Apprentissage de représentation pour la détection de source dans les réseaux sociaux

Simon Bourigault — Sylvain Lamprier — Patrick Gallinari

Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France

RÉSUMÉ. *Récemment, divers travaux se sont interessés à la détection de source de diffusion dans les réseaux sociaux : il s'agit de déterminer l'utilisateur à partir duquel une information propagée a initiallement été émise. Dans cet article, nous proposons une nouvelle méthode pour la détection de source de diffusion, basée sur des techniques d'apprentissage de représentation. Plutôt que de s'appuyer sur un modèle de diffusion appris a priori pour estimer la source des diffusions observées, l'idée est de projeter les utilisateurs du réseau dans un espace de représentation, dans lequel la source de diffusion peut être efficacement extraite en fonction des positions relatives des utilisateurs infectés par l'information propagée. Cela permet d'établir un modèle de prédiction bien moins sensible au bruit et à l'incomplétude des données que les modèles existants, pour un temps de calcul bien plus faible en prédiction. Le modèle proposé a en effet démontré de bonnes performances sur divers jeux de données réels et artificiels.*

ABSTRACT. *In this paper, we study the problem of source detection in the context of information diffusion through online social networks. We propose a representation learning approach that leads to a robust model able to deal with the sparsity of the data. From learned continuous projections of the users, our approach is able to efficiently predict the source of any newly observed diffusion episode. Our model does rely neither on a known diffusion graph nor on a hypothetical probabilistic diffusion law, but directly infers the source from diffusion episodes. It is also less complex than alternative state of the art models. It showed good performances on artificial and real-world datasets, compared with various state of the art baselines.*

MOTS-CLÉS : *Détection de source, Réseaux sociaux, Diffusion d'information.*

KEYWORDS: *Source Detection, Social Networks, Information Diffusion.*

1. Introduction

Au cours des dix dernières années, les réseaux sociaux en ligne ont pris une importance capitale dans la vie personnelle et professionnelle de millions, voire de milliards, de personnes. Ces nouveaux media ont aujourd'hui un impact considérable sur la manière dont l'information se propage à travers le monde. Cela a motivé un grand nombre de travaux de recherche autour des thématiques de diffusion d'information. Dans le cadre de tâches de prédiction de la diffusion, l'objectif est de déterminer quels utlisateurs (ou combien) seront infectés par une information donnée lorsque celle-ci est initialement émise à un point connu du réseau (Guille *et al.*, 2013).

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. Ce phénomène a motivé un certain 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).

Bien que divers travaux se sont déjà penchés sur le sujet, toutes les approches proposées sont basées sur des modèles graphiques de diffusion appris a priori, dont elles se servent pour prédire la source la plus vraissemblable étant donné un ensemble d'infectés observés (Pinto *et al.*, 2012 ; Shah et Zaman, 2010 ; Farajtabar *et al.*, 2015). Cela implique d'avoir à disposition un modèle de diffusion bien représentatif des dynamiques de propagation en jeu dans le réseau, ce qui peut être difficile dans bien des cas tant les données issues des réseaux sociaux peuvent être bruitées ou incomplètes. En outre, ce genre d'approche paraît être peu robuste aux évolutions du réseau, la suppression d'un noeud dans le modèle de diffusion pouvant modifier fortement sa structure en termes de chemins de propagation possibles. Enfin, ce type d'approche implique de lourds traitements pour la détection de la source de chaque nouvel épisode de diffusion.

Dans cet article, nous proposons de dépasser ce genre de limites par la définition d'un modèle spécifiquement dédié à la détection de source, basé sur des techniques d'apprentissage de représentation. Il s'agit de projeter les utilisateurs du réseau dans un espace de représentation dans lequel leurs positions respectives permet une détection efficace de la source étant donné un ensemble d'infectés observés. Cette méthode ne requiert pas de disposer d'un graphe de relations sociales du réseau, et peut être appliquée dans le cas où seuls les états d'un sous ensemble de la population ont pu être observés. De plus, elle permet de facilement considérer divers facteurs additionnels, tels que la nature du contenu propagé.

La suite de cet article est organisée de la manière suivante. La section 2 donne un aperçu des approches existantes dans la litterature. La section 3 introduit le modèle. Enfin, la section 4 compare notre modèle à diverses approches de référence sur des jeux de données réels et artificiels.

2. Motivations et travaux connexes

L'article fondateur de la détection de source dans le cadre de la diffusion d'information dans les réseaux sociaux date de 2010 (Shah et Zaman, 2010). Dans cet article, qui considère que le graphe de diffusion $G = (U, E)$ est connu et non-orienté, les auteurs se basent sur un modèle de diffusion similaire au populaire modèle NetRate (Gomez-Rodriguez *et al.*, 2011), dans lequel l'information transite de proche en proche dans le réseau, avec des délais de transmission entre voisins déterminés de manière indépendante, selon une loi exponentielle de paramètre fixé pour l'ensemble du réseau. Ignorant les temps d'infection de chaque utilisateur infecté par une diffusion donnée sur une période de durée T, l'objectif est de déterminer parmi les utilisateurs infectés observés U_T lequel est l'utilisateur source de l'information diffusée. Les auteurs définissent un estimateur de type maximum de vraisemblance :

$$\hat{u}_s = \arg\max_{u_s \in U_T} P(U_T|u_s) \quad [1]$$

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 considéré. 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. Pour le cas où G est un arbre, il est montré que l'estimation de la source peut s'écrire :

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

avec $RC(U_T, u_s)$ une mesure de « centralité de rumeur » correspondant au nombre de séquences d'infection des utilisateurs de U_T il est possible d'observer si l'information a initialement été émise par u_s. Dans le cas général où G est un graphe quelconque, les auteurs proposent d'en extraire un arbre $\mathcal{T}(u_s, G)$ par exploration en largeur d'abord de G en partant de u_s et limitée à U_T, avant d'appliquer la mesure de centralité de rumeur pour quantifier la propension de u_s a avoir été la source de la diffusion. Au lieu de considérer tous les chemins possibles pour estimer $P(U_T|u_s)$, cela revient à ne considérer que les plus courts chemins dans le graphe. Les auteurs ont notamment montré que ce genre d'approche permet d'obtenir de meilleurs résultats, en terme de distance à la vraie source, qu'une mesure de centralité de distance (qui considère la source la plus proche de l'ensemble des infectés en terme de distance dans le graphe). Ce travail a ensuite été poursuivi dans (Shah et Zaman, 2012), où des résultats théoriques sont donnés pour d'autres types de graphes.

En parallèle, les auteurs de (Luo *et al.*, 2015b) proposent de considérer le centre de Jordan du graphe :

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

où $Dist(u_s, u_i)$ est la longueur du plus court chemin entre u_s et u_i. L'utilisation du centre de Jordan revient à sélectionner la source minimisant le nombre de pas nécessaires dans le graphe pour contaminer tous 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. Enfin, le problème a également été abordé dans (Dong *et al.*, 2013) où l'on considère 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.

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). Dans ce cadre, 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 considèrent alors un estimateur par maximum de vraisemblance similaire à celui de la formule 1 mais défini seulement sur les infectés observés. L'estimation est très complexe car il s'agit d'estimer la probabilité des infectés selon tous les chemins possibles, tout en énumérant les états possibles pour les utilisateurs non-observés. Le calcul exact ne passant pas l'échelle, les auteurs adoptent une méthodologie similaire à celle de (Shah et Zaman, 2010) : 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. À noter que plus récemment, (Farajtabar *et al.*, 2015) ont proposé une approche similaire mais considérant les délais d'infection successifs dans le calcul de vraissemblance et proposant une méthode d'échantillonnage préférentiel pour en réduire la complexité.

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 et 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. Or, il se peut que l'on ne soit pas en mesure d'observer tout réseau et que les informations d'infection que nous avons soient incomplètes. En outre, il est courant que l'on ne dispose pas du graphe de relations du réseau considéré, ou que ces relations ne soient que faiblement représentatives des réels canaux de diffusion du réseau (Ver Steeg et Galstyan, 2013).

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.

Afin de dépasser ces limitations, nous proposons de considérer l'emploi de techniques d'apprentissage de représentation pour le problème de la détection de source. Cela fait suite au travail présenté dans (Bourigault *et al.*, 2014), qui appliquait ce genre de techniques pour des tâches de prédiction de diffusion. Récemment, l'apprentissage de représentation a été employé dans de nombreux domaines, tels que la prédiction de listes d'écoute (Chen *et al.*, 2012) ou la modélisation de la langue (Mikolov *et al.*, 2013). Ces méthodes ont pour principe de projecter les entités relationnelles considérées, telles que des chansons, des utilisateurs ou des mots dans un espace de représentation latent, dans lequel les positions relatives des entités sont déterminées en fonction des relations existant entre elles. L'apprentissage de relations a au moins les avantages suivants dans le contexte de la détection de source :

– Les capacités de compression offertes par ces techniques permettent la définition de modèles plus compacts ;

– Les relations d'influence, qui sont encodées dans un espace de représentation commun, sont naturellement régularisées : les utilisateurs aux comportements similaires sont susceptibles d'être projetés dans les mêmes zones de l'espace, et ont donc tendance à impacter les mêmes autres utilisateurs. Cela permet l'obtention de modèles généralisant mieux, d'autant plus dans le cadre de problèmes d'apprentissage avec des données aussi parcimonieuses que celui de la modélisatoin de la diffusion ;

– Une représentation d'un épisode de diffusion peut être efficacement déduite des représentations individuelles de utilisateurs infectés observés. Cela permet des procédures de détection de source bien plus rapides que de passer par des cacluls de vraissemblance dans les graphes ;

– Le contenu diffusé, ou toute autre information additionnelle, peut facilement être considéré, en définissant des transformations spécifiques de l'espace de représentation.

Plutôt que de renverser un modèle de diffusion donné, tel que c'est fait classiquement dans les approches de la littérature, nous proposons d'apprendre des projections des utilisateurs permettant une détection efficace de la source de diffusion à partir des infectés observés.

3. Modèle

Soit $\mathcal{U} = \{u_1, \ldots, u_N\}$ un ensemble de N utilisateurs qui communiquent de l'information. Lorsqu'un contenu se *propage* dans cette population, nous observons un *épisode de diffusion*, qui correspond à une séquence d'utilisateurs infectés associés à leur temps d'infection :

$$D = \{(u_i, t_i), (u_j, t_j)...\}$$

Un épidose de diffusion peut correspondre, par exemple, à une séquence d'utilisateurs qui ont signalé une vidéo ou reposté un message sur le media social considéré. Les permier utilisateur de la séquence est l'utilisateur *source*, dénoté par s_D. Dans la suite, on note $\mathcal{U}_D$ l'ensemble des utilisateurs infectés par l'épisode D et $\hat{\mathcal{U}}_D$ le même ensemble mais sans l'utilisateur source de D (i.e., $\hat{\mathcal{U}}_D = \mathcal{U}_D \setminus \{s_D\}$). Notre objectif est de retrouver la source s_D étant donné un épisode de diffusion D dans lequel la source est manquante (i.e. on observe $\hat{\mathcal{U}}_D$).

3.1. *Modèle de base*

L'idée du modèle proposé est de projeter les utilisateurs du réseau dans un espace de représentation, de manière à ce qu'il soit possible de prédire la source de chaque épisode de diffusion D en considérant simplement les positions des utilisateurs observés infectés $\hat{\mathcal{U}}_D$ dans cet espace. Chaque utilisateur $u_i \in U$ est en fait associé à deux représentations z_i et ω_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 $\hat{U}^D$.

Selon ce principe, la représentation de l'épisode z_D correspond à un point initial de la diffusion, à partir duquel on a commencé à émettre pour atteindre tous les utilisateurs infectés de l'épisode D. Dans ce contexte, la construction du modèle de diffusion revient à faire coïncider la représentation de l'utilisateur source s_D avec ce point de diffusion initiale z_D. Plusieurs définitions de $\phi : 2^U \rightarrow \mathbb{R}^d$ sont possibles[1]. Nous choisissons d'utiliser une moyenne, qui a l'avantage d'être rapide à calculer :

$$z_D = \phi(\hat{U}^D) = \frac{1}{|\hat{U}^D|} \sum_{u_i \in \hat{U}^D} \omega_i \qquad [2]$$

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

1. Rappelons que 2^U désigne l'ensemble des parties de U

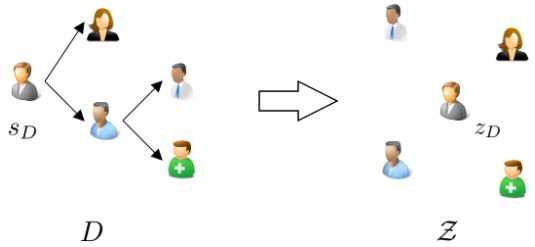

Figure 1 – Les utilisateurs de l'épisode de diffusion D sont projetés de façon à ce que la source se trouve au centre des représentations des utilisateurs infectés.

.

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 1, où la source de l'épisode D est projetée près du centre des utilisateurs $\hat{U}^D$.

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 :

$$\hat{s} = \underset{u_i \in U \setminus \hat{U}^D}{\arg\min} ||z_i - z_D||^2 \qquad [3]$$

où z_D est calculée selon la formule 2 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 3 soit valide, nous minimisons la fonction de coût suivante :

$$\mathcal{L}(\mathcal{Z}, \Omega; \mathcal{D}) = \sum_{D \in \mathcal{D}} \sum_{u_i \notin U^D} h\left(||z_i - z_D||^2 - ||z_{s_D} - z_D||^2\right) \qquad [4]$$

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 formule 3.

Ce coût peut être minimisé en utilisant une descente de gradient stochastique. Celle-ci est détaillée dans l'algorithme 1. À chaque itération, un épisode D et un utilisateur « non-source » u_j ne faisant pas partie de U^D_∞ (ligne 6) sont échantillonnés. 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 9), toutes les projections concernées (i.e les représentations récepteurs des utilisateurs de $\hat{U}^D$, ainsi que z_j et z_{s_D}) sont mises à jour selon un pas de gradient (lignes 10 à 13). 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 18). À noter que le tirage aléatoire réalisé en ligne 6 introduit un biais dans l'appren-

Algorithme 1 : Apprentissage de représentations pour la détection de source

Entrées :
U : Ensemble d'utilisateurs ; $\mathcal{D}$: Ensemble d'épisodes ; d : Nombre de dimensions ; ϵ : Pas de gradient ; F : Fréquence des tests de convergences ;
Sorties :
$Z = \{\forall u_i \in U : z_i \in \mathbb{R}^d\}$; $\quad \Omega = \{\forall u_i \in U : \omega_i \in \mathbb{R}^d\}$;

1 **pour** $u_i \in U$ **faire**
2 Initialiser z_i et ω_i aléatoirement, de façon uniforme sur $[-1, 1]^d$
3 **fin**
4 $it \leftarrow 0$; $oldL \leftarrow 0$;
5 **tant que** *true* **faire**
6 Tirer un épisode $D \in \mathcal{D}$ et $u_j \notin U^D$;
7 Calculer z_D suivant la formule 2 ;
8 $d_s \leftarrow ||z_{s_D} - z_D||^2$; $d_j \leftarrow ||z_j - z_D||^2$; $\beta \leftarrow \frac{|\mathcal{D}| \times (N - |U^D|)}{\sum_{D \in \mathcal{D}} (|U| - |U^D|)}$;
9 **si** $d_j - d_s < 1$ **alors**
10 $z_{s_D} \leftarrow z_{s_D} - \epsilon \times \beta \times 2\,(z_{s_D} - z_D)$;
11 $z_j \leftarrow z_j + \epsilon \times \beta \times 2\,(z_j - z_D)$;
12 **pour** $u_x \in \hat{U}^D$ **faire**
13 $\omega_x \leftarrow \omega_x - \epsilon \times \beta \times \frac{2}{|\hat{U}^D|}\,(z_j - z_{s_D})$
14 **fin**
15 **fin**
16 **si** $it \mod F = 0$ **alors**
17 $L \leftarrow \mathcal{L}(\mathcal{Z}, z)$
18 **si** $L \geq oldL$ **alors**
19 **retourner** $(\mathcal{Z}, \Omega)$;
20 **fin**
21 $oldL \leftarrow L$;
22 **fin**
23 $it \leftarrow it + 1$
24 **fin**

tissage, en donnant trop de poids aux utilisateurs non-infectés dans les épisodes longs. Le poids β calculé à la ligne 8 permet alors de corriger ce biais.

3.2. *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.*, 2013). Pour prendre en compte cette propriété, nous ajoutons un terme de régularisation au coût :

$$\mathcal{L}_\lambda(\mathcal{Z}, \Omega; \mathcal{D}) = \sum_{D \in \mathcal{D}} \sum_{u_i \notin U^D} h\left(||z_i - z_D||^2 - ||z_{s_D} - z_D||^2\right) + \lambda \sum_{u_i} ||z_i - \omega_i||^2 \quad [5]$$

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.

3.3. *Extensions*

3.3.1. *Modélisation de l'importance des utilisateurs*

Nous présentons maintenant une première extension possible de notre modèle, consistant à associer un *poids* α_i à chaque utilisateur en redéfinir z_D de la manière suivante :

$$z_D = \sum_{u_i \in \hat{U}^D} \frac{e^{S.\alpha_i}}{\sum\limits_{(u_j \in \hat{U}^D)} e^{S.\alpha_j}} \omega_i \quad [6]$$

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).

3.3.2. *Intégration du contenu*

Le contenu d'une information pouvant avoir un impact important sur sa diffusion, nous proposons une extension de notre modèle permettant de prendre en compte le contenu d'une information pour la détection de source. Pour cela, bien que diverses

	$\|U\|$	$\|E\|$	$\|\mathcal{D}\|$	Densité
Artificiel	100	262	10000	2%
Lastfm	1984	235011	331829	5%
Weibo	5000	20784	44345	0.08%
Twitter	4107	128855	16824	1%

Tableau 1 – Quelques statistiques sur les corpus : nombre d'utilisateurs $|U|$, de liens dans le graphe $|E|$, d'épisodes de diffusion, et densité du graphe pour la définition du graphe utilisé).

autres transformations auraient pu être envisagées, nous proposons une simple translation de la représentation z_D en fonction du contenu de l'information considérée. Le contenu d'un épisode D est alors représenté par un vecteur-ligne $w_D \in \mathbb{R}^Q$, et Pour cela, nous apprenons les paramètres $\theta \in \mathbb{R}^{Q \times d}$ d'une fonction linéaire f_θ permettant de projeter le contenu $w_D \in \mathbb{R}^Q$ d'un épisode D dans $\mathbb{R}^d$ (avec Q la taille du vocabulaire) : $f_\theta(w_D) = w_D.\theta$. La représentation de D est alors calculée comme :

$$z_D = \left(\frac{1}{|\hat{U}^D|} \sum_{u_i \in \hat{U}^D} \omega_i \right) + f_\theta(w_D) \qquad [7]$$

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.

4. Expériences

Les jeux de données suivants ont été utilisés pour nos expériences :

Artificial Episodes de diffusion générés selon le modèle IC (Saito *et al.*, 2008) sur un réseau invariant d'échelle de 100 utilisateurs ;

Lastfm Jeu de données extrait d'un site de diffusion de musique en continu. Chaque épisode regroupe les utilisateurs ayant écouté une chanson donnée ;

Weibo Cascades de retweet extraites du site de microbloging en utilisant la procédure décrite dans (Leskovec *et al.*, 2009). Le jeu de données a été collecté par (wa Fu *et al.*, 2013) ;

Twitter Episodes de hashtags sur Twitter, pour une population d'environ 5000 utilisateurs pendant la campagne présidentielle américaine de 2012.

Chaque corpus a été filtré pour ne conserver que ses 5000 utilisateurs les plus actifs. La table en donne quelques statistiques.

Nous comparons notre approche à plusieurs heuristiques ou modèles issus de la littérature :

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.

Jordan Center : 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.

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 maximal.

Pour toutes ces approches, le graphe de diffusoin est obtenu en utilisant l'algorithme décrit dans (Lamprier *et al.*, 2015) pour apprendre es paramètres de modèle Independent Cascade (IC) selon l'ensemble d'apprentissage $\mathcal{D}$. Nous conservons dans le graphe les liens (u_i, u_j) ainsi appris 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 1.

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.

4.1. *Choix du nombre de dimensions*

En premier lieu, nous étudions l'impact du nombre de dimensions de l'espace de représetation utilisé sur le temps d'apprentissage avant convergence et les performances du modèle. La figure 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.

À noter que 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, inférer la source est beaucoup plus rapide avec notre modèle : cela prend en général moins d'une seconde par épisode, alors que les modèles basés sur le graphe 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 donc

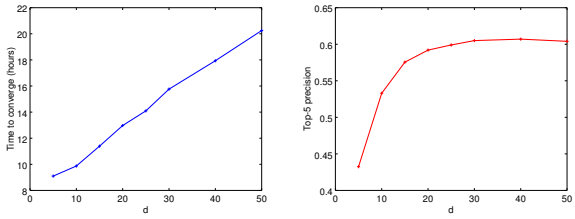

Figure 2 – Durée de l'apprentissage et performances obtenues (en Top-5) sur le corpus Weibo.

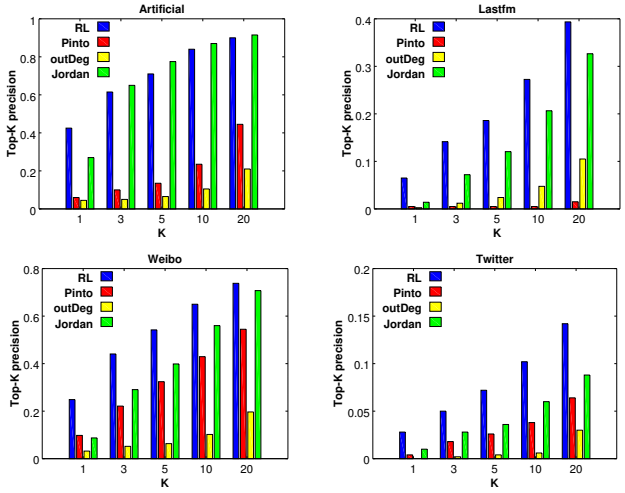

Figure 3 – Détection de source sur des épisodes de diffusion complets.

susceptible de mieux passer à l'échelle, ce qui est important lorsque l'on manipule de grands réseaux sociaux en ligne.

4.2. *Détection de source*

L'objectif 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 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 étant générés selon un modèle IC, 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 a priori. Le modèle de Pinto, par contre, base sa prédiction 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 a des difficultés pour retrouver un graphe de diffusion bien représentatif des dynamiques de diffusion en jeu. 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).

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 en outre 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.

Enfin, dans certaines applications réelles, il est possible que les épisodes de diffusion ne soient que partiellement observés. Pour étudier l'impact de ce phénomène sur les performances, nous proposons un jeu d'expériences additionnelles, pour lesquelles nous retirons au hasard des utilisateurs de $\hat{U}^D$ avant de réaliser l'apprentissage des modèles, en ne gardant que 20% de ceux-ci. Les résultats se trouvent en figure 4. Sur l'ensemble des corpus réels, les performances relatives des modèles sont similaires à celles obtenues dans l'expérience précédente. En revanche, on peut noter que sur le corpus artificiel, notre modèle obtient de biens meilleurs résultats que le modèle de Jordan, ce qui n'était pas le cas avec les données complètes où les graphes de diffusion appris pouvaient être de très bonne qualité. Avec des données manquantes, le graphe appris par IC est cette fois-ci beaucoup moins pertinent, le retrait de certains infectés des episodes d'entraînement dégrade la capacité à capturer les chemins de diffusion les plus vraissemblables. Cela dénote d'une meilleure robustesse de notre approche au retrait d'observations pour l'apprentissage des modèles.

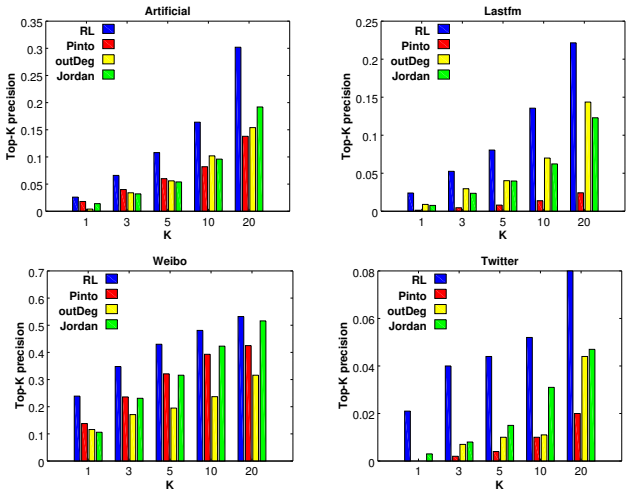

Figure 4 – Détection de source sur des épisodes de diffusion partiels (20%) avec apprentissage sur des épisodes également partiels (20%).

Twitter

Top-K	1	3	5	10	20
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

Top-K	1	3	5	10	20
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

Top-K	1	3	5	10	20
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%

Tableau 2 – Détection de source avec prise en compte de l'importance des utilisateurs.

4.3. *Importance des utilisateurs*

Nous testons maintenant l'extension décrite en section 3.3.1. Nous comparons les résultats obtenus par celle-ci à ceux de la version de base, sur les corpus Twitter et LastFM. Les résultats sont présentés en table 2. Nous constatons que sur le corpus Twitter, l'utilisation de poids utilisateur 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

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%

Tableau 3 – Intégration du contenu sur le corpus Twitter

0.15 sur Lastfm, contre 0.08 sur Weibo. Ces résultats pourraient en outre 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).

4.4. *Intégration du contenu*

Enfin, nous testons la version avec contenu de notre modèle décrite en section 3.3.2. 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. Les résultats sont présentés en table 3. Nous pouvons voir que la prise en compte du contenu augmente largement nos performances, en particulier en Top-1.

5. Conclusion

Dans ce papier, nous avons proposé une nouvelle méthode de détection de source dans les épisodes de diffusion, basée sur des techniques d'apprentissage de représentation. Cette méthode considère un espace de représentation dans lequel sont projetés les utilisateurs du réseau pour rendre compte des tendances de diffusion en place et produire une manière efficace d'extraire l'utilisateur source à partir des infections observées. 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 deux extensions pour la sélection des utilisateurs les plus importants et l'intégration du contenu qui ont permis d'observer une amélioration des résultats. Les travaux futurs concernent la définition de nouvelles manières de construire la représentation synthétique des épisodes et la définition de modes de régularisation permettant d'accroître encore la robustesse de l'approche à l'incomplétude des données observées. Des approches de complétion de cascade sont également envisagées en s'appuyant sur les travaux présentés ici.

6. Bibliographie

Barbieri N., Bonchi F., Manco G., « Cascade-based Community Detection », WSDM '13, ACM, New York, NY, USA, p. 33-42, 2013.

Bourigault S., Lagnier C., Lamprier S., Denoyer L., Gallinari P., « Learning Social Network Embeddings for Predicting Information Diffusion », WSDM'14, ACM, New York, NY, USA, p. 393-402, 2014.

Chen S., Moore J. L., Turnbull D., Joachims T., « Playlist prediction via metric embedding », SIGKDD'12, ACM, p. 714-722, 2012.

Dong W., Zhang W., Tan C. W., « Rooting out the rumor culprit from suspects », ISIT'13, 2013.

Farajtabar M., Gomez-Rodriguez M., Zamani M., Du N., Zha H., Song L., « Back to the Past : Source Identification in Diffusion Networks from Partially Observed Cascades. », AISTATS'15, 2015.

Gomez-Rodriguez M., Balduzzi D., Schölkopf B., « Uncovering the Temporal Dynamics of Diffusion Networks », ICML '11, ACM, p. 561-568, 2011.

Guille A., Hacid H., Favre C., Zighed D. A., « Information Diffusion in Online Social Networks : A Survey », *SIGMOD Rec.*, vol. 42, n° 2, p. 17-28, July, 2013.

Lamprier S., Bourigault S., Gallinari P., « Extracting Diffusion Channels from Real Social Data : A Delay-Agnostic Learning of Transmission Probabilities », ASONAM'15, ACM, 2015.

Lappas T., Terzi E., Gunopulos D., Mannila H., « Finding effectors in social networks », SIGKDD'10, ACM, p. 1059-1068, 2010.

Leskovec J., Backstrom L., Kleinberg J., « Meme-tracking and the dynamics of the news cycle », KDD '09, ACM, p. 497-506, 2009.

Luo W., Tay W. P., Leng M., « Rumor Spreading Maximization and Source Identification in a Social Network », ASONAM'15, ACM, New York, NY, USA, p. 186-193, 2015a.

Luo W., Tay W. P., Leng M., Guevara M., « On the universality of the Jordan center for estimating the rumor source in a social network », DSP'15, p. 760-764, 2015b.

Mikolov T., Chen K., Corrado G., Dean J., « Efficient estimation of word representations in vector space », *arXiv :1301.3781*, 2013.

Pinto P. C., Thiran P., Vetterli M., « Locating the source of diffusion in large-scale networks », *Physical review letters*, vol. 109, n° 6, p. 068702, 2012.

Saito K., Nakano R., Kimura M., « Prediction of Information Diffusion Probabilities for Independent Cascade Model », KES '08, Springer-Verlag, p. 67-75, 2008.

Seo E., Mohapatra P., Abdelzaher T., « Identifying rumors and their sources in social networks », SPIE'12, p. 83891I-83891I, 2012.

Shah D., Zaman T., « Detecting Sources of Computer Viruses in Networks : Theory and Experiment », SIGMETRICS'10, ACM, p. 203-214, 2010.

Shah D., Zaman T., « Rumor centrality : a universal source detector », SIGMETRICS'12, ACM,vol. 40, ACM, p. 199-210, 2012.

Ver Steeg G., Galstyan A., « Information-theoretic measures of influence based on content dynamics », WSDM '13, ACM, New York, NY, USA, p. 3-12, 2013.

Fu K., Chan C., Chau M., « Assessing Censorship on Microblogs in China : Discriminatory Keyword Analysis and the Real-Name Registration Policy », *Internet Computing, IEEE*, vol. 17, n° 3, p. 42-50, May, 2013.