, u_j) tel que u_i et u_j soient contaminés dans l'épisode D et tel que $t_i^D < t_j^D$, $K_{\mathcal{Z}}$ doit être défini de façon à ce que $\forall t, K_{\mathcal{Z}}(t,s_D,u_i) > K_{\mathcal{Z}}(t,s_D,u_j)$;
— pour tout couple d'utilisateurs (u_i, u_j) tel que $u_i \in U_\infty^D$ et $u_j \notin U_\infty^D$, $K_{\mathcal{Z}}$ doit être défini de façon à ce que $\forall t, K_{\mathcal{Z}}(t,s_D,u_i) > K_{\mathcal{Z}}(t,s_D,u_j)$.

Soit :

$$\big(\forall (u_i,u_j)\in U^2, \forall D \in \mathcal{D}_{i,j}^{?}, \forall t\big) : K_{\mathcal{Z}}(t,s_D,u_i) > K_{\mathcal{Z}}(t,s_D,u_j)$$
$$\big(\forall (u_i,u_j)\in U^2, \forall D \in \mathcal{D}_{i,j}^{-}, \forall t\big) : K_{\mathcal{Z}}(t,s_D,u_i) > K_{\mathcal{Z}}(t,s_D,u_j)$$

La fonction $K_{\mathcal{Z}}$ est décroissante selon la distance séparant les projections des utilisateurs concernés pour t fixé, i.e :

$$\forall (u_i,u_j,u_k,t)\in U\times U\times U\times \mathbb{R}^+, ||z_k - z_i||^2 \leq ||z_k - z_j||^2 \implies K_{\mathcal{Z}}(t,u_k,u_i) \geq K_{\mathcal{Z}}(t,u_k,u_j)$$

Les contraintes peuvent donc être réécrites, plus simplement, ainsi :

$$\big(\forall (u_i,u_j)\in U^2, \forall D \in \mathcal{D}_{i,j}^{?}\big) : ||z_{s_D} - z_i||^2 < ||z_{s_D} - z_j||^2 \tag{6.7}$$

$$\big(\forall (u_i,u_j)\in U^2, \forall D \in \mathcal{D}_{i,j}^{-}\big) : ||z_{s_D} - z_i||^2 < ||z_{s_D} - z_j||^2 \tag{6.8}$$

Ces contraintes expriment le fait que des utilisateurs infectés plus tôt dans un épisode D doivent être plus proches de la source s_D dans l'espace de représentation que ceux infectés plus tard (ou jamais). En utilisant une fonction de type hingeloss, nous pouvons exprimer

ces contraintes en définissant une fonction de coût d'ordonnancement l_{rang} :

$$\begin{aligned} l_{\text{rang}}(K_{\mathcal{Z}}(.,s_D,.),D) = & \sum_{\substack{u_i,u_j \in U^2 \\ D \in \mathcal{D}^?_{i,j}}} max(0, 1 - (||z_{s_D} - z_j||^2 - ||z_{s_D} - z_i||^2)) \\ & + \sum_{\substack{u_i,u_j \in U^2 \\ D \in \mathcal{D}^-_{i,j}}} max(0, 1 - (||z_{s_D} - z_j||^2 - ||z_{s_D} - z_i||^2)) \\ l_{\text{rang}}(K_{\mathcal{Z}}(.,s_D,.),D) = & \sum_{\substack{u_i,u_j \in U^2 \\ D \in (\mathcal{D}^?_{i,j} \cup \mathcal{D}^-_{i,j})}} max(0, 1 - (||z_{s_D} - z_j||^2 - ||z_{s_D} - z_i||^2)) \end{aligned} \tag{6.9}$$

Nous baptisons ce modèle « HDK », pour « Heat Diffusion Kernel ».

6.2.1.4 Relation temps-probabilité

Remarquons que nous avons défini ce modèle HDK de façon à ce qu'un utilisateur *infecté* dans un épisode de diffusion d'apprentissage D soit plus proche de la source qu'un utilisateur *non-infecté*, mais aussi de façon à ce qu'un utilisateur infecté *plus tôt* soit plus proche de la source qu'un utilisateur infecté *plus tard*. Ainsi, la distance séparant un utilisateur quelconque de la source d'une information représente à la fois sa propension à être infecté par cette information et sa tendance à être infecté plus ou moins rapidement par celle-ci. Nous avons donc, implicitement, considéré en définissant notre fonction de coût qu'un utilisateur infecté *plus tôt* dans un épisode de diffusion d'apprentissage était également *plus susceptible* d'être infecté, puisque ces deux propriétés sont modélisées par la distance dans l'espace de représentations.

Cette hypothèse sur la « relation temps-probabilité » nous permet d'améliorer la robustesse du modèle en augmentant le nombre d'exemples sur lesquelles l'apprentissage de celui-ci se base. Par exemple, les épisodes de diffusion (u_1, u_2, u_3, u_4) et (u_1, u_4, u_3, u_2) utilisés en apprentissage seraient équivalents si nous ne considérions pas l'*ordre* des utilisateurs infectés après la source u_1.

Nous justifions cette hypothèse par les observations suivantes :

1. Un utilisateur proche de la source dans le *graphe* d'un réseau social (au sens de la longueur du plus court chemin) a plus de chances d'être contaminé par les informations émanant de cette source, car moins de transmissions sont nécessaires.
2. Pour la même raison, un utilisateur proche de la source dans le *graphe* d'un réseau social a également tendance à être infecté *plus tôt* par les informations en provenance de cette source.

Le fait qu'un utilisateur soit infecté tôt dans un épisode d'apprentissage D peut donc indiquer qu'il avait également plus de chances d'être infecté.

Cette hypothèse est également justifiée par les résultats obtenus par Manuel Gomez-Rodrigez et Jure Leskovec dans leurs travaux sur l'inférence de graphe [Gomez Rodriguez et al., 2010, Gomez-Rodriguez et al., 2011, Gomez Rodriguez et al., 2013], basés sur une idée similaire. Notons toutefois que contrairement à eux, nous avons uniquement considéré cette relation temps-probabilités de façon *relative*, et non pas *absolue* : nous ne cherchons pas à reproduire les temps d'infection *exacts*, uniquement *l'ordre* dans lequel les utilisateurs sont infectés.

6.2.2 Algorithme d'apprentissage

Le coût final à minimiser est :

$$\mathcal{L}_{rang}(\mathcal{Z};\mathcal{D}) = \sum_{D\in\mathcal{D}} l_{rang}(K_{\mathcal{Z}}(.,s_D,.),D) \tag{6.10}$$

Différentes méthodes peuvent être utilisées pour optimiser la fonction objectif. Nous proposons d'utiliser la descente de gradient stochastique décrite dans l'algorithme 3. Après avoir initialisé aléatoirement les projections des utilisateurs (ligne 2), l'algorithme échantillonne à chaque itération un épisode D (ligne 8) et deux utilisateurs u_i et u_j, avec u_j un utilisateur non-infecté dans D ou infecté *après* u_i (lignes 9, 10). Si les contraintes définies en equations 6.7 et 6.8 ne sont pas respectées avec une marge suffisante pour cet épisode et ces utilisateurs (ligne 15), les projections z_i, z_j et z_{s_D} sont déplacées dans la direction du gradient de la fonction de coût, selon un pas d'apprentissage α (lignes 16 à 17). L'apprentissage continue ainsi jusqu'à ce que la valeur de $\mathcal{L}_{\mathcal{Z}}$ ne diminue plus (ligne 22).

Comme dans le chapitre précédent, le tirage effectué en lignes 8 à 10 introduit un biais, les différents triplets d'utilisateurs n'étant pas considérés dans les mêmes proportions que dans la fonction de coût (équations 6.9 et 6.10).

Le tirage aléatoire de l'algorithme considère chaque triplet (s_D, u_i, u_j) dans l'épisode D avec une probabilité :

$$\frac{1}{|D| \times |U^D_\infty| \times |\bar{U}^D_{t^D_i+1}|}$$

Nous voudrions que chaque triplet soit tiré selon la même probabilité que dans l'équation 6.10, soit :

$$\frac{1}{\sum_{D\in\mathcal{D}} \sum_{u_i\in U^D_\infty} |\overline{U}^D_{t^D_i+1}|}$$

Nous calculons donc un facteur de correction β, appliqué lors de la mise à jour des paramètres (lignes 16 à 17).

$$\beta = \frac{|D| \times |U_\infty^D| \times |\bar{U}^D_{t_i^D+1}|}{\sum_{D \in \mathcal{D}} \sum_{u_i \in U_\infty^D} |\overline{U}^D_{t_i^D+1}|}$$

6.2.3 Diffusion basée sur le contenu

Nous proposons maintenant une extension du modèle précédent capable de prendre en compte le contenu de chaque épisode de diffusion. En effet, dans la réalité, deux épisodes partant de la même source mais concernant des sujets différents n'infecteront pas les mêmes utilisateurs, ou pas dans le même ordre. Nous considérons que le contenu d'un épisode de diffusion D est représenté par un vecteur $w_D \in \mathbb{R}^Q$. Suivant le contexte applicatif, le vecteur w_D pourra correspondre à une représentation d'un texte (sac de mots ou tf-idf), ou à un vecteur de caractéristiques extraites d'une image, par exemple.

L'extension proposée se base sur le principe suivant : le contenu d'un épisode D a pour effet de modifier les représentations des utilisateurs, et donc de modifier la façon dont ce contenu se propage.

Pour cela, nous apprenons les paramètres θ d'une fonction linéaire permettant d'obtenir une représentation du contenu dans $\mathbb{R}^d$, définie ainsi :

$$f_\theta(w_D) = w_D.\theta$$

en considérant que la représentation du contenu w_D est un vecteur-ligne et que θ est une matrice à Q lignes et d colonnes. C'est cette représentation $f_\theta(w_D) \in \mathbb{R}^d$ qui est ensuite utilisée pour modifier les représentations des utilisateurs. Nous avons exploré deux façons différentes de modifier ces représentations. Dans les deux cas, les paramètres θ sont appris en même temps que $\mathcal{Z}$ en adaptant l'algorithme de descente de gradient.

6.2.3.1 Translation

Dans cette version, la représentation de l'utilisateur-source est translatée dans l'espace de représentation selon le vecteur $f_\theta(w_D)$ (on retrouve une idée similaire dans [Bordes et al., 2013]). La diffusion a donc lieu, dans l'espace de représentation, depuis le point $z_{s_D} + f_\theta(w_D)$. La fonction K devient devient :

$$K^{\text{trans}}_{\mathcal{Z},\theta}(w_D, t, s_D, u_i) = (4\pi t)^{-\frac{d}{2}} e^{-\frac{||z_{s_D} + f_\theta(w_D) - z_i||^2}{4t}} \tag{6.11}$$

Une illustration du principe est donnée en figure 6.2. Ce modèle est baptisé « HDK-T ».

Algorithme 3 : Apprentissage de représentations pour la prédiction de diffusion (HDK - Heat Diffusion Kernel)

Entrées :

U : ensemble des utilisateurs ; $\mathcal{D}$: ensemble des épisodes de diffusion;
d : nombre de dimensions ; α : pas d'apprentissage;
$freq$: fréquence des tests de convergence;

Sorties :

$\mathcal{Z} = \{\forall u_i \in U : z_i \in \mathbb{R}^d\}$;

pour $u_i \in U$ **faire**
  Tirage uniforme de $z_i \in [-1, 1]^d$;
fin
$oldL \leftarrow +\infty$;
$\tau \leftarrow 0$;
nbTriplets $\leftarrow \sum_{D\in\mathcal{D}} \sum_{u_i \in U^D_\infty} |\bar{U}_{t_i^D+1}|$;
tant que *vrai* **faire**
  Tirer $D \in \mathcal{D}$;
  Tirer $u_i \in U^D_\infty$;
  Tirer $u_j \in U$ avec $t_i^D < t_j^D$ ou $u_j \in \bar{U}^D_\infty$;
  $d_i \leftarrow ||z_{s_D} - z_i||^2$;
  $d_j \leftarrow ||z_{s_D} - z_j||^2$;
  $\delta_d \leftarrow d_j - d_i$;
  $\beta \leftarrow \frac{|D|\times|U^D_\infty|\times|\bar{U}^D_{t_i^D+1}|}{\text{nbTriplets}}$;
  si $\delta_d < 1$ **alors**
    $z_i \leftarrow z_i + \alpha \times \beta \times 2(z_{s_D} - z_i)$;
    $z_j \leftarrow z_j + \alpha \times \beta \times 2(z_j - z_{s_D})$;
    $z_{s_D} \leftarrow z_{s_D} + \alpha \times \beta \times 2(z_i - z_j)$;
  fin
  si $\tau \ mod \ freq = 0$ **alors**
    $L \leftarrow \mathcal{L}_{\text{rang}}(\mathcal{Z})$;
    si $L \geq oldL$ **alors**
      retourner $\mathcal{Z}$;
    fin
    $oldL \leftarrow L$;
  fin
  $\tau \leftarrow \tau + 1$;
fin

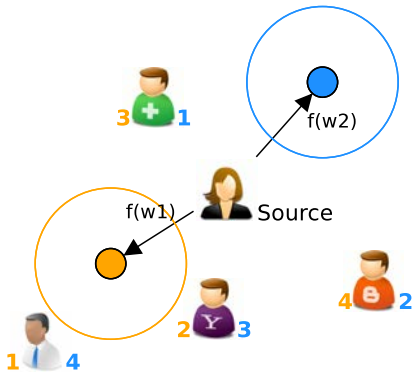


FIGURE 6.2 – Prise en compte du contenu, version HDK-T. Une même source partage deux contenus $w1$ et $w2$. Ceux-ci sont projetés avec la fonction f, et leurs représentations sont *translatées* dans l'espace suivant $f(w1)$ et $f(w2)$. Les indices près des utilisateurs indiquent dans quel ordre ils sont contaminés par chaque contenu.

6.2.3.2 Déformation

Dans cette deuxième version, les projections de tous les utilisateurs sont modifiées de la façon suivante : la x-ième composante $z_i^{(x)}$ de chaque représentation z_i est multipliée par la x-ième composante du vecteur $f_\theta(w_D)$. La nouvelle représentation z'_i de chaque utilisateur s'écrit donc :

$$\forall x \in [\![0, d-1]\!], z_i'^{(x)} = z_i^{(x)}.\left(f_\theta(w_D)\right)^{(x)}$$

Cette expression peut s'écrire :

$$z'_i = z_i.(I^{f_\theta(w_D)})$$

où $I^{f_\theta(w_D)}$ est la matrice diagonale de taille $d \times d$ dont les valeurs sont égales à celles du vecteur $f_\theta(w_D)$. Il s'agit donc d'une *matrice de changement d'échelle*. Cette modification des représentations des utilisateurs conduit à la définition de K suivante :

$$K^{\text{morph}}_{\mathcal{Z},\theta}(w_D, t, s_D, u_i) = (4\pi t)^{-\frac{d}{2}} \exp\left(-\frac{||z_{s_D}.I^{f_\theta(w_D)} - z_i.I^{f_\theta(w_D)}||^2}{4t}\right)$$

L'application de ce changement d'échelle revient donc à considérer que la diffusion d'information ne se fait plus de façon uniforme dans $\mathbb{R}^d$, mais à une vitesse différente le long de chaque dimension : plus la valeur de la x-ième composante du vecteur $f_\theta(w_D)$ est élevée, plus la diffusion est lente le long de la x-ième dimension de l'espace de représentation.

Une illustration du principe est donnée en figure 6.3. Ce modèle est baptisé « HDK-M ».

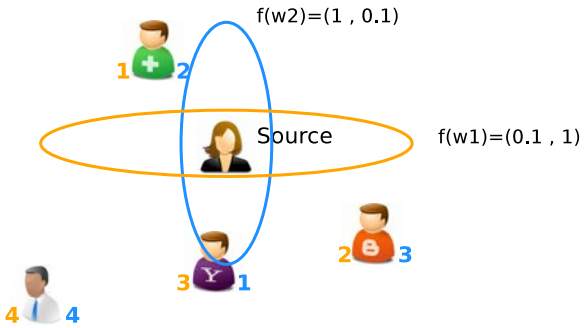


FIGURE 6.3 – Prise en compte du contenu, modèle HDK-M. Les deux contenus $w1$ et $w2$ ne se diffusent pas à la même vitesse sur les différentes dimensions de l'espace de représentation : le contenu $w1$ se difuse plus vite sur l'axe horizontal, et le contenu $w2$ se diffuse plus vite sur l'axe vertical. Les indices près des utilisateurs indiquent dans quel ordre ils sont contaminés par chaque contenu.

6.2.4 Version asymétrique

Enfin, de la même façon que dans le chapitre 5, nous pouvons également définir une version « asymétrique » du modèle, en apprenant *deux* projections z_i et ω_i par utilisateur, permettant de modéliser son comportement en tant que source et son comportement en tant que récepteur, respectivement. La fonction K s'écrit alors :

$$K^{\text{asym}}_{\mathcal{Z},\Omega}(t, s_D, u_i) = (4\pi t)^{-\frac{d}{2}} e^{-\frac{||z_{s_D} - \omega_i||^2}{4t}} \tag{6.12}$$

Les projections des utilisateurs sont apprises en appliquant le même algorithme à cette définition de $K_{\mathcal{Z}}$. Des versions asymétriques peuvent également être définies de la même façon pour les extensions prenant en compte le contenu, HDK-T et HDK-M.

6.3 Expériences

Dans cette section, nous évaluons les différentes versions du modèle HDK et de ses extensions en réalisant plusieurs expériences sur des données réelles et artificielles. Nous nous comparons aux modèle DAIC (chapitre 3) et IC projeté (chapitre 5) en utilisant la mesure MAP sur un ensemble d'épisodes de diffusion de test. Nous effectuons également quelques évaluations empiriques, décrites en fin de section.

6.3.1 Données artificielles

Nous commençons par étudier le comportement du modèle HDK sur des données artificielles. Pour cela, nous considérons que le contenu d'une information est constitué d'un

Modèle \ Q			5 mots	20 mots	40 mots
DAIC			0,30	0,16	0,11
IC proj.			0,30	0,17	0,11
Version	Symétrie	d			
HDK	Symétrique	5	0,13	0,11	0,08
		40	0,21	0,15	0,09
		80	0,21	0,15	0,09
	Asymétrique	5	0,15	0,12	0,09
		40	0,29	0,19	0,10
		80	0,28	0,18	0,10
HDK-T	Symétrique	5	0,15	0,12	0,09
		40	0,26	0,20	0,13
		80	0,25	0,18	0,12
	Asymétrique	5	0,18	0,14	0,10
		40	0,35	0,23	0,14
		80	0,34	0,24	0,14
HDK-M	Symétrique	5	0,16	0,12	0,08
		40	0,40	0,27	0,13
		80	0,46	0,31	0,17
	Asymétrique	5	0,19	0,15	0,10
		40	**0,58**	**0,38**	0,16
		80	**0,58**	**0,38**	**0,21**

TABLE 6.1 – Valeurs de MAP obtenues par les différents modèles sur les données artificielles. Les performances de notre modèle sont données pour plusieurs tailles d de l'espace de représentation, en version symétrique ou asymétrique, et avec ou sans prise en compte du contenu.

seul mot parmi un dictionnaire de taille Q. Pour chacune des Q valeurs possibles de w_D, nous construisons un graphe aléatoire invariant d'échelle sur une population de 1000 utilisateurs, en utilisant l'algorithme de Barabási-Albert. Nous tirons ensuite, pour chaque arête du graphe, une probabilité de transmission uniformément entre 0 et 0.1. Ces graphes pondérés sont ensuite utilisés pour générer 10000 épisodes d'apprentissage et 10000 épisodes de test. Chaque épisode est généré en tirant une source $u_i \in U$ et un contenu w_D composé d'un mot parmi Q, puis en simulant une diffusion selon le modèle IC sur le graphe correspondant à w_D. La procédure est répétée pour différentes valeurs de Q. Remarquons qu'une valeur de $Q = 1$ correspond à des épisodes de diffusion artificiels générés selon un seul modèle IC.

Les résultats sont donnés en table 6.1. Tout d'abord, nous pouvons voir que les performances de tous les modèles décroissent quand la variance du contenu (nombre de mots considérés) augmente. La tâche devient en effet de plus en plus difficile. Toutefois, nous pouvons constater que notre modèle est bien plus robuste à cette augmentation de la complexité : ses performances diminuent moins vite que celles des modèles itératifs.

Nous constatons également que la prise en compte du contenu, qu'il s'agisse de la version HDK-T ou HDK-M, permet à notre modèle d'obtenir de meilleurs résultats et de gagner encore en robustesse. Le modèle HDK-M est bien meilleur que tous les autres, car la façon dont il prend en compte le contenu modélise bien la façon dont les données artificielles ont été générées.

De plus, nous remarquons dans ce tableau que la version asymétrique permet d'obtenir de meilleurs résultats, les données étant générées par un modèle IC asymétrique ($p_{i,j} \neq p_{j,i}$). Pour d fixé, la version asymétrique des différentes versions de HDK dispose certes de plus de paramètres, mais ce surplus n'explique pas tous les gains en performances, comme nous pouvons le vérifier en comparant les résultats des modèles symétriques pour $d = 80$ à ceux des modèles asymétriques pour $d = 40$.

Enfin, nous pouvons voir que d'une façon générale, augmenter le nombre de dimensions de l'espace de représentation permet d'améliorer les performances de notre approche. Cependant, dans certains cas, celles-ci stagnent ou diminuent légèrement lors du passage d'un même modèle de 40 à 80 dimensions, en particulier sur les valeurs de Q plus faibles, c'est à dire quand d devient trop élevé par rapport à la complexité du problème. Il s'agit d'un phénomène de surapprentissage (nous n'avons pas observé de stagnation des performances sur les données d'apprentissage).

6.3.2 Données réelles

Nous évaluons a présent notre approche sur les corpus Digg, Twitter et ICWSM présentés dans les chapitres précédents (page 83). Pour chaque épisode de diffusion D, nous extrayons une représentation w_D du contenu de la façon suivante :

Digg : le contenu w_D est donné par la « catégorie » à laquelle appartient l'information se diffusant. Cette information est fournie directement par l'API utilisée pour récupérer les données. Dix catégories existent : technologies, business, politique, international, "lifestyle", sciences, divertissements, insolite, jeux vidéo et sports. Le vecteur $w_D \in \mathbb{R}^{10}$ a donc une seule composante à 1, les autres étant à 0, de la même façon que sur les données artificielles.

Twitter et ICWSM : le vecteur w_D est obtenu au moyen d'une représentation en sac de mots des messages faisant partie de D, sur un dictionnaire de 2000 mots.

La table 6.2 présente les statistiques des corpus.

	|U|	|D|	Densité	Longueur moyenne
Digg	6424	31861	0.04	9.57
ICWSM	2270	19027	0.001	2.22
Twitter	4088	18636	0.08	7.8

TABLE 6.2 – Statistiques de jeux de données.

Modèle \ Corpus			Twitter	Digg	ICWSM
DAIC			0,105	0,527	**0,785**
IC proj.			**0,111**	**0,545**	**0,785**
Version	Symétrie	d			
HDK	Symétrique	10	0,054	0,373	0,766
		50	0,083	0,510	0,780
		100	0,087	0,507	0,775
	Asymétrique	10	0,065	0,457	0,780
		50	0,092	0,535	0,776
		100	0,096	0,530	0,771
HDK-T	Symétrique	10	0,053	0,488	0,744
		50	0,090	0,539	0,773
		100	**0,101**	0,530	**0,785**
	Asymétrique	10	0,057	0,487	0,735
		50	0,093	**0,551**	0,768
		100	**0,101**	0,546	**0,785**
HDK-M	Symétrique	10	0,061	0,371	0,760
		50	0,080	0,509	0,782
		100	0,088	0,511	0,783
	Asymétrique	10	0,065	0,460	0,768
		50	0,092	0,537	0,782
		100	0,093	0,520	0,781

TABLE 6.3 – Valeurs de MAP obtenues sur les corpus réels. Les valeurs en gras indiquent les meilleures MAP obtenues par les modèles itératifs d'une part et par notre approche d'autre part.

Les résultats sont donnés en table 6.3. Nous pouvons voir que d'une façon générale, les modèles HDK obtiennent des résultats très proches de ceux des modèles itératifs, et même légèrement supérieurs sur le corpus Digg. L'approche prédictive présentée dans ce chapitre est toutefois bien plus rapide que les modèles itératifs, puisque la prédiction se réalise en calculant simplement des distances entre les utilisateurs, là où les modèles itératifs utilisent une méthode d'estimation de type Monte-Carlo. À titre d'exemple, HDK met quelques minutes à inférer tous les scores sur les épisodes de test du corpus *Digg*, alors que les modèles itératifs mettent près d'une heure pour traiter le même volume de données[7]. Cette vitesse d'inférence peut être importante pour certaines applications « en ligne », c'est à dire quand la prédiction doit être réalisée en temps réel.

De toutes les versions d'HDK que nous testons, une semble se dégager : le modèle HDK-T en version asymétrique. En comparant plus précisément les différentes versions du modèle

7. Les expérimentations reportées ici ont été effectuées sur un processeur *Intel Core i7 CPU 950@3.07GHz* avec 16 gigas de mémoire RAM.

HDK, nous arrivons globalement à des conclusions similaires à celles des corpus artificiels :

1. L'augmentation du nombre de dimensions de l'espace de représentation permet d'améliorer les performances jusqu'à un certain point. Ainsi, dans certains cas, les performances diminuent légèrement entre $d = 50$ et $d = 100$. C'est assez souvent le cas sur Digg.
2. L'apprentissage de *deux* représentations par utilisateur (versions asymétriques des modèles) donne de meilleurs résultats. Comme sur les données artificielles, cela peut être partiellement expliqué par l'augmentation du nombre de paramètres alloués à la modélisation de chaque utilisateur. Cependant, dans la plupart des cas, les résultats des versions symétriques pour $d = 100$ restent moins bons que ceux des versions asymétriques pour $d = 50$. Cela confirme l'idée selon laquelle la modélisation de deux comportements par utilisateur est importante pour la prédiction de la diffusion.
3. L'intégration du contenu permet d'améliorer les résultats du modèle HDK. Cette amélioration est plus ou moins importante selon le corpus. Sur Digg, le contenu est une catégorie parmi dix, indiquée sur le site. Ce contenu n'est donc pas bruité et est nécessairement pertinent et informatif. C'est donc sur ce corpus que nous observons la meilleure amélioration liée à la prise en compte du contenu. Sur Twitter et ICWSM, le contenu est beaucoup plus bruité, et il est bien plus difficile d'en extraire des informations pertinentes avec une représentation en sac-de-mots. L'amélioration des performances est donc plus limitée sur ces corpus.

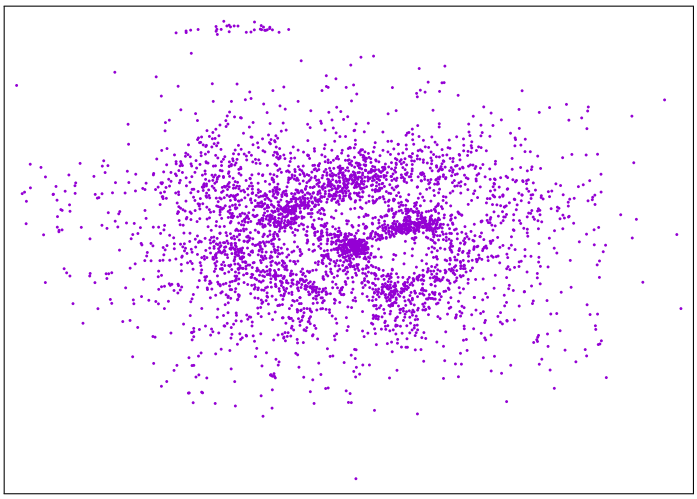
6.3.3 Résultats empiriques

Afin de mieux étudier le comportement de notre modèle HDK, nous réalisons dans cette sous-section quelques évaluations empiriques.

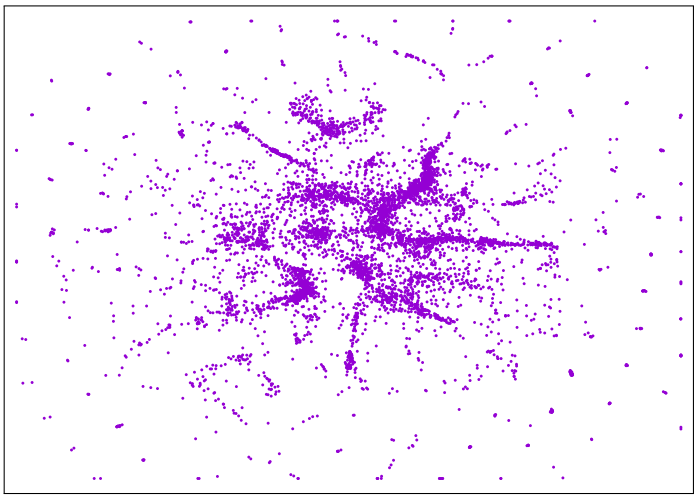
6.3.3.1 Visualisation des projections

La figure 6.4 montre les représentations des utilisateurs des trois corpus apprises dans des espaces à deux dimensions. Nous pouvons voir que notre modèle a tendance à former des groupes d'utilisateurs similaires. De plus, ces regroupements semblent d'autant plus marqués que les résultats obtenus sur le corpus correspondant sont bons.

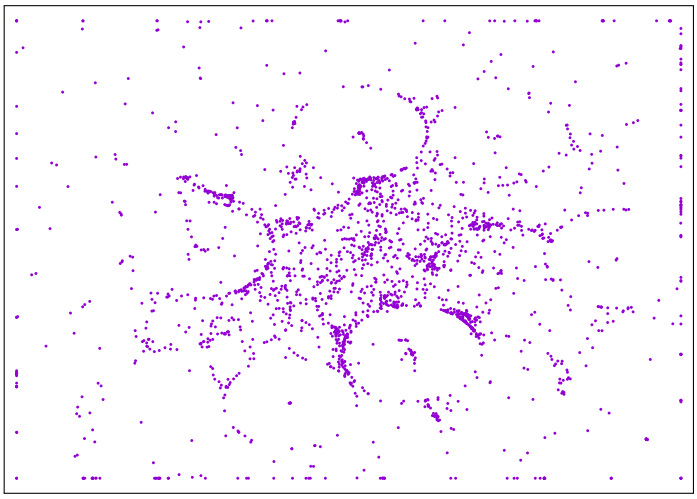
Ainsi, ce phénomène est particulièrement fort sur le corpus ICWSM, avec des groupes de points entourés par des grandes marges circulaires pratiquement vides. Ces points correspondent à des groupes de sites interagissant quasi-exclusivement entre eux, qui se retrouvent donc isolés des autres dans l'espace de représentation. Vers le centre de la figure se trouvent d'autres groupes de points moins marqués. À la périphérie de l'espace, nous retrouvons des utilisateurs participant à très peu d'épisodes et qui sont donc éloignés des



(a) Twitter



(b) Digg



(c) ICWSM

FIGURE 6.4 – Représentations des utilisateurs apprises par HDK dans des espaces à deux dimensions.

autres jusqu'à saturation. Sur le corpus Digg, nous n'observons pas ce type de structures. En revanche, nous pouvons repérer divers regroupements assez marqués. Enfin, sur le corpus Twitter (où nous obtenons les moins bons résultats en MAP), les groupements d'utilisateurs sont peu nombreux et très peu marqués.

Ces observations permettent d'envisager d'autres utilisations de notre approche, puisque l'espace de représentation semble pouvoir servir de base à plusieurs applications, en particulier la détection de communautés d'utilisateurs.

6.3.3.2 Impact du contenu

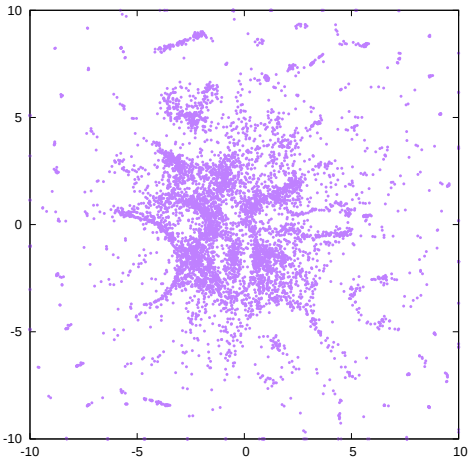
Ce type de visualisation nous permet également d'étudier l'impact du contenu sur le corpus Digg, où la représentation de ce contenu est assez simple (une catégorie parmi dix). Rappelons que les modèles HDK-T et HDK-M apprennent les paramètres θ d'une fonction linéaire $f_\theta(w_D) = w_D.\theta$ permettant d'obtenir une représentation du contenu dans $\mathbb{R}^d$. Les paramètres θ correspondent donc à une matrice de taille $Q \times d$, où chaque ligne θ_i peut être interprétée comme une représentation d'un mot du corpus (ou d'une catégorie dans le cas de Digg) dans $\mathbb{R}^d$

La figure 6.5 montre les représentations des utilisateurs dans un espace à deux dimensions apprises par le modèle HDK-M symétrique, ainsi que les valeurs du changement d'échelle θ_i appris pour chacune des dix catégories du corpus Digg. Quelques exemples de l'effet du changement d'échelle sont donnés en bas de la figure Nous pouvons voir que la plupart des catégories ont pour effet une déformation assez importante. De plus, ces catégories peuvent être séparées en deux groupes suivant l'orientation de leurs déformations :

— *international*, *business*, *politique*, *technologies*, *lifestyle* et *sciences* déforment l'espace horizontalement ;
— *sports*, *jeux vidéo*, *insolite* et *divertissement* déforment l'espace verticalement.

Nous pouvons remarquer que ces deux groupes correspondent à une division entre d'une part les sujets plus « sérieux », et d'autre part des sujets plus légers.

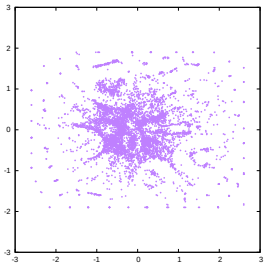
Sur le corpus Twitter, nous ne pouvons pas visualiser le contenu ainsi. Toutefois, en nous intéressant à la version HDK-T apprise avec $d = 100$, nous pouvons calculer la liste des mots dont les représentations θ_i ont les normes $||\theta_i||^2$ les plus élevées. Dans le cadre du modèle HDK-T, il s'agit des mots ayant l'impact le plus important sur la diffusion, car ils tendent à déplacer la représentation de la source bien plus loin. La liste de ces mots est donnée en table 6.4. Nous pouvons constater qu'il s'agit exclusivement de mots appartenant au champ lexical de la politique, ce qui n'est pas surprenant étant donné que nous avons collecté ce corpus pendant la campagne présidentielle américaine de 2012. La politique n'est pas l'unique sujet discuté au sein de ce corpus, mais ces résultats laissent penser qu'il s'agit de celui impactant le plus la diffusion, de la même façon que sur le corpus Digg, où la catégorie *politique* est l'une de celles appliquant la plus forte déformation de l'espace.



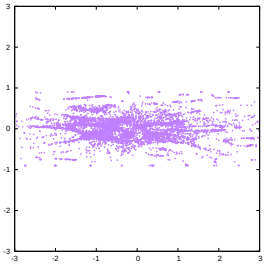
(a) Positions normales des utilisateurs

Catégorie	Déformation	
	x	y
Technologies	0,26	0,19
Business	0,43	0,09
Politique	10,0	0,01
International	2,41	0,21
Lifestyle	0,26	0,21
Sciences	0,59	0,01
Divertissements	0,01	2,03
Insolite	0,1	0,33
Jeux Vidéo	0,01	9,99
Sports	0,02	9,99

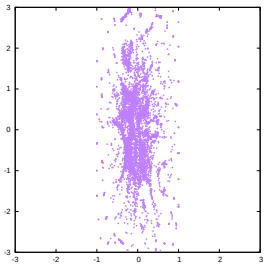
(b) Déformations des catégories



(c) Technologies



(d) Business



(e) Insolite

FIGURE 6.5 – Modèle HDK-M symétrique en deux dimensions sur Digg. Les projections normales des utilisateurs sont déformées selon un vecteur correspondant au contenu associé. Quelques exemples de déformations sont donnés.

6.4 Conclusion

Dans ce chapitre, nous avons proposé une approche prédictive du problème de prédiction de diffusion. Au lieu de baser notre modèle sur l'inférence, durant l'apprentissage, d'une information manquante (les probabilités de transmission), nous avons proposé un algorithme visant à reproduire uniquement l'*ordre* dans lesquels les utilisateurs étaient infectés. Cet algorithme est basé sur l'apprentissage d'un ensemble de représentations des utilisateurs et le calcul des distances les séparant, ce qui permet d'obtenir un modèle beaucoup plus rapide en inférence, en plus de régulariser les relations entres eux comme dans le chapitre 5. De plus, nous avons proposé des extensions permettant de prendre facilement en compte le contenu de l'information diffusée.

Les résultats sur des données réelles et synthétiques nous ont montré que cette approche obtenait des résultats comparables à ceux des modèles itératifs. De plus, nous avons vu que l'intégration du contenu permettait d'améliorer les performances, lorsque celui-ci était assez pertinent.

Norme	Mot
252	Ohio
250	GOP
229	Romney
222	Poll
221	TeaParty
217	Obama2012
196	Sandy
192	Voter
187	Obama
186	Vote

TABLE 6.4 – Liste des mots ayant les représentations θ_i les plus grandes, sur le corpus Twitter.

Bien que les résultats finaux soient proches de ceux obtenus dans le chapitre 5, le gain en complexité temporelle de l'inférence peut s'avérer très important pour certaines applications. Cette propriété nous a notamment encouragé à étudier la tâche de *détection de source* avec le même type de modèle, qui fait l'objet du chapitre suivant.

Chapitre 7

Détection de source

Résumé Ce chapitre détaille notre dernière contribution [Bourigault et al., 2016a], qui concerne le problème de la détection de source. Nous apprenons des représentations des utilisateurs dans $\mathbb{R}^d$ de façon à ce que la représentation de la source d'un épisode de diffusion soit proche de la représentation de l'épisode lui-même. Des expériences sur des données réelles montrent que le modèle obtenu est à la fois meilleur et plus rapide que ceux présentés dans le chapitre 2, basés sur le graphe. Nous présentons également une extension permettant de prendre en compte le *contenu* de l'information se diffusant, ainsi qu'une extension permettant d'apprendre l'importance de chaque utilisateur dans la détection.

7.1 Introduction

7.1.1 Problème

Dans ce chapitre, nous nous intéressons à la tâche de *détection de source*, qui est la tâche inverse de celle de *prédiction de diffusion*. Le but est de retrouver, à partir du résultat de la diffusion, l'utilisateur ayant créé une information donnée. La principale application concrète de ce problème est de repérer la source d'une fausse information ou d'une fuite. En effet, sur beaucoup de réseaux sociaux en ligne, les contenus ne sont pas modérés *a priori*. De nombreuses rumeurs ou fausses informations se propagent donc facilement [Sénécat, 2016]. De plus, les réseaux sociaux sont en général le premier lieu où les *fuites* ont lieux. Par exemple, il arrive qu'un film ou un épisode de série télévisée soit diffusé sur internet avant sa date de sortie officielle, en général par un utilisateur ayant accès à une version presse. Dans ce genre de situation, les producteurs chercheront à savoir d'où provient la fuite afin de ne plus lui fournir de version presse à l'avenir [Hooton, 2015]. Tous ces phénomènes ont motivé un certain nombre de travaux sur le sujet, présentés dans le chapitre 2, section 2.8.

7.1.2 Limites des approches existantes

Pratiquement tous les modèles décrits dans l'état de l'art partagent les mêmes principes généraux.

1. Ils considèrent que le graphe du réseau social est connu ;
2. Ils font l'hypothèse que la diffusion d'information suit un modèle de diffusion fixé, au sein de ce graphe. Il peut s'agir d'un modèle SI, d'une extension temporelle du modèle SI [Shah and Zaman, 2010], d'un modèle IC [Lappas et al., 2010], ou d'un modèle continu comme NetRate ou CTIC [Farajtabar et al., 2015, Pinto et al., 2012].
3. Ils utilisent un estimateur de type maximum de vraisemblance : quand ils observent le résultat d'une diffusion, ils cherchent l'utilisateur source maximisant la vraisemblance de l'observation, sous l'hypothèse du modèle de diffusion considéré.

En d'autres termes, ces approches *inversent* des modèles de diffusion classiques. Cela pose plusieurs problèmes :

1. La qualité de la prédiction dépend entièrement de la pertinence du modèle de diffusion considéré, et du graphe utilisé. Nous avons déjà parlé dans les chapitres précédents des problèmes liés à l'utilisation d'un graphe fixé a priori : données manquantes, non-pertinence des liens explicites, etc...
2. L'estimation de la source la plus probable $\hat{s}$ est coûteuse en calcul. Il est souvent nécessaire, pour trouver $\hat{s}$, de calculer la longueur des plus courts chemins entre toutes les paires d'utilisateurs du graphe.

De plus, les modèles de détection de source décrits dans le chapitre 2 sont la plupart du temps testés sur des données synthétiques, i.e. des épisodes de diffusion générés *par le modèle de diffusion utilisé en prédiction*. Bien que les résultats ainsi obtenus soient intéressants, seules des expériences sur des épisodes de diffusion réels permettraient de vraiment rendre compte des capacités de ces modèles. Or, seuls de rares travaux [Pinto et al., 2012, Farajtabar et al., 2015] s'évaluent sur des épisodes de diffusion réels, et les résultats obtenus sont largement inférieurs à ceux obtenus sur des données synthétiques. En particulier, dans [Farajtabar et al., 2015], certains modèles basés sur les graphes obtiennent des résultats nuls sur des épisodes de diffusion réels, et ne parviennent à identifier la source qu'en observant *plusieurs* épisodes de diffusion commençant par le même utilisateur.

Nous proposons donc dans ce chapitre d'appliquer l'apprentissage de représentations à ce problème. Cela nous permet, comme dans les chapitres précédents, d'obtenir un modèle plus compact, bien plus rapide, et de régulariser les relations entre utilisateurs sans nous limiter à un graphe fixé. De plus, nous proposons plusieurs extensions permettant de prendre en compte l'importance des utilisateurs et le contenu de l'information se propageant. Cette approche est testée sur des épisodes de diffusion réelles et artificielles.

7.2 Apprentissage de représentations pour la détection de source.

7.2.1 Modèle

Soit D un épisode de diffusion. Le premier utilisateur infecté dans celui-ci est noté s_D, et correspond à la source de l'information considérée. Nous notons $\hat{U}^D = U_\infty^D \setminus \{s_D\}$ l'ensemble des utilisateurs infectés dans D privé de la source. Notre but dans ce chapitre est donc de **retrouver s_D à partir de $\hat{U}^D$.**

Pour cela, nous apprenons deux projections z_i et ω_i pour chaque utilisateur u_i, modélisant respectivement son comportement en tant que source et son comportement en tant que récepteur de contenu. Ces projections sont apprises en suivant le principe suivant :

> La représentation z_{s_D} de l'utilisateur s_D devrait être située au point $z_D = \phi(\hat{U}^D)$, qui correspond à une représentation de l'épisode de diffusion D, calculée à partir des projections ω_i des utilisateurs de $\hat{U}^D$.

Plusieurs définitions de $\phi : 2^U \to \mathbb{R}^d$ sont possibles[8]. Nous choisissons d'utiliser une moyenne, qui a l'avantage d'être rapide à calculer :

$$z_D = \phi(\hat{U}^D) = \frac{1}{|\hat{U}^D|} \sum_{u_i \in \hat{U}^D} \omega_i \tag{7.1}$$

Cette définition présente également l'avantage d'être relativement stable par rapport aux utilisateurs manquants : en effet, pour $|\hat{U}^D|$ suffisamment grand, $\forall u_i \in U : \phi(\hat{U}^D \cup \{u_i\}) \approx \phi(\hat{U}^D)$. Cela permet à la représentation de rester pertinente dans le cas où le modèle manipule des épisodes de diffusion incomplets. Une illustration de ce principe (avec une seule projection par utilisateur) est donnée en figure 7.1, où la source de l'épisode D est projetée près du centre des utilisateurs $\hat{U}^D$. Cette représentation z_D peut, d'une certaine façon, être vue comme la source de l'information dans l'espace de représentation, de la même façon que dans le chapitre 6. Les deux modèles ne sont toutefois pas équivalents (cf fonction objectif 7.3).

Pour retrouver la source d'un épisode de diffusion D étant donné $\hat{U}^D$, le modèle recherche l'utilisateur u_i dont la projection-source z_i est la plus proche de la représentation z_D :

$$\hat{s} = \underset{u_i \in U \setminus \hat{U}^D}{\arg\min} ||z_i - z_D||^2 \tag{7.2}$$

où z_D est calculée selon la formule 7.1 appliquée aux utilisateurs de $\hat{U}^D$. Afin d'apprendre les ensembles de projections $\mathcal{Z} = (z_i)_{u_i \in U}$ et $\Omega = (\omega_i)_{u_i \in U}$ de façon à ce que la formule

8. Rappelons que 2^U désigne l'ensemble des parties de U

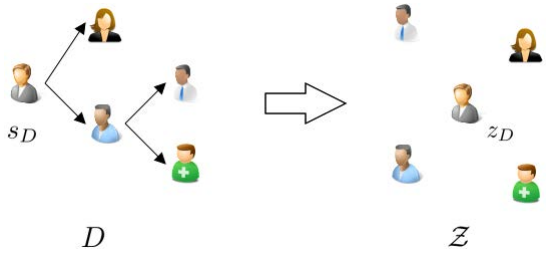


FIGURE 7.1 – Les utilisateurs de l'épisode de diffusion D sont projetés de façon à ce que la source se trouve au centre des représentations des utilisateurs infectés.

.

7.2 soit valide, nous minimisons la fonction de coût suivante :

$$\begin{aligned}\mathcal{L}(\mathcal{Z}, \Omega; \mathcal{D}) &= \sum_{D \in \mathcal{D}} \sum_{\substack{u_i \neq s_D \\ u_i \notin \hat{U}^D}} h\left(||z_i - z_D||^2 - ||z_{s_D} - z_D||^2\right) \\ &= \sum_{D \in \mathcal{D}} \sum_{u_i \notin U^D} h\left(||z_i - z_D||^2 - ||z_{s_D} - z_D||^2\right)\end{aligned} \tag{7.3}$$

où h est une fonction hingeloss : $h(x) = \max(1 - x, 0)$. $\mathcal{L}$ est donc une fonction de coût d'ordonnancement « paire à paire » exprimant le fait que la représentation-source de s_D, notée z_{S_D}, doit être plus proche de la représentation de D (second terme de la soustraction) que les représentations-sources des autres utilisateurs (premier terme de la soustraction) de façon à être celle qui serait prédite par la fonction 7.2.

Ce coût peut être minimisé en utilisant une descente de gradient stochastique proche de celle du chapitre 6. Celle-ci est détaillée dans l'algorithme 4. Nous commençons par initialiser toutes les projections au hasard (lignes 2 et 3). Puis, à chaque itération, nous tirons un épisode D (ligne 9) et un utilisateur « non-source » u_j ne faisant pas partie de U^D_∞ (ligne 10). Si la projection z_{s_D} de la vraie source de D n'est pas plus proche de la représentation z_D que z_j avec une marge de 1 (ligne 15), toutes les projections concernées (i.e les représentation récepteurs des utilisateur de $\hat{U}^D$, ainsi que z_j et z_{s_D}) sont mises à jour avec un pas de gradient (lignes 16 à 19). Ce pas de gradient rapproche la représentation z_D de z_{s_D} et l'éloigne de z_j. L'apprentissage continue jusqu'à convergence, qui est testée en observant l'évolution de la valeur de $\mathcal{L}$ toutes les F itérations (ligne 24).

De la même façon que dans les chapitres précédents, le tirage aléatoire réalisé en lignes 9 et 10 introduit un biais dans l'apprentissage, les utilisateurs n'apparaissant *pas* dans les épisodes plus longs étant considérés plus souvent. Chaque terme de la double somme 7.3 est tiré avec une probabilité :

$$\frac{1}{|\mathcal{D}| \times (N - |U^D|)}$$

Algorithme 4 : Apprentissage de représentations pour la détection de source

Entrées :

U : Ensemble d'utilisateurs ;
$\mathcal{D}$: Ensemble d'apprentissage ;
d : Nombre de dimensions
ϵ : Pas de gradient ;
F : Fréquence des tests de convergences ;

Sorties :

$Z = \{\forall u_i \in U : z_i \in \mathbb{R}^d\}$; $\Omega = \{\forall u_i \in U : \omega_i \in \mathbb{R}^d\}$;

pour $u_i \in U$ **faire**
  Initialiser z_i aléatoirement, de façon uniforme sur $[-1, 1]^d$
  Initialiser ω_i aléatoirement, de façon uniforme sur $[-1, 1]^d$
fin
$it \leftarrow 0$;
$oldL \leftarrow 0$;
nbTermes $\leftarrow \sum_{D \in \mathcal{D}} \left(N - |U^D|\right)$
tant que *true* **faire**
  Tirer un épisode $D \in \mathcal{D}$;
  Tirer $u_j \notin U^D$;
  Calculer z_D suivant la formule 7.1 ;
  $d_s \leftarrow ||z_{s_D} - z_D||^2$;
  $d_j \leftarrow ||z_j - z_D||^2$;
  $\beta \leftarrow \frac{|\mathcal{D}| \times (N - |U^D|)}{\text{nbTermes}}$;
  si $d_j - d_s < 1$ **alors**
    $z_{s_D} \leftarrow z_{s_D} - \epsilon \times \beta \times 2\left(z_{s_D} - z_D\right)$;
    $z_j \leftarrow z_j + \epsilon \times \beta \times 2\left(z_j - z_D\right)$;
    pour $u_x \in \hat{U}^D$ **faire**
      $\omega_x \leftarrow \omega_x - \epsilon \times \beta \times \frac{2}{|\hat{U}^D|}\left(z_j - z_{s_D}\right)$
    fin
  fin
  si $it \mod F = 0$ **alors**
    $L \leftarrow \mathcal{L}(\mathcal{Z}, z)$
    si $L \geq oldL$ **alors**
      retourner $(\mathcal{Z}, \Omega)$;
    fin
    $oldL \leftarrow L$;
  fin
  $it \leftarrow it + 1$
fin

Pour éviter le biais, chaque terme de la double somme 7.3 devrait être tiré avec la même probabilité, c'est à dire :

$$\frac{1}{\sum\limits_{D\in\mathcal{D}} N - |U^D|}$$

Nous calculons donc, en ligne 14, un poids β permettant de corriger ce biais :

$$\beta = \frac{|\mathcal{D}| \times (N - |U^D|)}{\sum\limits_{D\in\mathcal{D}} N - |U^D|}$$

Ce poids est appliqué lors de la mise à jour des paramètres (lignes 16 à 19).

7.2.1.1 Régularisation des projections

Dans le coût défini au dessus, les deux représentations des utilisateurs sont apprises indépendamment, pour modéliser son comportement en tant que source et en tant que récepteur. En pratique, bien que ces comportements puissent être assez différents, il est raisonnable de penser qu'ils ne sont pas décorrélés : ces deux comportements sont en effet des conséquences des centres d'intérêt de l'utilisateur [Barbieri et al., 2013a]. Pour prendre en compte cette propriété, nous ajoutons un terme de régularisation au coût :

$$\mathcal{L}_\lambda(\mathcal{Z}, \Omega; \mathcal{D}) = \sum_{D\in\mathcal{D}} \sum_{u_i \notin U^D} h\left(||z_i - z_D||^2 - ||z_{s_D} - z_D||^2\right) + \lambda \sum_{u_i} ||z_i - \omega_i||^2 \tag{7.4}$$

Le terme de régularisation favorise les projections telles que z_i et ω_i soient plus proches, suivant un hyperparamètre λ . Cette régularisation peut également améliorer les capacités de généralisation de notre modèle : sans ce terme, aucune représentation z_i ne pourrait être apprise pour un utilisateur n'apparaissant jamais en tant que source dans $\mathcal{D}$. Avec ce terme de régularisation liant les deux représentations z_i et ω_i, une partie de l'information apprise sur ω_i peut être transférée sur z_i.

7.2.2 Extensions

7.2.2.1 Modélisation de l'importance des utilisateurs

Nous présentons maintenant une première extension possible de notre modèle, consistant à apprendre pour chaque utilisateur un *poids* α_i pour redéfinir z_D ainsi :

$$z_D = \sum_{u_i \in \hat{U}^D} \frac{e^{S.\alpha_i}}{\sum\limits_{(u_j \in \hat{U}^D)} e^{S.\alpha_j}} \omega_i \tag{7.5}$$

où $S \in \mathbb{R}$ est un paramètre et où la fraction correspond à une fonction softmax permettant de transformer un vecteur de k valeurs réelles en un vecteur de $[0,1]^k$ sommant à 1. Ainsi, z_D devient un barycentre des représentations-récepteurs des utilisateurs de $\hat{U}^D$, pondérées par les valeurs α. Le poids de chaque utilisateur modélise donc son importance pour la détection de source. Par exemple, sur Twitter, certains utilisateurs ne sont que des robots, repostant automatiquement les hashtags et les tweets populaires dans le but de gagner en visibilité afin de poster des publicités. Dans ce cas, l'infection de cet utilisateur donne très peu d'information sur l'identité de la source, et le modèle pourra apprendre un poids $\alpha_i \approx 0$. De plus, autoriser le modèle à se concentrer sur les utilisateurs les plus discriminants peut aussi permettre de sélectionner les utilisateurs les plus importants dans certains contextes applicatifs, où seul un nombre réduit d'entre eux peuvent être monitorés (comme dans [Seo et al., 2012], par exemple). La valeur de S, fixée à 1 dans nos expériences, permet de modifier l'importance de l'utilisateur de poids maximum (plus S est élevé, plus le softmax se rapproche d'une fonction maximum).

7.2.2.2 Intégration du contenu

De la même façon que dans le chapitre 6, nous proposons une extension de notre modèle permettant de prendre en compte le contenu d'une information pour la détection de source. Pour cela, nous transformons la représentation z_D en fonction du contenu de l'information considérée. Nous utilisons la même idée que celle du modèle HDK-T, qui a donné les meilleurs résultats dans le chapitre 6 : le contenu d'un épisode D est représenté par un vecteur-ligne $w_D \in \mathbb{R}^Q$, et nous apprenons les paramètres $\theta \in \mathbb{R}^{Q \times d}$ d'une fonction linéaire f_θ permettant de projeter ce contenu dans $\mathbb{R}^d$.

$$f_\theta(w_D) = w_D.\theta$$

La représentation de D est alors calculée comme :

$$z_D = \left(\frac{1}{|\hat{U}^D|} \sum_{u_i \in \hat{U}^D} \omega_i \right) + f_\theta(w_D) \tag{7.6}$$

Les paramètres θ sont appris en même temps que les projections des utilisateurs, avec l'algorithme de descente de gradient appliqué à cette définition de z_D.

7.3 Expériences

Nous présentons dans cette section les résultats de plusieurs expériences effectuées sur des données réelles et artificielles, et dans plusieurs contextes expérimentaux différents. Nous évaluons aussi l'impact des deux extensions présentées dans la section précédente.

	$\|U\|$	$\|E\|$	$\|\mathcal{D}\|$	Densité
Artificiel	100	262	10000	2%
Lastfm	1984	235011	331829	5%
Weibo	5000	20784	44345	0.08%
Twitter	4107	128855	16824	1%

TABLE 7.1 – Quelques statistiques sur les corpus : nombre d'utilisateurs $|U|$, de liens dans le graphe $|E|$, d'épisodes de diffusion, et densité du graphe (voir section 7.3.1.2 pour la définition du graphe utilisé).

7.3.1 Paramètres

7.3.1.1 Corpus

Nous utilisons les corpus Lastfm et Twitter, ainsi qu'un corpus issus du site Weibo. Ce site est un service de micro-blogging similaire à Twitter, utilisé essentiellement en Chine. Le corpus est constitué de l'ensemble de l'activité du site sur une période d'un an [wa Fu et al., 2013]. Les épisodes de diffusion en sont extraits selon une méthode similaire à celle de [Gomez-Rodriguez et al., 2011].

1. Le corpus est vu comme un grand graphe hétérogène contenant deux types de nœuds : les *utilisateurs* et les *messages*.
2. Chaque message est relié à son auteur et aux messages qu'il référence par le biais de *retweets* ou de *réponses*.
3. Chaque composante connexe du sous-graphe des messages correspond ainsi à un ensembles de messages discutant d'un même sujet et s'influençant les uns les autres.
4. Les auteurs des messages de chacune de ces composantes connexes, associés aux temps auxquels ils ont posté ces messages, forment un épisode de diffusion.

Le corpus est ensuite filtré pour ne garder que 5000 utilisateurs parmi les plus actifs. De plus, nous effectuons des expériences sur un corpus artificiel généré comme suit. Nous commençons par construire un graphe aléatoire invariant d'échelle, en utilisant l'algorithme de Barabási-Albert, contenant 100 utilisateurs. Nous tirons ensuite sur les liens de ce graphe des probabilités de transmission, uniformément sur $[0, 0.1]$, et utilisons celles-ci pour générer des épisodes de diffusion avec le modèle IC. Les propriétés de ces corpus sont résumées dans la table 7.1.

7.3.1.2 Modèles de références

Nous comparons notre approche à plusieurs heuristiques ou modèles issus de la littérature, toutes basées sur le graphe du réseau social.

OutDeg : cette heuristique simple a été proposée dans [Farajtabar et al., 2015]. À partir de de $\hat{U}^D$, nous recherchons l'ensemble des « sources possibles » dans le graphe, i.e tous les utilisateurs à partir desquels il existe un chemin vers *chaque* élément de $\hat{U}^D$ dans le graphe. Ces différentes « sources possibles » sont ensuite classées par degré sortant, le plus élevé correspondant à la source la plus vraisemblable.

Centre de Jordan : l'utilisation du centre de Jordan comme estimateur de source a été étudiée dans [Luo et al., 2015a]. Notre contexte expérimental n'étant pas exactement le même que dans [Luo et al., 2015a], nous en adaptons un peu la formulation : la source prédite est celle minimisant la distance maximale à tout utilisateur infecté $\hat{U}^D$ dans le graphe.

$$\hat{s} = \arg\min_{u_i \notin \hat{U}^D} \max_{u_j \in \hat{U}^D} \text{dist}_G(u_i, u_j).$$

Pinto : le modèle décrit dans [Pinto et al., 2012]. Celui-ci est basé sur un modèle de diffusion continu avec des délais de transmission suivant une loi gaussienne, et utilise une heuristique basée sur l'extraction d'un arbre couvrant (voir le chapitre 2).

Toutes ces méthodes sont basées sur le graphe du réseau social. Comme dans les chapitres précédents, nous ne connaissons pas ce graphe dans nos corpus. Toutefois, nous ne pouvons pas utiliser le graphe des « exemples positifs » comme dans le chapitre 3, c'est à dire créer à lien (u_i, u_j) à chaque fois qu'un épisode de diffusion d'apprentissage contient l'utilisateur u_i puis l'utilisateur u_j. En effet, plusieurs modèles de référence utilisent les longueurs des plus courts chemins dans le graphe. Ces longueurs auraient peu de sens dans le graphe ainsi construit. Nous utilisons donc l'algorithme du chapitre 3 pour apprendre les paramètres d'un modèle IC à partir de l'ensemble d'apprentissage DAIC du chapitre 3. Puis, nous conservons dans le graphe les liens (u_i, u_j) tels que $p_{i,j} > S$, où S est un seuil fixé empiriquement à partir des résultats obtenus sur un ensemble de validation. C'est la densité de ce graphe qui est indiquée dans la table 7.1. Rappelons bien que ce graphe n'est utilisé que par certains modèles de référence, notre approche n'en ayant pas besoin.

7.3.1.3 Évaluation

Les performances des différents modèles sont évaluées sur un ensemble de test $\mathcal{D}'$ avec une mesure de Top-K. Celle-ci est calculée en classant les différents utilisateurs susceptibles d'être sources suivant leurs scores (vraisemblance, degré ou distance à z_D, suivant le modèle considéré). Si la vraie source s_D se trouve parmi les K utilisateurs les mieux classés, la valeur du Top-K est 1, sinon 0.

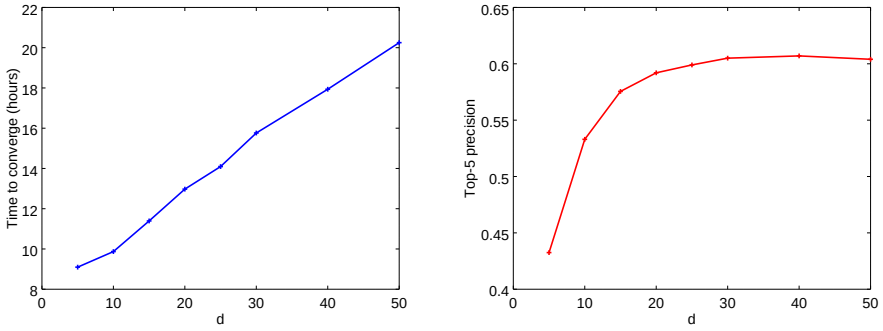


FIGURE 7.2 – Durée de l'apprentissage et performances obtenues (en Top-5) sur le corpus Weibo.

7.3.2 Résultats

7.3.2.1 Choix du nombre de dimensions

Nous commençons par reproduire l'expérience du chapitre 5 pour sélectionner le nombre de dimensions. Nous observons la durée de l'apprentissage et les performances obtenues par notre modèle pour différentes valeurs de d, sur le corpus Weibo. La figure 7.2 présente les résultats obtenus. Nous constatons que la durée de l'apprentissage croit linéairement avec d, mais que les performances du modèle stagnent à partir d'une trentaine de dimensions. Nous utiliserons donc une valeur de $d = 30$ dans toutes nos expériences.

Nous testons maintenant notre modèle dans plusieurs contextes expérimentaux différents.

7.3.2.2 Détection de source classique

Il s'agit du contexte normal : notre but est de retrouver s_D à partir de $\hat{U}^D$. Les résultats de notre modèle (noté RL, pour « representation learning ») sont donnés pour une valeur du paramètre de régularisation $\lambda = 10^{-4}$, qui nous permettait d'obtenir les meilleurs résultats sur un ensemble de validation. Les résultats sont présentés en figure 7.3.

Nous pouvons tout d'abord voir que sur le corpus artificiel, notre modèle et celui des centres de Jordan obtiennent de meilleurs résultats que les autres. Rappelons que sur ce corpus, les épisodes de diffusion sont générés selon un modèle DAIC. Comme le corpus est assez petit, la méthode d'extraction de graphe utilisée (basée sur l'apprentissage des paramètres d'un modèle IC) retrouve facilement les vrais liens du graphe à partir de $\mathcal{D}$. Dans ce contexte, le modèle Jordan obtient d'excellentes performances car il est basé sur le calcul exhaustif de toutes les distances dans le graphe. Notre approche est capable d'obtenir des résultats proches de ceux-ci, sans faire l'hypothèse d'un modèle de diffusion fixé et connu et sans utiliser ce graphe. Le modèle de Pinto, par contre, base sa prédiction

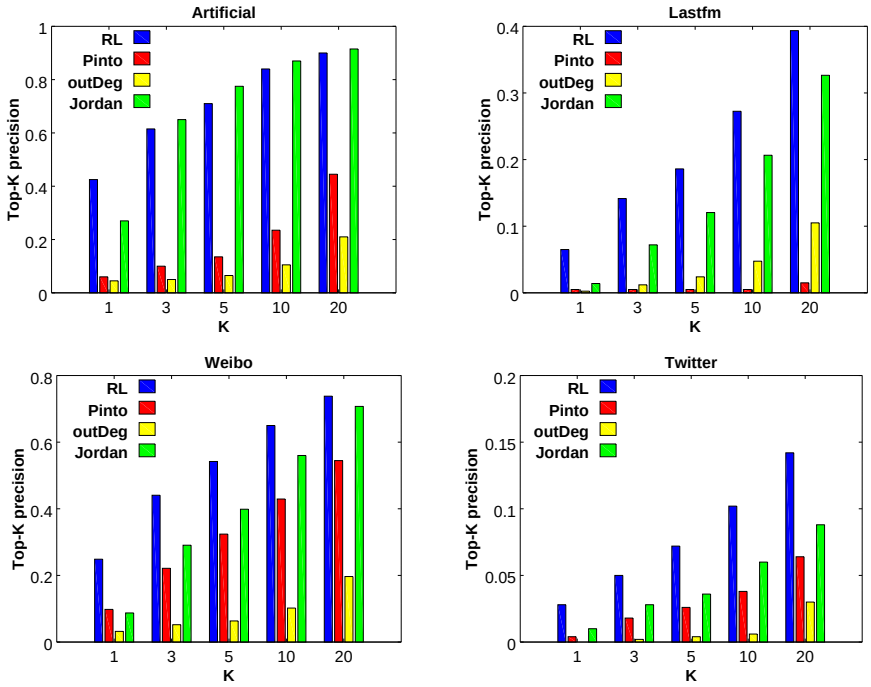


FIGURE 7.3 – Détection de source sur des épisodes de diffusion complets.

sur un arbre extrait du graphe par un parcours en largeur d'abord, et ignore donc beaucoup d'informations pertinentes, ce qui limite ses performances.

Sur le corpus Weibo, le modèle IC appris ne peut retrouver le vrai graphe de diffusion (comme nous l'avons vu dans le chapitre 5). Dès lors, les résultats des modèles Pinto et Jordan sont plus proches. En revanche, notre modèle bat tous les autres, car il ne repose pas sur une connaissance a priori de ce graphe. Le fait que le modèle Pinto soit légèrement moins bon que le modèle Jordan peut s'expliquer par le fait que le premier fait l'hypothèse que les délais de transmission suivent une loi Gaussienne, ce qui n'est pas réaliste dans des corpus réels [Farajtabar et al., 2015].

Enfin, les corpus Twitter et Lastfm sont plus difficiles : le fait que deux utilisateurs aient écouté la même chanson ou utilisé le même hashtag ne veut pas forcément dire qu'il y a eu contamination de l'un par l'autre. Dans ce contexte, le graphe extrait de $\mathcal{D}$ devient moins pertinent : il peut s'agir de liens de *corrélation* et non de *causalité*. Tous les modèles de référence étant basés sur ce graphe, ils obtiennent des résultats moins bons que ceux de notre modèle.

Notons que bien que les résultats de l'ensemble des modèles puissent sembler assez mauvais sur Twitter, ils peuvent tout de même être utilisés dans certains contextes, comme celui décrit dans [Luo et al., 2015a] : quand l'administrateur d'un réseau doit décider quels utilisateurs inspecter pour retrouver la source d'une rumeur (avec un coût associé à cette inspection), tout modèle donnant des résultats meilleurs qu'un modèle aléatoire est susceptible d'être important.

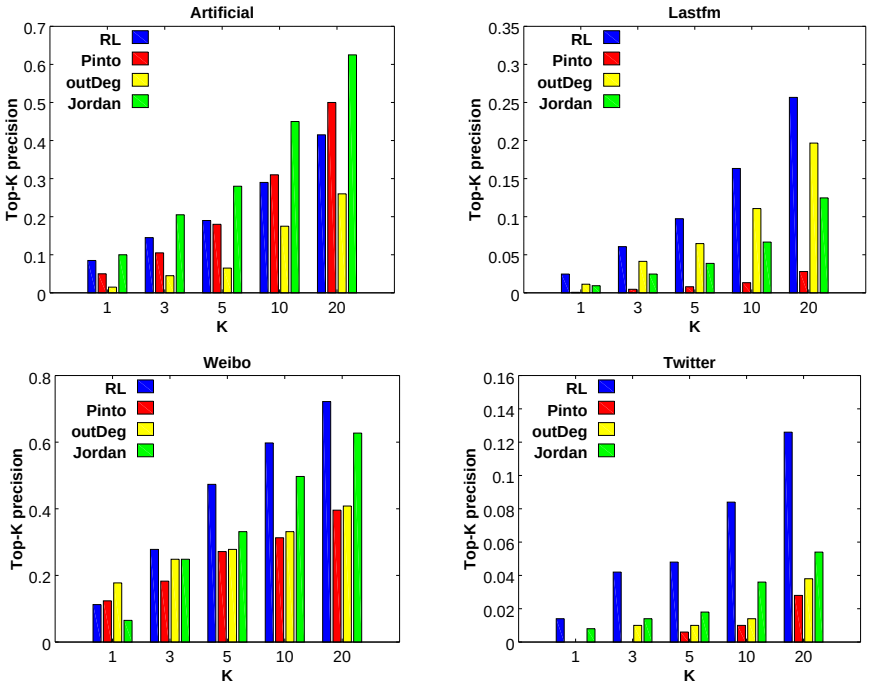


FIGURE 7.4 – Détection de source sur des épisodes de diffusion partiels (20%).

7.3.2.3 Détection de source sur des cascades partielles

Dans certaines applications réelles, il est possible que les épisodes de diffusion ne soient que partiellement observés durant la détection de source. Pour étudier l'impact de ce phénomène sur les performances, nous retirons au hasard des utilisateurs de $\hat{U}^D$ avant de réaliser la prédiction, en ne gardant que 20% de ceux-ci. Les résultats se trouvent en figure 7.4.

Sur le corpus artificiel, les performances de tous les modèles chutent clairement. Notre modèle se retrouve au même niveau que Pinto, et largement en dessous du modèle Jordan. Ici, la supériorité du modèle Jordan est due au fait qu'un faible nombre d'utilisateurs observés suffise à réduire grandement le nombre de sources possibles, le graphe étant assez creux. De plus, le calcul des plus courts chemins dans le graphe traduit bien la façon dont l'information se diffuse dans un modèle IC (les plus courts chemins correspondant souvent aux plus vraisemblables). En revanche, sur le corpus Weibo, les modèles restent assez stables, et notre approche reste meilleure.

Il est intéressant de remarquer ensuite que les résultats sur Lastfm et Twitter sont différents. Sur le corpus Lastfm, outDeg obtient de meilleurs résultats que les deux autres modèles de référence, alors que c'est le modèle des centres de Jordan qui bat les deux autres sur Twitter. Une explication possible est que sur Lastfm, les longues chaînes de diffusion sont rares, les chansons étant d'abord écoutées par des « early adopters », qui sont responsables de la plupart des infections suivantes. Le degré sortant est donc dans ce

cas un bon indicateur de l'influence des utilisateurs. Sur Twitter, les longues chaînes de diffusion sont plus courantes. Cela rend la mesure des centres de Jordan plus pertinente, car elle revient à rechercher la source minimisant le nombre de retweets nécessaires pour atteindre tous les utilisateurs de $\hat{U}^D$. De la même façon que sur Weibo, les performances du modèle Pinto sont faibles car ce modèle prend aussi en compte les délais de transmissions, qui sont compliqués à extraire et très bruités, comme nous l'avons vu dans le chapitre 3. Sur ces deux corpus, Lastfm et Twitter, notre approche obtient de meilleurs résultats.

D'une façon générale, les résultats obtenus vont dans le même sens que ceux du chapitre 5 : certains corpus semblent plus « faciles » que d'autres, et les performances relatives des modèles varient d'un corpus à l'autre. Toutefois, notre approche obtient systématiquement de meilleurs résultats sur les corpus réels, grâce à l'utilisation d'un espace de représentation qui la rend plus robuste au bruit et à la parcimonie des données.

7.3.2.4 Apprentissage sur des cascades partielles

Dans l'expérience précédente, nous avons considéré que nous avions accès à des épisodes de diffusion d'apprentissage *complets*, et à des épisodes de test *partiels*. En pratique, il est possible que les épisodes d'apprentissage soient eux même partiellement observés. Pour étudier ce cas, nous filtrons les épisodes d'apprentissage de la même façon que les épisodes de test, en gardant seulement 20% des utilisateurs, de façon aléatoire sur chaque épisode de d'apprentissage. Les résultats sont donnés en figure 7.5

Sur la plupart des corpus, les performances relatives des modèles sont similaires à celles obtenues dans l'expérience précédente, ce qui n'est pas surprenant puisque les ensembles de test sont les mêmes. En revanche, sur le corpus artificiel, notre modèle bat largement celui de Jordan, ce qui n'était pas le cas avant. Cela est dû au fait que le graphe appris est cette fois-ci beaucoup moins pertinent, puisque les données d'apprentissage sont partielles. Au final, dans cette expérience, notre approche est meilleure que tous les modèles de référence.

7.3.3 Complexité

Pour l'apprentissage, notre modèle et l'extraction de graphe utilisent des algorithmes itératifs prenant à peu près autant de temps à converger. Notre modèle nécessite toutefois de stocker beaucoup moins de paramètres. En revanche, de la même façon que dans le chapitre 6, inférer la source est beaucoup plus rapide avec notre modèle : cela prend en général moins d'une seconde par épisode, alors que les modèles graphiques peuvent prendre jusqu'à quelques minutes, car le calcul des plus courts chemins dans le graphe est bien plus lent que celui des distances dans l'espace de représentation. Notre modèle est

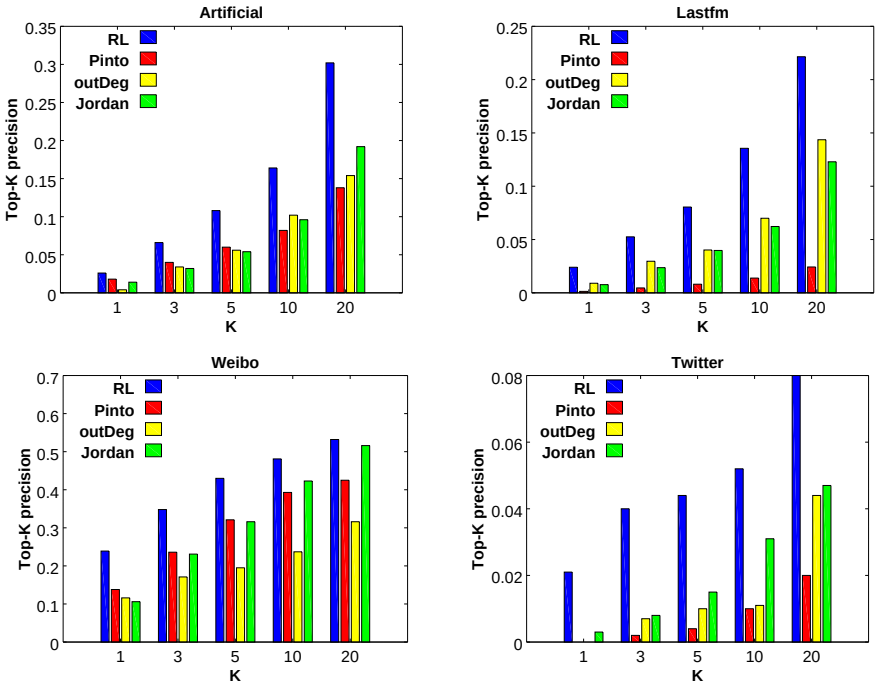


FIGURE 7.5 – Détection de source sur des épisodes de diffusion partiels (20%) avec apprentissage sur des épisodes également partiels (20%).

donc susceptible de mieux passer à l'échelle, ce qui est important lorsque l'on manipule de grands réseaux sociaux en ligne.

7.3.4 Importances des utilisateurs

Nous testons maintenant l'extension décrite en section 7.2.2.1. Nous comparons les résultats obtenus par celle-ci à ceux de la version de base, sur les corpus réels. Les résultats sont présentés en table 7.2. Nous constatons que sur le corpus Twitter, l'utilisation de poids utilisateurs améliore les résultats d'environ 10%. En effet, Twitter est un réseau social largement utilisé et particulièrement bruité. Apprendre des poids modélisant l'importance des utilisateurs permet à notre modèle de limiter l'impact des utilisateurs les plus chaotiques. Nous observons un effet similaire sur le corpus Lastfm. Sur le corpus Weibo, en revanche, les résultats restent sensiblement égaux à ceux du modèle normal, ce qui pourrait indiquer que les utilisateurs sont beaucoup plus homogènes dans ce corpus. Nous pouvons le vérifier en calculant la variance des valeurs α_i apprises sur chaque jeu de données : celle-ci est de 0.12 sur Twitter et de 0.15 sur Lastfm, contre 0.08 sur Weibo. Ces résultats pourraient permettre de sélectionner les M utilisateurs à utiliser pour obtenir la meilleure détection possible, dans le cadre d'un problème de *sélection de moniteurs* comme celui décrit dans [Seo et al., 2012].

Top-K	1	3	5	10	20
			Twitter		
RL	0.020	0.042	0.058	0.099	0.141
RL + poids	0.021	0.047	0.073	0.107	0.154
gain	3%	10%	25%	8%	9%
			Lastfm		
RL	0.052	0.12	0.166	0.2545	0.374
RL + poids	0.065	0.1335	0.175	0.2605	0.378
gain	25%	11%	5%	2%	1%
			Weibo		
RL	0.31	0.51	0.59	0.72	0.82
RL + poids	0.31	0.50	0.60	0.75	0.84
gain	0%	-2.3%	+0%	+4%	+1%

TABLE 7.2 – Détection de source avec prise en compte de l'importance des utilisateurs. Les modèles sont testés sur les épisodes de longueur 3 ou plus. En effet, sur les épisodes de longueur 2 (pour lesquels $|\hat{U}^D| = 1$), l'utilisation d'une pondération ne change pas la prédiction.

Top-K	1	3	5	10	20
RL	0.028	0.05	0.072	0.102	0.142
RL avec contenu	0.043	0.069	0.099	0.128	0.179
gain	56%	38%	38%	26%	26%

TABLE 7.3 – Intégration du contenu sur le corpus Twitter

7.3.5 Intégration du contenu

Enfin, nous testons la version avec contenu de notre modèle décrite en section 7.2.2.2. Cette version est testée sur le corpus Twitter. Nous extrayons de chaque épisode de diffusion une représentation de son contenu sous la forme d'un sac de mots des tweets qu'il contient. Le dictionnaire est filtré pour ne garder que 2000 mots. La récupération des données s'est limitée aux tweets anglophones, mais l'approche reste valide pour d'autres langues. Les résultats sont présentés en table 7.3. Nous pouvons voir que la prise en compte du contenu augmente largement nos performances, en particulier en Top-1.

7.3.5.1 Évaluation empirique

Comme nous l'avons vu dans le chapitre 6, section 6.3.3.2, les paramètres θ de la fonction de projection du contenu forment une matrice de taille $2000 \times d$, dont chaque ligne θ_i peut être vue comme une représentation du i-ème mot du dictionnaire dans $\mathbb{R}^d$. La table 7.4 présente la liste des dix mots dont les représentations ont les normes les plus élevées, c'est à dire les mots ayant le plus grand impact sur la prédiction de la source selon notre modèle. Nous pouvons voir qu'à l'exception de « new » et « retweet », il s'agit de mots

Mot	Norme
new	9.9646
obama2012	9.4358
music	9.2675
2012	8.9344
president	8.1841
iran	7.9415
nyc	7.2585
game	7.223
ohio	7.0147
retweet	6.8428

TABLE 7.4 – Liste des dix mots les plus impactants d'après notre modèle

opesr	occupyhq
leisure	getaway
music	hipster
iran	iranian
masen	mapoli

TABLE 7.5 – Paires de mots ayant les plus grandes similarités cosinus entre leurs représentations

assez informatifs, qui indiquent bien le sujet de l'information se diffusant.

De plus, dans notre modèle, deux mots ayant des représentations similaires devraient avoir le même effet sur la diffusion. Pour vérifier cette propriété, nous indiquons en table 7.5 les paires de mots ayant les plus grandes similarités, en terme de *similarité cosinus* calculée sur leurs paramètres θ_i respectifs :

$$\text{sim}(x, y) = \frac{\theta_x.\theta_y}{||\theta_x|| \times ||\theta_y||}$$

Nous pouvons voir que ces paires correspondent effectivement à des mots ayant soit des sens proches (*leisure*/*getaway* ou *iran*/*iranian*) soit à des mots utilisés dans des contextes similaires. *OpESR* (Operation Empire State Rebellion) et *OccupyHQ* font référence à des mouvements civiques américains de la mouvance « Occupy Wall Street ». *Masen* et *Mapoli* sont des abréviations de « Massachusetts Senate » et « Massachusetts Politics ».

7.4 Conclusion

Dans ce chapitre, nous avons appliqué notre méthode d'apprentissage de représentations au problème de la prédiction de source. Notre idée consistait à utiliser les représentations

des utilisateurs infectés dans un épisode de diffusion pour calculer une représentation de celui-ci, afin de trouver l'utilisateur le plus proche.

Contrairement aux modèles existants, notre approche ne repose pas sur la définition préalable d'un modèle de diffusion et n'utilise pas de graphe. Cela lui permet d'être beaucoup plus rapide à calculer en inférence.

Les résultats obtenus dans divers contextes expérimentaux ont montré la robustesse et la supériorité de notre modèle par rapport à différentes approches graphiques, qui reposent sur des hypothèses fortes et sont donc assez sensibles au bruit. Nous avons également proposé plusieurs extensions de notre modèle permettant de modéliser d'autres paramètres. Ces extensions nous ont permis d'améliorer nos résultats et ouvrent la voie à d'autres applications.

Troisième partie

Conclusion

Chapitre 8

Conclusions et perspectives

8.1 Conclusions et discussions

Au cours de ce travail de thèse, nous avons exploré divers aspects de la diffusion sur les réseaux sociaux, en revisitant certaines hypothèses couramment admises et en nous attachant à la définition de modèles robustes adaptés aux données bruitées telles que celles issues de réseaux sociaux en ligne.

8.1.1 Utilisation du temps dans la diffusion

Dans une première contribution sur laquelle nous nous sommes appuyés dans toute la suite de cette thèse, nous avons étudié une méthode d'apprentissage du modèle IC basée sur les ordres partiels d'infection plutôt que sur les temps d'infection exacts. Nous avons comparé le modèle IC ainsi appris à d'autres modèles explicatifs faisant des hypothèse plus sophistiquées sur les délais de transmission. Les résultats ont montré que notre méthode obtenait de meilleurs résultats, ce qui a conforté notre hypothèse selon laquelle les délais de transmission sont délicats à modéliser et gênent l'apprentissage. De plus notre but était uniquement de prédire les infections des utilisateurs, et non pas l'instant où ces infections avait lieu. Toute la suite du manuscrit a donc suivi ce principe, et n'a pris en compte que l'*ordre* d'infection des utilisateurs.

8.1.2 Apprentissage de représentations

L'idée centrale de cette thèse fut d'employer des techniques d'*apprentissage de représentations*, et d'étudier de ce que cela pouvait apporter pour plusieurs tâches liées à la prédiction de diffusion.

	IC	DAIC	IC Proj.	HDK
Complexité spatiale	élevée	élevée	**faible**	**faible**
Complexité inférence	élevée	élevée	élevée	**faible**
Performances	faibles	**élevées**	**élevées**	**élevées**

TABLE 8.1 – Résumé des propriétés générales des modèles de prédiction de diffusion étudiés dans ce manuscrit.

Nous avons ainsi proposé d'utiliser une méthode d'apprentissage de représentations pour définir les probabilités de transmission du modèle IC, et avons adapté l'algorithme d'apprentissage du modèle IC à cette formulation. Nous avons obtenu un modèle plus compact, avec une meilleure capacité de généralisation. Puis, en définissant la diffusion d'information comme un phénomène de diffusion de chaleur continue, nous avons proposé une approche prédictive du problème. Le modèle obtenu s'était beaucoup plus rapide qu'un modèle génératif. Nous avons également défini une extension de cette approche permettant de prendre en compte le contenu de l'information se diffusant. Enfin, nous avons appliqué cette approche au problème inverse de celui de prédiction de diffusion, la détection de source. Notre modèle prédictif s'est révélé meilleur et plus rapide que les approches existantes basées sur l'utilisation de graphes fixés.

La table 8.1 donne une rapide vue d'ensemble des différents modèles étudiés pour la prédiction de diffusion. Nous pouvons y voir que chaque modèle s'est montré meilleur que le précédent sur au moins un point.

Ces différents travaux nous ont permis d'identifier plusieurs propriétés intéressantes de l'apprentissage de représentations appliqué à la diffusion dans les réseaux sociaux.

— Les modèles définis sont plus compacts, le nombre de paramètres appris pour chaque utilisateur étant assez limité (au plus une centaine).
— Les propriétés intrinsèques des relations entre les utilisateurs sont naturellement prises en compte par l'emploi d'un espace latent. Cela permet en particulier à ces modèles d'inférer des relations utilisateurs n'existant pas dans l'ensemble d'apprentissage, comme nous avons notamment pu le voir dans le chapitre 5 (page 113). Les expériences en visualisation du chapitre 6 (page 129) nous ont également permis de voir que notre modèle identifiait des groupes d'utilisateurs aux comportements similaires.
— Les modèles basés sur l'apprentissage de représentations sont simples à étendre. Ainsi, dans les chapitres 6 et 7, nous avons pu définir des extensions permettant notamment de prendre en compte le contenu de l'information diffusée, ce qui nous a permis d'améliorer nos résultats.
— L'apprentissage de représentations nous a également permis de définir des modèles beaucoup plus rapides à utiliser en inférence que les modèles itératifs dans les chapitres 6 et 7.

8.2 Perspectives

Après avoir étudié la prédiction de diffusion, nous avons dans le dernier chapitre de ce manuscrit appliqué notre approche à la tâche de détection de source. D'autres perspectives seraient aussi susceptibles d'être étudiées.

8.2.1 Extension des modèles d'apprentissage de représentations

8.2.1.1 Ajout de connaissances supplémentaires

Les modèles définis dans les chapitres 6 et 7 peuvent être facilement étendus pour prendre en compte d'autres connaissances. Par exemple, dans le cas où certaines propriétés des utilisateurs sont connues (age, nationalité, langue), il est possible de calculer une *similarité* entre deux utilisateurs $s(u_i, u_j)$ basée sur ces caractéristiques, et d'ajouter dans la fonction de coût un terme supplémentaire de la forme :

$$\lambda \sum_{(u_i,u_j)} s(u_i, u_j)||z_i - z_j||^2$$

Ce terme favorise, avec un poids λ, les projections telles que les utilisateurs similaires soient plus proches. Cette méthode peut également servir à prendre en compte d'autres connaissances à priori : existence de certains liens particuliers, appartenance des utilisateurs à des communautés, etc...

8.2.1.2 Modélisation de plusieurs types de diffusion

Nous avons vu dans le chapitre 4 des méthodes d'apprentissage de représentations permettant de modéliser différents types de relations entre les éléments projetés. Dans nos modèles, nous en avons modélisée une seule. Il serait donc possible d'utiliser des méthodes de projections de données multi-relationnelles pour modéliser différents types de diffusion à partir d'une même représentation des utilisateurs. Par exemple, sur Twitter, les diffusions de hashtag et de retweets seraient susceptibles de répondre à des dynamiques différentes.

8.2.1.3 Communautés d'utilisateurs

Nous avons vu dans le chapitre 6 comment les utilisateurs, dans un espace de représentation à deux dimensions, semblaient former des *communautés*, c'est à dire des groupes d'utilisateurs similaires.

La détection de communautés dans les réseaux sociaux est un sujet vaste, que nous n'avons pas abordé dans ce manuscrit. Beaucoup de méthodes existantes modélisent la tâche comme un problème de partition de graphe selon un certain critère. Ces méthodes sont donc coûteuses en calcul, le problème étant souvent NP-difficile. Tout notre discours concernant les limitations des méthodes basées sur des graphes est donc valable également pour cette tâche. Utiliser l'apprentissage de représentations (en particulier l'algorithme du chapitre 6) nous permettrait d'identifier des communautés de diffusion possiblement plus robustes que celles obtenues par des méthodes classiques.

Diverses pistes peuvent être envisagées pour la détection de communautés dans le cadre de la diffusion d'information. Il serait par exemple possible de partitionner l'espace de représentation après l'apprentissage afin d'identifier des groupes d'utilisateurs. Un autre possibilité serait d'ajouter des contraintes de regroupement dans l'apprentissage des projections des utilisateurs.

Ce type de regroupement pourrait ensuite permettre d'étudier des tâches de prédiction de diffusion à différents niveaux de granularité, c'est à dire en étudiant les infections de *groupes* d'utilisateurs plutôt que celles des utilisateurs eux-mêmes. Ces infections de groupes seraient également susceptibles d'être plus régulières et moins bruitées.

8.2.1.4 Complétion de cascades

Nous avons étudié dans ce manuscrit la tâche de prédiction de diffusion et son inverse, la tâche de détection de source. Nous pouvons remarquer que ces deux tâches sont des cas particuliers d'une problématique plus générale, la « complétion de cascades » : comment retrouver, à partir d'une partie des utilisateurs infectés, la liste de tous les utilisateurs infectés avant et après ceux-ci ? Les modèles que nous avons définis peuvent s'appliquer à ce problème moyennant quelques modifications. Des expériences préliminaires nous ont donnés des résultats encourageants, en particulier pour les épisodes sur lesquels très peu d'utilisateurs sont observés.

Un modèle adapté à cette tâche permettrait de mieux comprendre les dynamiques de la diffusion d'information dans leur globalité, et d'étudier plus précisément les relations existant entre la diffusion d'information et la recommandation.

Bibliographie

[Albert and Barabási, 2002] Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74 :47–97.

[Anagnostopoulos et al., 2008] Anagnostopoulos, A., Kumar, R., and Mahdian, M. (2008). Influence and correlation in social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 7–15. ACM.

[Aral et al., 2009] Aral, S., Muchnik, L., and Sundararajan, A. (2009). Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. volume 106, pages 21544–21549. National Acad Sciences.

[Bakshy et al., 2011] Bakshy, E., Hofman, J. M., Mason, W. A., and Watts, D. J. (2011). Everyone's an influencer : quantifying influence on twitter. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 65–74. ACM.

[Barbieri et al., 2013a] Barbieri, N., Bonchi, F., and Manco, G. (2013a). Cascade-based community detection. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 33–42, New York, NY, USA. ACM.

[Barbieri et al., 2013b] Barbieri, N., Bonchi, F., and Manco, G. (2013b). Topic-aware social influence propagation models. *Knowledge and information systems*, 37(3) :555–584.

[Bass, 1969] Bass, F. M. (1969). A new product growth for model consumer durables. *Management Science*, 15 :215–227.

[Bengio et al., 2013] Bengio, Y., Courville, A. C., and Vincent, P. (2013). Representation learning : A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8) :1798–1828.

[Bengio et al., 2006] Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., and Gauvain, J.-L. (2006). Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.

[Bharathi et al., 2007] Bharathi, S., Kempe, D., and Salek, M. (2007). Competitive influence maximization in social networks. In *Internet and Network Economics*, pages 306–311. Springer.

[Bordes et al., 2013] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

[Bordes et al., 2011] Bordes, A., Weston, J., Collobert, R., and Bengio, Y. (2011). Learning structured embeddings of knowledge bases. In *Conference on Artificial Intelligence*, number EPFL-CONF-192344.

[Bóta et al., 2013] Bóta, A., Krész, M., and Pluhár, A. (2013). Approximations of the generalized cascade model. *Acta Cybern.*, 21(1) :37–51.

[Bourigault et al., 2014] Bourigault, S., Lagnier, C., Lamprier, S., Denoyer, L., and Gallinari, P. (2014). Learning social network embeddings for predicting information diffusion. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 393–402, New York, NY, USA. ACM.

[Bourigault et al., 2016a] Bourigault, S., Lamprier, S., and Gallinari, P. (2016a). Learning distributed representations of users for source detection in online social networks. In *Proceedings of the 2016 European conference on Machine Learning and Knowledge Discovery in Databases*, ECML PKDD'16. Springer-Verlag.

[Bourigault et al., 2016b] Bourigault, S., Lamprier, S., and Gallinari, P. (2016b). Representation learning for information diffusion through social networks : An embedded cascade model. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, pages 573–582, New York, NY, USA. ACM.

[Brauer et al., 2001] Brauer, F., Castillo-Chavez, C., and Castillo-Chavez, C. (2001). *Mathematical models in population biology and epidemiology*, volume 40. Springer.

[Celma, 2010] Celma, O. (2010). *Music Recommendation and Discovery*. Springer.

[Cha et al., 2010] Cha, M., Haddadi, H., Benevenuto, F., and Gummadi, K. P. (2010). Measuring User Influence in Twitter : The Million Follower Fallacy. In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media (ICWSM)*.

[Chen et al., 2013] Chen, G. H., Nikolov, S., and Shah, D. (2013). A latent source model for nonparametric time series classification. In *Advances in Neural Information Processing Systems*, pages 1088–1096.

[Chen et al., 2007] Chen, M., Yang, Q., and Tang, X. (2007). Directed graph embedding. In *Proceedings of the 2007 International Joint Conference on Artificial Intelligence*.

[Chen et al., 2012] Chen, S., Moore, J. L., Turnbull, D., and Joachims, T. (2012). Playlist prediction via metric embedding. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 714–722. ACM.

[Chen et al., 2011] Chen, W., Collins, A., Cummings, R., Ke, T., Liu, Z., Rincon, D., Sun, X., Wang, Y., Wei, W., and Yuan, Y. (2011). Influence maximization in social networks when negative opinions may emerge and propagate. In *Proceedings of the 2011 SIAM International Conference on Data Mining*. SIAM.

[Chen et al., 2010] Chen, W., Wang, C., and Wang, Y. (2010). Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038. ACM.

[Chen et al., 2009] Chen, W., Wang, Y., and Yang, S. (2009). Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM.

[Cheng et al., 2014] Cheng, J., Adamic, L., Dow, P. A., Kleinberg, J. M., and Leskovec, J. (2014). Can cascades be predicted ? In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 925–936, New York, NY, USA. ACM.

[Cho et al., 2014] Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar*.

[Daley et al., 2001] Daley, D. J., Gani, J., and Gani, J. M. (2001). *Epidemic modelling : an introduction*, volume 15. Cambridge University Press.

[Daneshmand et al., 2014] Daneshmand, H., Gomez-Rodriguez, M., Song, L., and Schoelkopf, B. (2014). Estimating diffusion network structures : Recovery conditions, sample complexity & soft-thresholding algorithm. In *Proceedings of the 31th International Conference on Machine Learning*.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

[Dong et al., 2013] Dong, W., Zhang, W., and Tan, C. W. (2013). Rooting out the rumor culprit from suspects. In *Proceedings of the 2013 IEEE International Symposium on Information Theory (ISIT)*. IEEE.

[Du et al., 2012] Du, N., Song, L., Yuan, M., and Smola, A. J. (2012). Learning networks of heterogeneous influence. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 25*, pages 2780–2788. Curran Associates, Inc.

[Farajtabar et al., 2015] Farajtabar, M., Gomez-Rodriguez, M., Zamani, M., Du, N., Zha, H., and Song, L. (2015). Back to the past : Source identification in diffusion networks from partially observed cascades. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[Freeman, 1978] Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social networks*, 1(3) :215–239.

[Fushimi et al., 2008] Fushimi, T., Kawazoe, T., Saito, K., Kimura, M., and Motoda, H. (2008). What does an information diffusion model tell about social network structure ? In *Pacific Rim Knowledge Acquisition Workshop*, pages 122–136. Springer.

[Goldenberg et al., 2001] Goldenberg, J., Libai, B., and Muller, E. (2001). Talk of the network : A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3) :211–223.

[Gomez-Rodriguez et al., 2011] Gomez-Rodriguez, M., Balduzzi, D., and Schölkopf, B. (2011). Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pages 561–568. ACM.

[Gomez Rodriguez et al., 2010] Gomez Rodriguez, M., Leskovec, J., and Krause, A. (2010). Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10. ACM.

[Gomez-Rodriguez et al., 2013] Gomez-Rodriguez, M., Leskovec, J., and Schölkopf, B. (2013). Modeling information propagation with survival theory. In *Proceedings of the 30st International Conference on Machine Learning (ICML-10)*.

[Gomez Rodriguez et al., 2013] Gomez Rodriguez, M., Leskovec, J., and Schölkopf, B. (2013). Structure and dynamics of information pathways in online media. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 23–32. ACM.

[Gomez Rodriguez et al., 2012] Gomez Rodriguez, M., Schölkopf, B., Pineau, L. J., et al. (2012). Influence maximization in continuous time diffusion networks. pages 1–8.

[Gong et al., 2014] Gong, Y., Ke, Q., Isard, M., and Lazebnik, S. (2014). A multi-view embedding space for modeling internet images, tags, and their semantics. *International journal of computer vision*, 106(2) :210–233.

[Goyal et al., 2010] Goyal, A., Bonchi, F., and Lakshmanan, L. V. (2010). Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250. ACM.

[Granovetter, 1973] Granovetter, M. S. (1973). The strength of weak ties. *American journal of sociology*, pages 1360–1380.

[Granovetter, 1978] Granovetter, M. S. (1978). Threshold Models of Collective Behavior. *American Journal of Sociology*, 83(6) :1420–1143.

[Graves et al., 2013] Graves, A., Mohamed, A., and Hinton, G. E. (2013). Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pages 6645–6649.

[Gruhl et al., 2004] Gruhl, D., Guha, R., Liben-Nowell, D., and Tomkins, A. (2004). Information diffusion through blogspace. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 491–501, New York, NY, USA. ACM.

[Guàrdia-Sebaoun et al., 2015] Guàrdia-Sebaoun, E., Guigue, V., and Gallinari, P. (2015). Latent trajectory modeling : A light and efficient way to introduce time in recommender

systems. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 281–284. ACM.

[Guille and Hacid, 2012] Guille, A. and Hacid, H. (2012). A predictive model for the temporal dynamics of information diffusion in online social networks. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion. ACM.

[Harko et al., 2014] Harko, T., Lobo, F. S., and Mak, M. (2014). Exact analytical solutions of the susceptible-infected-recovered (sir) epidemic model and of the sir model with equal death and birth rates. *Applied Mathematics and Computation*, 236 :184–194.

[Hethcote, 2000] Hethcote, H. W. (2000). The mathematics of infectious diseases. *SIAM review*, 42(4) :599–653.

[Hooton, 2015] Hooton, C. (2015). Game of thrones season 5 : This is why episodes 1 to 4 leaked. `http://www.independent.co.uk/arts-entertainment/tv/news/this-is-why-the-game-of-thrones-season-5-episodes-leaked-10175800.html`.

[Huberman et al., 2008] Huberman, B., Romero, D., and Wu, F. (2008). Social networks that matter : Twitter under the microscope. *First Monday*, 14(1).

[Jacob et al., 2014] Jacob, Y., Denoyer, L., and Gallinari, P. (2014). Learning latent representations of nodes for classifying in heterogeneous social networks. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 373–382. ACM.

[Jamali and Ester, 2010] Jamali, M. and Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142. ACM.

[Keeling and Rohani, 2008] Keeling, M. J. and Rohani, P. (2008). *Modeling infectious diseases in humans and animals*. Princeton University Press.

[Kempe et al., 2003] Kempe, D., Kleinberg, J., and Tardos, E. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 137–146. ACM.

[Kempe et al., 2005] Kempe, D., Kleinberg, J., and Tardos, É. (2005). Influential nodes in a diffusion model for social networks. In *Automata, languages and programming*, pages 1127–1138. Springer.

[Kermack and McKendrick, 1927] Kermack, W. O. and McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London A : Mathematical, Physical and Engineering Sciences*, 115(772) :700–721.

[Kimura and Saito, 2006] Kimura, M. and Saito, K. (2006). Tractable models for information diffusion in social networks. In *Proceedings of the 10th European Conference on Principle and Practice of Knowledge Discovery in Databases*, PKDD'06, pages 259–271.

[Kitsak et al., 2010] Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., and Makse, H. A. (2010). Identification of influential spreaders in complex networks. *Nature physics*, 6(11) :888–893.

[Kleinberg, 1999] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5) :604–632.

[Klimt and Yang, 2004] Klimt, B. and Yang, Y. (2004). The enron corpus : A new dataset for email classification research. In *Proceedings of the 2004 European conference on Machine Learning*, pages 217–226. Springer.

[Kondor and Lafferty, 2002] Kondor, R. I. and Lafferty, J. (2002). Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th international conference on machine learning*, pages 315–322.

[Koren et al., 2009] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, (8) :30–37.

[Kruskal, 1964] Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1) :1–27.

[Kwak et al., 2010] Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is twitter, a social network or a news media ? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600, New York, NY, USA. ACM.

[Lagnier et al., 2013] Lagnier, C., Denoyer, L., Gaussier, E., and Gallinari, P. (2013). Predicting information diffusion in social networks using content and user's profiles. In *Advances in Information Retrieval*, pages 74–85. Springer Berlin Heidelberg.

[Lamprier et al., 2015] Lamprier, S., Bourigault, S., and Gallinari, P. (2015). Extracting diffusion channels from real-world social data : A delay-agnostic learning of transmission probabilities. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. ACM.

[Lappas et al., 2010] Lappas, T., Terzi, E., Gunopulos, D., and Mannila, H. (2010). Finding effectors in social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1059–1068. ACM.

[Lerman and Hogg, 2010] Lerman, K. and Hogg, T. (2010). Using a model of social dynamics to predict popularity of news. In *Proceedings of the 19th international conference on World wide web*, pages 621–630. ACM.

[Leskovec et al., 2009] Leskovec, J., Backstrom, L., and Kleinberg, J. (2009). Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 497–506, New York, NY, USA. ACM.

[Li et al., 2013] Li, Y., Chen, W., Wang, Y., and Zhang, Z.-L. (2013). Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 657–666. ACM.

[Lin et al., 2015a] Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., and Liu, S. (2015a). Modeling relation paths for representation learning of knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

[Lin et al., 2015b] Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015b). Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the 29th Conference on Artificial Intelligence*.

[Luo et al., 2015a] Luo, W., Tay, W. P., and Leng, M. (2015a). Rumor spreading maximization and source identification in a social network. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, ASONAM '15, pages 186–193, New York, NY, USA. ACM.

[Luo et al., 2015b] Luo, W., Tay, W. P., Leng, M., and Guevara, M. (2015b). On the universality of the jordan center for estimating the rumor source in a social network. In *Digital Signal Processing (DSP), 2015 IEEE International Conference on*, pages 760–764.

[Luu et al., 2012] Luu, D. M., Lim, E.-P., Hoang, T.-A., and Chua, F. C. T. (2012). Modeling diffusion in social networks using network properties. In *Sixth International AAAI Conference on Weblogs and Social Media*.

[Ma et al., 2008] Ma, H., Yang, H., Lyu, M. R., and King, I. (2008). Mining social networks using heat diffusion processes for marketing candidates selection. In *Proceedings of the 17th ACM conference on Information and knowledge management*, CIKM '08, pages 233–242, New York, NY, USA. ACM.

[Ma et al., 2011] Ma, H., Zhou, T. C., Lyu, M. R., and King, I. (2011). Improving recommender systems by incorporating social contextual information. *ACM Transactions on Information Systems (TOIS)*, 29(2) :9.

[Mahajan et al., 1995] Mahajan, V., Muller, E., and Bass, F. M. (1995). Diffusion of new products : Empirical generalizations and managerial uses. *Marketing Science*, 14(3_supplement) :G79–G88.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv :1301.3781*.

[Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

[Mislove et al., 2007] Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., and Bhattacharjee, B. (2007). Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 29–42, New York, NY, USA. ACM.

[Morstatter et al., 2013] Morstatter, F., Pfeffer, J., Liu, H., and Carley, K. (2013). Is the sample good enough ? comparing data from twitter's streaming api with twitter's

firehose. In *Proceedgins of the 2013 International AAAI Conference on Web and Social Media*.

[Mosca, 2013] Mosca, M. (2013). « nuit debout », loi travail... et leur cortège de fausses photos. `http://www.lemonde.fr/les-decodeurs/article/2016/04/11/nuit-debout-loi-travail-et-leur-cortege-de-fausses-photos_4899993_4355770.html`.

[Myers and Leskovec, 2012] Myers, S. A. and Leskovec, J. (2012). Clash of the contagions : Cooperation and competition in information diffusion. In *Proceedings of the IEEE 12th International Conference onData Mining (ICDM)*, pages 539–548. IEEE.

[Najar et al., 2012] Najar, A., Denoyer, L., and Gallinari, P. (2012). Predicting information diffusion on social networks with partial knowledge. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 1197–1204, New York, NY, USA. ACM.

[Newman, 2003] Newman, M. E. (2003). The structure and function of complex networks. *SIAM review*, 45(2) :167–256.

[Opsahl and Panzarasa, 2009] Opsahl, T. and Panzarasa, P. (2009). Clustering in weighted networks. *Social networks*, 31(2) :155–163.

[O'Reilly, 2005] O'Reilly, T. (2005). What is web 2.0 - design patterns and business models for the next generation of software. `http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html`.

[Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking : bringing order to the web.

[Pal and Counts, 2011] Pal, A. and Counts, S. (2011). Identifying topical authorities in microblogs. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 45–54. ACM.

[Pinto et al., 2012] Pinto, P. C., Thiran, P., and Vetterli, M. (2012). Locating the source of diffusion in large-scale networks. *Physical review letters*, 109(6) :068702.

[Prakash et al., 2012] Prakash, B. A., Vreeken, J., and Faloutsos, C. (2012). Spotting culprits in epidemics : How many and which ones ? In *2012 IEEE 12th International Conference on Data Mining (ICDM)*, pages 11–20. IEEE.

[Romero et al., 2011] Romero, D. M., Galuba, W., Asur, S., and Huberman, B. A. (2011). Influence and passivity in social media. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 18–33.

[Rosenfeld et al., 2016] Rosenfeld, N., Nitzan, M., and Globerson, A. (2016). Discriminative learning of infection models. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, WSDM '16.

[Saito et al., 2009] Saito, K., Kimura, M., Ohara, K., and Motoda, H. (2009). Learning continuous-time information diffusion model for social behavioral data analysis. In *Proceedings of the 1st Asian Conference on Machine Learning : Advances in Machine Learning*, ACML '09, pages 322–337, Berlin, Heidelberg. Springer-Verlag.

[Saito et al., 2010a] Saito, K., Kimura, M., Ohara, K., and Motoda, H. (2010a). Behavioral analyses of information diffusion models by observed data of social network. In *Advances in Social Computing*, pages 149–158. Springer.

[Saito et al., 2010b] Saito, K., Kimura, M., Ohara, K., and Motoda, H. (2010b). Generative models of information diffusion with asynchronous timedelay. *Journal of Machine Learning Research - Proceedings Track*, 13 :193–208.

[Saito et al., 2010c] Saito, K., Kimura, M., Ohara, K., and Motoda, H. (2010c). Selecting information diffusion models over social networks for behavioral analysis. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 180–195. Springer.

[Saito et al., 2008] Saito, K., Nakano, R., and Kimura, M. (2008). Prediction of information diffusion probabilities for independent cascade model. In *Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems, Part III*, KES '08, pages 67–75. Springer-Verlag.

[Saito et al., 2011] Saito, K., Ohara, K., Yamagishi, Y., Kimura, M., and Motoda, H. (2011). Learning diffusion probability based on node attributes in social networks. In *Proceedings of the 19th International Conference on Foundations of Intelligent Systems*, ISMIS'11, pages 153–162, Berlin, Heidelberg. Springer-Verlag.

[Schwenk, 2007] Schwenk, H. (2007). Continuous space language models. *Computer Speech and Language*, 21(3) :492 – 518.

[Seo et al., 2012] Seo, E., Mohapatra, P., and Abdelzaher, T. (2012). Identifying rumors and their sources in social networks. In *SPIE Defense, Security, and Sensing*, pages 83891I–83891I. International Society for Optics and Photonics.

[Shah and Zaman, 2010] Shah, D. and Zaman, T. (2010). Detecting sources of computer viruses in networks : Theory and experiment. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '10, pages 203–214, New York, NY, USA. ACM.

[Shah and Zaman, 2012] Shah, D. and Zaman, T. (2012). Rumor centrality : a universal source detector. In *ACM SIGMETRICS Performance Evaluation Review*, volume 40, pages 199–210. ACM.

[Szabo and Huberman, 2010] Szabo, G. and Huberman, B. A. (2010). Predicting the popularity of online content. *Communications of the ACM*, 53(8) :80–88.

[Sénécat, 2016] Sénécat, A. (2016). « nuit debout », loi travail... et leur cortège de fausses photos. `http://www.lemonde.fr/les-decodeurs/article/2016/04/11/nuit-debout-loi-travail-et-leur-cortege-de-fausses-photos_4899993_4355770.html`.

[Tsur and Rappoport, 2012] Tsur, O. and Rappoport, A. (2012). What's in a hashtag ? : content based prediction of the spread of ideas in microblogging communities. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 643–652. ACM.

[Vapnik, 2013] Vapnik, V. (2013). *The nature of statistical learning theory*. Springer Science & Business Media.

[Vaswani and Duttachoudhury,] Vaswani, S. and Duttachoudhury, N. Learning influence diffusion probabilities under the linear threshold model.

[wa Fu et al., 2013] wa Fu, K., hong Chan, C., and Chau, M. (2013). Assessing censorship on microblogs in china : Discriminatory keyword analysis and the real-name registration policy. *Internet Computing, IEEE*, 17(3) :42–50.

[Wang et al., 2014] Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

[Weng et al., 2010] Weng, J., Lim, E.-P., Jiang, J., and He, Q. (2010). Twitterrank : Finding topic-sensitive influential twitterers. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 261–270, New York, NY, USA. ACM.

[Weston et al., 2011] Weston, J., Bengio, S., and Usunier, N. (2011). Wsabie : Scaling up to large vocabulary image annotation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Three*, IJCAI'11, pages 2764–2770. AAAI Press.

[Woo et al., 2011] Woo, J., Son, J., and Chen, H. (2011). An sir model for violent topic diffusion in social media. In *Intelligence and Security Informatics (ISI), 2011 IEEE International Conference on*, pages 15–19.

[Yang and Leskovec, 2010] Yang, J. and Leskovec, J. (2010). Modeling information diffusion in implicit networks. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 599–608, Washington, DC, USA. IEEE Computer Society.

[Yang et al., 2014] Yang, J., McAuley, J., and Leskovec, J. (2014). Detecting cohesive and 2-mode communities indirected and undirected networks. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 323–332, New York, NY, USA. ACM.

[Zhang et al., 2007] Zhang, Y.-C., Medo, M., Ren, J., Zhou, T., Li, T., and Yang, F. (2007). Recommendation model based on opinion diffusion. *EPL (Europhysics Letters)*, 80(6) :68003.

[Zhao et al., 2010] Zhao, J., Wu, J., and Xu, K. (2010). Weak ties : Subtle role of information diffusion in online social networks. *Physical Review E*, 82(1) :016105.

[Zhu and Ying, 2013] Zhu, K. and Ying, L. (2013). Information source detection in the sir model : A sample path based approach. In *Information Theory and Applications Workshop (ITA), 2013*, pages 1–9. IEEE.

Appendices

Annexe A

Preuve de la formule de mise à jour de DAIC (formule 3.7)

Démonstration.

Calculons la dérivée de $\mathcal{Q}(\mathcal{P}|\hat{\mathcal{P}})$ par rapport à un paramètre $p_{i,j}$ tel que $(u_i, u_j) \in E$. Remarquons que la double somme dans l'équation 3.6 fait que pour un $p_{i,j}$ considéré, la dérivée de Φ^D par rapport à $p_{i,j}$ ne sera non-nulle que si $D \in \mathcal{D}^?_{i,j}$. Il en résulte que :

$$\begin{aligned}
\frac{\partial}{\partial p_{i,j}}\mathcal{Q}(\mathcal{P}|\hat{\mathcal{P}}) &= \sum_{D\in\mathcal{D}} \frac{\partial}{\partial p_{i,j}}\Phi^D(\mathcal{P}|\hat{\mathcal{P}}) + \sum_{D\in\mathcal{D}} \sum_{u_y\in\bar{U}^D_\infty} \sum_{u_x\in U^D_\infty} \frac{\partial}{\partial p_{i,j}} \log(1-p_{x,y}) \\
&= \sum_{D\in\mathcal{D}} \frac{\partial}{\partial p_{i,j}}\Phi^D(\mathcal{P}|\hat{\mathcal{P}}) - |\mathcal{D}^-_{i,j}|\frac{1}{1-p_{i,j}} \\
&= \sum_{D\in\mathcal{D}^?_{i,j}} \left(\hat{P}^D_{i\to j}\frac{1}{p_{i,j}} - \frac{1}{1-p_{i,j}} + \hat{P}^D_{i\to j}\frac{1}{1-p_{i,j}} \right) - |\mathcal{D}^-_{i,j}|\frac{1}{1-p_{i,j}}
\end{aligned} \tag{A.1}$$

Pour annuler cette dérivée, nous résolvons donc :

$$\begin{aligned}
\sum_{D\in\mathcal{D}^?_{i,j}}\left(\hat{P}^D_{i\to j}\frac{1}{p_{i,j}}-\frac{1}{1-p_{i,j}}+\hat{P}^D_{i\to j}\frac{1}{1-p_{i,j}}\right)&=|\mathcal{D}^-_{i,j}|\frac{1}{1-p_{i,j}}\\
\sum_{D\in\mathcal{D}^?_{i,j}}\left(\hat{P}^D_{i\to j}\frac{1-p_{i,j}}{p_{i,j}}-1+\hat{P}^D_{i\to j}\right)&=|\mathcal{D}^-_{i,j}|\\
\sum_{D\in\mathcal{D}^?_{i,j}}\left(\hat{P}^D_{i\to j}\frac{1-p_{i,j}}{p_{i,j}}\right)-|\mathcal{D}^?_{i,j}|+\sum_{D\in\mathcal{D}^?_{i,j}}\hat{P}^D_{i\to j}&=|\mathcal{D}^-_{i,j}|\\
\frac{1-p_{i,j}}{p_{i,j}}\sum_{D\in\mathcal{D}^?_{i,j}}\hat{P}^D_{i\to j}+\sum_{D\in\mathcal{D}^?_{i,j}}\hat{P}^D_{i\to j}&=|\mathcal{D}^?_{i,j}|+|\mathcal{D}^-_{i,j}|\\
\frac{1-p_{i,j}}{p_{i,j}}&=\frac{|\mathcal{D}^?_{i,j}|+|\mathcal{D}^-_{i,j}|-\sum_{D\in\mathcal{D}^?_{i,j}}\hat{P}^D_{i\to j}}{\sum_{D\in\mathcal{D}^?_{i,j}}\hat{P}^D_{i\to j}}\\
\frac{1-p_{i,j}}{p_{i,j}}&=\frac{1-\frac{\sum_{D\in\mathcal{D}^?_{i,j}}\hat{P}^D_{i\to j}}{|\mathcal{D}^?_{i,j}|+|\mathcal{D}^-_{i,j}|}}{\frac{\sum_{D\in\mathcal{D}^?_{i,j}}\hat{P}^D_{i\to j}}{|\mathcal{D}^?_{i,j}|+|\mathcal{D}^-_{i,j}|}}\\
p_{i,j}&=\frac{\sum_{D\in\mathcal{D}^?_{i,j}}\frac{\hat{p}_{i,j}}{\hat{P}^D_j}}{|\mathcal{D}^?_{i,j}|+|\mathcal{D}^-_{i,j}|}
\end{aligned}\tag{A.2}$$

Il reste à montrer que ce point correspond bien à un maximum. Nous calculons pour cela la dérivée seconde de $\mathcal{Q}$ par rapport à $p_{i,j}$. Pour simplifier l'écriture, nous noterons $\gamma=\sum_{D\in\mathcal{D}^?_{i,j}}\frac{\hat{p}_{i,j}}{\hat{P}^D_j}=\sum_{D\in\mathcal{D}^?_{i,j}}\hat{P}^D_{i\to j}$

$$\begin{aligned}
\frac{\partial^2}{\partial p^2_{i,j}}\mathcal{Q}(\mathcal{P}|\hat{\mathcal{P}})&=\sum_{D\in\mathcal{D}^?_{i,j}}\left(\hat{P}^D_{i\to j}\frac{-1}{p^2_{i,j}}-\frac{1}{(1-p_{i,j})^2}+\hat{P}^D_{i\to j}\frac{1}{(1-p_{i,j})^2}\right)-|\mathcal{D}^-_{i,j}|\frac{1}{(1-p_{i,j})^2}\\
&=-\frac{\gamma}{p^2_{i,j}}-\frac{|\mathcal{D}^?_{i,j}|}{(1-p_{i,j})^2}+\frac{\gamma}{(1-p_{i,j})^2}-\frac{|\mathcal{D}^-_{i,j}|}{(1-p_{i,j})^2}\\
&=\frac{1}{(1-p_{i,j})^2}(\gamma-|\mathcal{D}^?_{i,j}|-|\mathcal{D}^-_{i,j}|)-\frac{1}{p^2_{i,j}}\gamma
\end{aligned}\tag{A.3}$$

En remarquant que $\gamma = \sum_{D \in \mathcal{D}^?_{i,j}} \hat{P}^D_{i \to j} < \sum_{D \in \mathcal{D}^?_{i,j}} 1 = |\mathcal{D}^?_{i,j}|$, nous pouvons déduire que le le terme $(\gamma - |\mathcal{D}^?_{i,j}| - |\mathcal{D}^-_{i,j}|)$ est négatif, et donc que la dérivée seconde de $\mathcal{Q}$ par-rapport à $p_{i,j}$ est toujours négative. Le point $p_{i,j} = \frac{\gamma}{|\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}|}$ constitue donc bien un maximum.

□

Annexe B

Preuve de l'effet du biais d'apprentissage dans DAIC (proposition 1)

Dans cette annexe, nous utilisons la notation $p_{i,j}^{(n)}$ pour désigner la valeur de $p_{i,j}$ estimée à la n-ième itération de l'algorithme d'apprentissage 1. De la même façon, $P_j^{D(n)}$ désigne la valeur de P_j^D calculée selon la formule 3.1 avec les valeurs $p_{i,j}^{(n)}$. Les formules écrites avec les notations $p_{i,j}$ et P_j^D sont implicitement valables à toutes les itérations. Cette convention nous permet d'alléger la notation, en faisant l'économie du quantificateur $\forall n$.

Commençons par démonter la proposition suivante :

Proposition 2.

$$\forall n > 0, \forall (u_i, u_j) \in E : \left(p_{i,j}^{(n)} \leq \frac{|D_{i,j}^?|}{|D_{i,j}^?| + |D_{i,j}^-|} \right)$$

Démonstration. Pour $n > 0$, cette propriété se démontre à partir de la formule de mise à jour 3.7 :

$$\forall (u_i, u_j) \in E : p_{i,j}^{(n+1)} = \frac{\sum_{D \in \mathcal{D}_{i,j}^?} \frac{p_{i,j}^{(n)}}{P_j^{D(n)}}}{|\mathcal{D}_{i,j}^?| + |\mathcal{D}_{i,j}^-|}$$

À partir de l'équation 3.1, nous pouvons remarquer que $P_j^{D(n)} \geq p_{i,j}^{(n)}$. Nous avons donc $\frac{p_{i,j}^{(n)}}{P_j^{D(n)}} \leq 1$. La proposition 2 en découle directement.

□

Nous notons $A_{i,j} = \frac{|D_{i,j}^?|}{|D_{i,j}^?|+|D_{i,j}^-|}$. La proposition 2 n'est cependant valable que pour $n > 0$. Nous admettons donc dans la suite, sans perte de généralité, que l'algorithme 1 initialise chaque paramètre $p_{i,j}^{(0)}$ à une valeur aléatoire sur l'intervalle $]0, A_{i,j}]$. Ainsi, la propriété est vérifiée pour tout n.

$$\forall n \geq 0, \forall (u_i, u_j) \in E : \left(p_{i,j}^{(n)} \leq \frac{|D_{i,j}^?|}{|D_{i,j}^?| + |D_{i,j}^-|} \right)$$

Poursuivons en montrant la proposition :

Proposition 3.

$$\forall (u_i, u_j) \in E : (|\mathcal{D}_{i,j}^-| = 0 \implies \forall n \geq 0 : (p_{i,j}^{(n+1)} \geq p_{i,j}^{(n)}))$$

Démonstration. Si $|\mathcal{D}_{i,j}^-| = 0$, et en rappelant que $p_{i,j}^{(n)}$ est toujours strictement supérieur à 0, nous avons d'après l'équation 3.7 :

$$\begin{aligned} p_{i,j}^{(n+1)} &= \frac{\sum_{D \in \mathcal{D}_{i,j}^?} \frac{p_{i,j}^{(n)}}{P_j^{D(n)}}}{|\mathcal{D}_{i,j}^?|} \\ \frac{p_{i,j}^{(n+1)}}{p_{i,j}^{(n)}} &= \frac{1}{|\mathcal{D}_{i,j}^?|} \sum_{D \in \mathcal{D}_{i,j}^?} \frac{1}{P_j^{D(n)}} \end{aligned} \tag{B.1}$$

Nous savons que $P_j^{D(n)}$ est toujours inférieur à 1. Nous pouvons en déduire que :

$$\begin{aligned} \frac{1}{P_j^{D(n)}} &\geq 1 \\ \sum_{D \in \mathcal{D}_{i,j}^?} \frac{1}{P_j^{D(n)}} &\geq |\mathcal{D}_{i,j}^?| \\ \frac{1}{|\mathcal{D}_{i,j}^?|} \sum_{D \in \mathcal{D}_{i,j}^?} \frac{1}{P_j^{D(n)}} &\geq 1 \\ \frac{p_{i,j}^{(n+1)}}{p_{i,j}^{(n)}} &\geq 1 \end{aligned} \tag{B.2}$$

□

Une fois ces proposition posées, nous pouvons commencer la démonstration proprement dite. Soient $D \in \mathcal{D}$ et $u_j \in U_\infty^D$. Notons $I_j^D = (U_{t_j}^D \cap \text{Preds}_j)$. Nous avons alors :

$$\begin{aligned} P_j^D &= 1 - \prod_{u_k \in I_j^D} (1 - p_{k,j}) \\ &= 1 - \prod_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|>0}} (1 - p_{k,j}) \prod_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|=0}} (1 - p_{k,j}) \\ &\leq 1 - \prod_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|>0}} (1 - A_{k,j}) \prod_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|=0}} (1 - p_{k,j}) \end{aligned}$$

Posons $B_j^D = \prod_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|>0}} (1 - A_{k,j})$. Remarquons que B_j^D est une constante n'évoluant pas au cours de l'apprentissage. Nous avons alors l'inégalité :

$$P_j^D \leq 1 - B_j^D \prod_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|=0}} (1 - p_{k,j}) \tag{B.3}$$

Considérons à présent qu'il existe un utilisateur $u_i \in I_j^D$ tel que $|\mathcal{D}_{i,j}^-| = 0$. Nous pouvons alors écrire :

$$P_j^D \leq 1 - B_j^D(1 - p_{i,j}) \prod_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|=0 \\ u_k \neq u_i}} (1 - p_{k,j})$$

Nous pouvons remarquer que :

$$\prod_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|=0 \\ u_k \neq u_i}} (1 - p_{k,j}) \geq \prod_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|=0 \\ u_k \neq u_i}} (1 - \max_{\substack{u_l \in I_j^D \\ |\mathcal{D}_{l,j}^-|=0 \\ u_l \neq u_i}} p_{l,j}) \tag{B.4}$$

$$\geq (1 - \max_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|=0 \\ u_k \neq u_i}} p_{k,j})^{|I_j^D \cap E^?|-1} \tag{B.5}$$

où $E^?$ désigne ici l'ensemble des couples (u_k, u_j) tels que $|\mathcal{D}_{i,j}^-| = 0$. Il en résulte l'inégalité :

$$P_j^D \leq 1 - B_j^D(1 - p_{i,j})(1 - \max_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|=0 \\ u_k \neq u_i}} p_{k,j})^{|I_j^D \cap E^?|-1} \tag{B.6}$$

La proposition 3 nous permet d'affirmer que la suite définie par :

$$v_n = (1 - \max_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|=0 \\ u_k \neq u_i}} p_{k,j}^{(n)})^{|I_j^D \cap E^?|-1}$$

est décroissante, puisque chaque valeur $p_{k,j}^{(n)}$ considérée dans le maximum est décroissante, vu que $|\mathcal{D}_{k,j}^-| = 0$. De plus, cette suite est bornée inférieurement par 0. Il en résulte que cette suite (v_n) converge vers une limite finie que nous noterons l. A partir de là, deux possibilités existent : l peut être égale à 0 ou strictement supérieure à 0.

Si $l = 0$, alors nous pouvons écrire :

$$\lim_{n \to \infty} \max_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|=0 \\ u_k \neq u_i}} p_{k,j}^{(n)} = 1$$

Or, l'équation 3.1 nous permet de savoir que $\forall u_k \in I_j^D : P_j^D \geq p_{k,j}$. La suite $P_j^{D(n)}$ est donc bornée inférieurement par une suite tendant vers 1, et bornée supérieurement par 1, ce qui nous permet de conclure, par encadrement, que $\lim_{n\to\infty} P_j^{D(n)} = 1$.

Si $l > 0$, nous avons donc l'inégalité :

$$(1 - \max_{\substack{u_k \in I_j^D \\ |\mathcal{D}_{k,j}^-|=0 \\ u_k \neq u_i}} p_{k,j})^{|I_j^D \cap E^?|-1} \geq l \tag{B.7}$$

En injectant cette inégalité dans l'inégalité B.6, nous obtenons :

$$\begin{aligned} P_j^D &\leq 1 - lB_j^D(1 - p_{i,j}) \\ &\leq 1 - \lambda(1 - p_{i,j}) \\ &\leq 1 - \lambda + \lambda p_{i,j} \end{aligned} \tag{B.8}$$

avec $\lambda = lB_j^D$. De plus, nous pouvons réécrire la formule de mise jour de l'équation B.1 en « sortant » l'épisode D de la somme :

$$p_{i,j}^{(n+1)} = \frac{\sum_{D' \in \mathcal{D}_{i,j}^? \setminus D} \frac{p_{i,j}^{(n)}}{P_j^{D'(n)}} + \frac{p_{i,j}^{(n)}}{P_j^{D(n)}}}{|\mathcal{D}_{i,j}^?|} \tag{B.9}$$

À partir de l'inégalité B.8, nous pouvons déduire que :

$$\begin{aligned} p_{i,j}^{(n+1)} &\geq \frac{\sum_{D' \in \mathcal{D}_{i,j}^? \setminus D} \frac{p_{i,j}^{(n)}}{1} + \frac{p_{i,j}^{(n)}}{1-\lambda+\lambda p_{i,j}}}{|\mathcal{D}_{i,j}^?|} \\ p_{i,j}^{(n+1)} &\geq \frac{(|\mathcal{D}_{i,j}^?| - 1)p_{i,j}^{(n)} + \frac{p_{i,j}^{(n)}}{1-\lambda+\lambda p_{i,j}}}{|\mathcal{D}_{i,j}^?|} \end{aligned} \tag{B.10}$$

Considérons maintenant la suite (w_n) définie ainsi :

$$\begin{cases} w_0 = p_{i,j}^{(0)} \\ w_{n+1} = f(w_n) \end{cases}$$

avec :

$$f(x) = \frac{(|\mathcal{D}_{i,j}^?| - 1)x + \frac{x}{1-\lambda+\lambda x}}{|\mathcal{D}_{i,j}^?|}$$

En remarquant que $(x \in]0,1[) \implies (1-\lambda+\lambda x < 1)$, il est facile de démontrer par récurrence que la suite (w_n) prend ses valeurs dans $]0,1[$. Nous pouvons alors calculer le rapport :

$$\frac{w_{n+1}}{w_n} = \frac{|\mathcal{D}^?_{i,j}| - 1 + \frac{1}{1-\lambda+\lambda w_n}}{|\mathcal{D}^?_{i,j}|} \tag{B.11}$$

$$\frac{w_{n+1}}{w_n} > 1 \tag{B.12}$$

La suite est donc strictement croissante, et admet une borne supérieure en 1. Elle est donc convergente. Étant donné que $f(1) = 1$, nous pouvons conclure que la suite (w_n) converge vers 1 (théorème du point fixe). Or, nous pouvons facilement montrer par récurrence que $\forall n : p^{(n)}_{i,j} > w_n$. Il en résulte que $\lim_{n\to\infty} p^{(n)}_{i,j} = 1$, ce qui nous permet de conclure que $\lim_{n\to\infty} P_j^{D(n)} = 1$.

À ce stade, nous avons donc démontré la proposition suivante :

Proposition 4. *Pour tout épisode de diffusion D et tout utilisateur $u_j \in U^D_\infty$, s'il existe un utilisateur $u_i \in I^D_j$ tel que $|\mathcal{D}^-_{i,j}| = 0$, alors nous avons :*

$$\lim_{n\to+\infty} P_j^{D(n)} = 1$$

Considérons maintenant la prémisse de la proposition 1. Si, pour un lien $(u_i, u_j) \in E$ tel que $|\mathcal{D}^-_{i,j}| > 0$, il existe pour chaque épisode $D \in \mathcal{D}^?_{i,j}$ un utilisateur $u_k \in I^D_j$ tel que $|\mathcal{D}^-_{k,j}| = 0$, nous pouvons déduire de la proposition 4 que :

$$\forall D \in \mathcal{D}^?_{i,j} : (\lim_{n\to+\infty} P_j^{D(n)} = 1)$$

Dès lors, nous pouvons considérer le rapport $\frac{p^{(n+1)}_{i,j}}{p^{(n)}_{i,j}}$ et calculer sa limite :

$$\frac{p^{(n+1)}_{i,j}}{p^{(n)}_{i,j}} = \frac{\sum_{D\in\mathcal{D}^?_{i,j}} \frac{1}{P_j^{D(n)}}}{|\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}|}$$

$$\lim_{n\to+\infty} \frac{p^{(n+1)}_{i,j}}{p^{(n)}_{i,j}} = \frac{|\mathcal{D}^?_{i,j}|}{|\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}|}$$

$$\lim_{n\to+\infty} \frac{p^{(n+1)}_{i,j}}{p^{(n)}_{i,j}} = A_{i,j} < 1$$

Nous pouvons en déduire qu'il existe une valeur m telle que :

$$\forall n \geq m \quad \frac{p_{i,j}^{(n+1)}}{p_{i,j}^{(n)}} < 1 \tag{B.13}$$

$$\forall n \geq m \quad p_{i,j}^{(n+1)} < p_{i,j}^{(n)} \tag{B.14}$$

Autrement dit, la suite $p_{i,j}^{(n)}$ est décroissante à partir d'un certain rang m. Cette suite étant bornée inférieurement par 0, nous pouvons conclure qu'elle converge vers une valeur finie, notée l. Enfin, nous pouvons démontrer par l'absurde que $l = 0$. En effet, si l était non-nulle, alors nous aurions $\lim_{n \to +\infty} \frac{p_{i,j}^{(n+1)}}{p_{i,j}^{(n)}} = 1$. Ce n'est pas le cas, et nous pouvons donc conclure que :

$$\lim_{n \to +\infty} p_{i,j}^{(n)} = 0$$

ce qui achève la démonstration de la proposition 1.

Annexe C

Preuve du polynome de mise à jour de DAIC régularisé (formule 3.11)

Démonstration. La démonstration est similaire est la précédente. La fonction $\mathcal{Q}(\mathcal{P}|\hat{\mathcal{P}})$ s'écrit cette fois-ci :

$$\mathcal{Q}(\mathcal{P}|\hat{\mathcal{P}}) = \sum_{D\in\mathcal{D}} \left(\Phi^D(\mathcal{P}|\hat{\mathcal{P}}) + \sum_{u_j\in\bar{U}^D_\infty} \sum_{u_i\in U^D_\infty} \log(1-p_{i,j}) \right) - \lambda \sum_{p_{i,j}\in\mathcal{P}} p_{i,j} \qquad \text{(C.1)}$$

Sa dérivation par rapport à $p_{i,j}$ s'effectue de la même manière et l'annulation de sa dérivée conduit à résoudre :

$$\begin{aligned}
\sum_{D\in\mathcal{D}^?_{i,j}} \left(\hat{P}^D_{i\to j}\frac{1}{p_{i,j}} - \frac{1}{1-p_{i,j}} + \hat{P}^D_{i\to j}\frac{1}{1-p_{i,j}}\right) &= |\mathcal{D}^-_{i,j}|\frac{1}{1-p_{i,j}} + \lambda \\
\sum_{D\in\mathcal{D}^?_{i,j}} \left(\hat{P}^D_{i\to j}\frac{1-p_{i,j}}{p_{i,j}} - 1 + \hat{P}^D_{i\to j}\right) &= |\mathcal{D}^-_{i,j}| + \lambda(1-p_{i,j}) \\
\sum_{D\in\mathcal{D}^?_{i,j}} \left(\hat{P}^D_{i\to j}\frac{1-p_{i,j}}{p_{i,j}}\right) - |\mathcal{D}^?_{i,j}| + \sum_{D\in\mathcal{D}^?_{i,j}} \hat{P}^D_{i\to j} &= |\mathcal{D}^-_{i,j}| + \lambda - \lambda p_{i,j} \\
\frac{1-p_{i,j}}{p_{i,j}} \sum_{D\in\mathcal{D}^?_{i,j}} \hat{P}^D_{i\to j} + \sum_{D\in\mathcal{D}^?_{i,j}} \hat{P}^D_{i\to j} &= |\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}| + \lambda - \lambda p_{i,j} \\
(1-p_{i,j})\gamma + p_{i,j}\gamma &= (|\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}| + \lambda)p_{i,j} - \lambda p^2_{i,j} \\
\gamma - p_{i,j}\gamma + p_{i,j}\gamma &= (|\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}| + \lambda)p_{i,j} - \lambda p^2_{i,j} \\
\lambda p^2_{i,j} - \beta p_{i,j} + \gamma &= 0
\end{aligned} \tag{C.2}$$

avec :

$$\beta = |\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}| + \lambda$$

$$\gamma = \sum_{D\in\mathcal{D}^?_{i,j}} \hat{P}^D_{i\to j} = \sum_{D\in\mathcal{D}^?_{i,j}} \frac{\hat{p}_{i,j}}{\hat{P}^D_j}$$

□

Annexe D

Preuve de la validité de la formule de mise à jour de DAIC régularisé (formule 3.14)

Le discriminant Δ du polynôme 3.11 vaut $\beta^2 - 4\lambda\gamma$, avec $\beta = |\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}| + \lambda$ et $\gamma = \sum_{D \in \mathcal{D}^?_{i,j}} \frac{\hat{p}_{i,j}}{\tilde{P}^D_j}$. Étant donné que $|\mathcal{D}^?_{i,j}| \geq \gamma$ (equation 3.13), nous avons :

$$\begin{aligned}\Delta &= (|\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}| + \lambda)^2 - 4\lambda\gamma \\ &\geq (|\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}| + \lambda)^2 - 4\lambda|\mathcal{D}^?_{i,j}| = \left(|\mathcal{D}^-_{i,j}| - |\mathcal{D}^?_{i,j}| + \lambda\right)^2 + 4|\mathcal{D}^-_{i,j}||\mathcal{D}^?_{i,j}| \\ &\geq 0\end{aligned} \tag{D.1}$$

L'équation 3.11 a donc toujours au moins une solution, dont la plus petite, notée $p'_{i,j}$ sera la formule utilisée à chaque itération de l'algorithme pour mettre à jour chaque paramètre $p_{i,j}$ en fonction des valeurs courantes de $\hat{\mathcal{P}}$:

$$p_{i,j} \leftarrow p'_{i,j} = \frac{\beta - \sqrt{\Delta}}{2\lambda} \tag{D.2}$$

Proposition 5. *La solution $p'_{i,j}$ donnée dans la formule D.2 est bien située dans l'intervalle $[0, 1]$ et peut être utilisée pour la mise à jour des paramètres à chaque étape de maximisation de la formule 3.10.*

Démonstration.

- Montrer que la valeur de $p'_{i,j}$ donnée dans l'équation D.2 est positive est simple. Il nous suffit de montrer que $\beta \geq \sqrt{\Delta}$. Les deux cotés de l'inégalité sont positifs, nous pouvons donc nous ramener à $\beta^2 \geq \Delta$, qui est toujours vrai, puisque $\Delta - \beta^2 = -4\lambda\gamma \leq 0$.
- Montrer que $p'_{i,j} \leq 1$ est équivalent à montrer que $\beta - \sqrt{\Delta} \leq 2\lambda$, ou $\beta - 2\lambda \leq \sqrt{\Delta}$.
  - Si $\lambda \geq |\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}|$, la démonstration est directe car $\beta - 2\lambda \leq 0 \leq \sqrt{\Delta}$
  - Dans le cas contraire, les deux cotés de l'inégalité sont positifs. Il est donc possible d'élever cette inégalité au carré, soit $(\beta - 2\lambda)^2 \leq \Delta$. Ceci est équivalent à $|\mathcal{D}^?_{i,j}| + |\mathcal{D}^-_{i,j}| - \gamma \geq 0$, ce qui est toujours vrai car nous savons que $|\mathcal{D}^?_{i,j}| > \gamma$.

□

Enfin, la dérivé seconde de $\mathcal{Q}$ est la même que celle sans régularisation (équation A.3) et est donc toujours négative. Le point $p'_{i,j}$ correspond donc bien à un maximum.