Apprentissage de Représentations Probabilistes pour la Prédiction de Diffusions d'Informations sur les Réseaux Sociaux[1]

Simon Bourigault, Sylvain Lamprier, Patrick Gallinari

Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France. CNRS, UMR 7606, LIP6, F-75005, Paris, France.
prénom.nom@lip6.fr

RÉSUMÉ. *Le modèle* Independent Cascades *(IC) est un modèle central pour la capture des dynamiques de diffusion d'information sur les réseaux sociaux. Nous nous intéressons ici à l'apprentissage des probabilités de transmission utilisées par ce modèle. Plutôt que de directement travailler sur le graphe du réseau social considéré, ce qui implique un coût important dû au nombre de paramètres à apprendre pour les réseaux denses, nous proposons une approche basée sur des techniques d'apprentissage de représentations, afin d'alléger le processus et de gagner en généralisabilité. L'idée est de chercher une projection des utilisateurs du réseau dans un espace vectoriel, de manière à ce que les distances entre les individus représentent leurs probabilités de transmission d'information. Les expérimentations menées démontrent la pertinence de l'approche pour la modélisation des dynamiques de diffusion.*

ABSTRACT. *Based on the well-known* Independent Cascade *(IC) model, we embed users of the social network in a representation space to extract more robust diffusion probabilities than those defined by classical graphical learning approaches for social influence modeling. Better generalization abilities provided by the use of such a projection space allows our approach to present good performances on various real-world datasets, for both diffusion prediction and influence relationships inference tasks.*

MOTS-CLÉS : *Apprentissage de Représentation, Diffusion d'Information.*

KEYWORDS: *Representation Learning, Information Diffusion.*

[1] Traduction d'un article publié en anglais à WSDM 2016 (Bourigault *et al.*, 2016)

1. Introduction

Au cours des quinze dernières années, les réseaux sociaux sont devenus un média d'information incontournable, mettant en jeu des dynamiques complexes de communication entre utilisateurs. La modélisation de la diffusion d'information sur les réseaux constitue depuis lors un enjeu majeur, pour diverses tâches telles que l'identification de leaders d'opinions, la prédiction ou la maximisation de l'impact d'un contenu diffusé, la détection de communautés d'opinions, ou plus généralement l'analyse des dynamiques du réseau considéré.

Nous proposons dans cet article une approche basée sur l'apprentissage de représentations. Plutôt que d'apprendre des probabilités de transmission pour tous les liens du réseau considéré, tel que c'est le cas dans les approches classiques basées sur le très populaire modèle *Independent Cascades* (IC) (Gruhl *et al.*, 2004), nous modélisons la diffusion au sein d'un espace de représentation vectoriel. Cela nous permet à la fois de limiter la complexité du modèle et de gagner en capacité de généralisation. De la même manière qu'IC, nous considérons un modèle itératif, dans lequel le contenu diffusé transite de noeuds en noeuds du réseau, mais où les probabilités de transmission sont issues des positions relatives des utilisateurs dans l'espace de projection. En outre, nous considérons que le graphe de diffusion est inconnu, et que l'information peut a priori se transmettre de n'importe quel utilisateur à n'importe quel autre du réseau. Nous modifions l'algorithme d'apprentissage proposé dans (Saito *et al.*, 2008) pour l'adapter à cette formulation.

La suite de cet article est organisée comme suit : la section 2 présente les motivations de l'approche proposée en la positionnant par rapport à d'autre travaux existants. La section 3 décrit notre approche. La section 4 présente les résultats obtenus sur différents jeux de données réels.

2. Motivations et Travaux Connexes

Bien que l'étude du phénomène de diffusion soit initialement issue des domaines de l'épidémiologie et de la sociologie, le développement des réseaux sociaux en ligne a permis à toute une littérature d'émerger dans le domaine de l'informatique. La plupart des travaux existants sont basés sur deux modèles fondamentaux : l'*Independent Cascades Model* (IC) (Kempe *et al.*, 2003) et le *Linear Threshold Model* (LT) (Goldenberg *et al.*, 2001). Ces deux approches modélisent la diffusion à l'échelle des utilisateurs : alors que le modèle IC fait une hypothèse d'indépendance des évènements d'infection[1] et considère un processus de diffusion comme une cascade s'écoulant le long des liens du réseau, le modèle LT s'appuie sur une hypothèse d'additivité de l'influence

[1]Les termes "infection" ou "contamination" sont utilisés pour désigner le fait qu'un individu a été atteint par le contenu diffusé.

et décide de l'infection d'un utilisateur en fonction de la pression sociale à laquelle il est exposé. Dans cet article, nous utilisons un formalisme proche de celui du modèle IC, qui a servi de base à de très nombreux travaux. Certains s'intéressent notamment aux délais de diffusion de l'information (Saito *et al.*, 2009 ; Gomez-Rodriguez *et al.*, 2011), d'autres cherchent à prendre en compte le contenu diffusé afin d'affiner la modélisation des dynamiques de diffusion (Saito *et al.*, 2011 ; Guille et Hacid, 2012 ; Lagnier *et al.*, 2013).

Les modèles de type IC font l'hypothèse que l'information transite d'un utilisateur à l'autre en suivant les arêtes d'un graphe connu : chaque utilisateur nouvellement infecté peut à son tour infecter chacun de ses successeurs selon des probabilités de diffusion définies sur les lien du graphe. Néanmoins, en plus d'être parfois difficiles à obtenir, les liens explicites d'un réseau social (liens d'amitié par exemple) ne sont pas toujours représentatifs des vrais canaux de diffusion reliant les utilisateurs (Najar *et al.*, 2012 ; Ver Steeg et Galstyan, 2013). Les modèles doivent alors être appliqués à une population d'utilisateurs sans graphe de diffusion connu. Dans ce cas, il est possible de rechercher les liens les plus à même d'expliquer la diffusion (Gomez Rodriguez *et al.*, 2010), ou, plus simplement, de considérer un graphe complet reliant tous les utilisateurs. Malheureusement, considérer le graphe complet du réseau implique une forte complexité algorithmique, le modèle IC apprenant un paramètre indépendant pour *chaque* lien du graphe. En projetant le modèle dans un espace vectoriel, nous réduisons considérablement la complexité du modèle puisque les probabilités sont déduites directement des coordonnées des utilisateurs dans l'espace de projection.

D'autre part, les graphes sociaux possèdent de nombreuses propriétés spécifiques : distribution des degrés en loi de puissance, faible diamètre, etc... (Mislove *et al.*, 2007). Une propriété importante est la présence de *communautés* dans ces graphes, que l'ont peut définir comme des groupe d'utilisateurs similaires, plus densément connectés les uns aux autres. Récemment, (Barbieri *et al.*, 2013) 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 le même autre groupe d'utilisateurs (on parle alors de communautés *bimodales*) :

– S'il y a influence 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 influence entre les utilisateurs a et c (communautés *cohésives*)

– S'il y a influence 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*)

L'impact de ces communautés sur la diffusion d'information a été récemment étudié dans (Barbieri *et al.*, 2013 ; Yang *et al.*, 2014). Dans l'apprentissage clas-

sique du modèle IC, les probabilités de transmission sont estimées sans aucune régularisation relative à la présence de ces communautés. Cela peut conduire à des problèmes de sur-apprentissage ou à des structures de graphes irréalistes. En revanche, bien que sans prise en compte explicite de ces communautés sociales, notre approche par projection des utilisateurs dans un espace de représentation vectoriel nous permet d'extraire naturellement ce genre de régularités. La figure 1 illustre cette approche. Nous projetons les utilisateurs de façon à ce que ceux qui interagissent soient proches les uns des autres. Deux utilisateurs similaires tendent ainsi à se rapprocher, ce qui implique des tendances de diffusion similaires sur les autres utilisateurs du réseau.

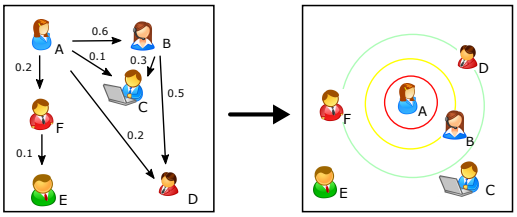

Figure 1. *Passage d'un graphe de diffusion à un espace de représentation vectoriel. À gauche, les valeurs associées aux arrêtes représentent les probabilités de transmission entre utilisateurs. À droite, ces probabilités de transmission sont calculées en fonction de la distance séparant les utilisateurs. Les cercles réprésentent des lignes de niveau d'équiprobabilité depuis l'utilisateur A.*

L'apprentissage de représentations est un domaine de recherche très actif qui a récemment donné lieu à de nombreuses approches efficaces pour diverses tâches mettant en jeu des dépendances relationnelles complexes (Bengio *et al.*, 2013). Ce genre de techniques a par exemple été employé dans les modèles de langage, comme par exemple pour le très populaire modèle *Word2Vec* (Mikolov *et al.*, 2013) où les projections des mots reflètent des similarités syntaxiques et sémantiques, et l'utilisation d'un espace de représentation vectoriel permet de régulariser ces relations. Typiquement, deux mots ayant un sens proche sont projetés plus près l'un de l'autre, et tendent donc à être prédits dans le même contexte. Plus proche de notre approche, un modèle probabiliste de prédiction de playlist a été récemment proposé dans (Chen *et al.*, 2012), dans lequel l'utilisation d'un espace de projection permet d'éviter d'avoir à définir des probabilités d'enchaînement pour chaque paire de morceaux. Notre problème diffère de ces approches par le fait que l'information se diffuse de manière arborescente[2], et non selon une séquence bien définie tel que c'est le cas dans tous les domaines où ce genre de techniques ont été employées. Cela complexifie le problème car les causes des événements observés ne sont pas

[2]Nous nous plaçons ici dans le cadre d'un modèle de type SI (*suceptible-infected*), dans lequel seule la première infection de chaque utilisateur est considérée.

connues : on ne sait pas a priori qui a passé le contenu diffusé à tel utilisateur infecté. Il s'agit alors de faire des suppositions sur le chemin emprunté par l'information pour sélectionner les relations à intensifier dans la représentation à apprendre. Notons que récemment l'utilisation de techniques d'apprentissage de représentation a été explorée pour la prédiction de diffusion d'information (Bourigault *et al.*, 2014), mais uniquement dans un cadre statique, avec un modèle simple de prédiction des infectés finaux à partir d'infections initiales. L'approche proposée ici va plus loin en définissant un modèle itératif qui permet une capture des dynamiques du réseau et une modélisation plus fine de ses relations d'influence.

3. Modèle

3.1. *Notations*

Dans la majorité des applications, la diffusion d'information dans un réseau social est observée sous la forme d'*épisodes de diffusion*. Un épisode de diffusion correspond à une séquence d'actions d'utilisateurs liées à un même contenu. Un tel épisode peut par exemple représenter une série d'utilisateurs ayant posté une vidéo particulière ou ayant utilisé une même expression se diffusant sur le réseau. L'information concernant *qui a été infecté par qui* est donc manquante. Les modèles de type cascades font généralement diverses hypothèses sur cette information manquante pour reconstituer le graphe de diffusion.

Soit un réseau social contenant N utilisateurs $\mathcal{U} = (u_1, ..., u_N)$. Un épisode de diffusion D est représenté par une séquence d'utilisateurs associés à leurs temps d'infection : $\{(u, t^D(u))|u \in \mathcal{U} \wedge t^D(u) < \infty\}$, où $t^D : \mathcal{U} \rightarrow \mathbb{R}^+ \cup \{\infty\}$ retourne l'instant auquel chaque utilisateur a été infecté dans l'épisode D, ou ∞ pour les utilisateurs non infectés (on considère que le premier utilisateur infecté l'est au temps $t = 0$, et que les autres temps sont donnés relativement à celui-ci). Nous notons $D(t)$ l'ensemble des utilisateurs infectés dans D avant le temps t : $D(t) = \{u \in \mathcal{U}|t^D(u) < t\}$, et $\bar{D}(t)$ son complémentaire. L'ensemble des utilisateurs infectés au cours de l'épisode D est noté $D(\infty)$.

3.2. *Modèle de Diffusion*

Comme évoqué plus haut, nous nous basons sur le modèle IC, que nous proposons de projeter dans un espace de représentation vectoriel, pour régulariser les relations entre les utilisateurs et en réduire la complexité spatiale. Néanmoins, le modèle IC classique considère que le temps est découpé en plages et que des transmissions de contenu sont uniquement possibles entre plages de temps successifs. Cela limite grandement la capacité d'explication d'épisodes de diffusion observés dans des données réelles, la diffusion correspondant à un processus essentiellement asynchrone. D'autre part, chercher à extraire

des régularités sur des délais de diffusion observés, tel que tentent de le faire des modèles comme *NetRate* (Gomez-Rodriguez *et al.*, 2011) ou *CTIC* (Saito *et al.*, 2009), paraît difficile lorsque l'on s'intéresse à des données sociales extraites de réseaux impliquant des échanges entre humains (souvent très bruités). Cela perturbe bien souvent l'inférence des probabilités de transmission de contenu entre les utilisateurs (Lamprier *et al.*, 2015). Nous considérons alors ici une version légèrement modifiée d'IC, dans laquelle les délais d'infections suivent une loi uniforme.

Suivant un processus de diffusion itératif, le modèle IC considère que chaque utilisateur u nouvellement infecté dispose d'une chance *unique* d'infecter chaque autre utilisateur v non encore infecté avec une probabilité $P_{u,v}$. Dans ce contexte, la probabilité qu'un utilisateur v devienne infecté par n'importe quel utilisateur d'un ensemble "d'infecteurs potentiels" $I \subseteq \mathcal{U}$ est définie selon :

$$P(v|I) = 1 - \prod_{u \in I} (1 - P_{u,v}) \quad [1]$$

où $P_{u,v}$ est la probabilité de transmission d'information de u vers v. $P(v|I)$ est donc la probabilité qu'au moins un utilisateur de I transmette l'information à v. Dans le modèle IC classique, chaque probabilité $P_{u,v}$ est définie de manière indépendante. Nous proposons ici de définir ces probabilités selon une fonction $f : \mathbb{R}^d \times \mathbb{R}^d \rightarrow [0, 1]$ considérant les positions relatives des utilisateurs dans l'espace de projection, avec d la taille de l'espace considéré :

$$P_{u,v} = f(z_u, \omega_v) \quad [2]$$

où $z_u \in \mathbb{R}^d$ est la représentation de u en tant qu'*émetteur* de contenu et $\omega_v \in \mathbb{R}^d$ celle de v en tant que *récepteur*. Deux ensembles de coordonnées sont donc définis pour chaque utilisateur : $Z = (z_u)_{u \in \mathcal{U}}$ et $\Omega = (\omega_u)_{u \in \mathcal{U}}$. Avec un seul ensemble, les probabilités de transmissions seraient symétriques, ce qui n'est pas réaliste dans le cadre de la diffusion sur les réseaux sociaux.

La fonction f peut être définie de multiples façons, la seule contrainte étant qu'elle doive fournir une valeur dans $[0, 1]$ de manière à ce qu'elle puisse être interprétée comme une probabilité. Nous proposons de considérer la fonction logistique suivante :

$$f(z_u, \omega_v) = \frac{1}{1 + exp(z_u^{(0)} + \omega_v^{(0)} + \sum_{i=1}^{d-1} (z_u^{(i)} - \omega_v^{(i)})^2)} \quad [3]$$

où $x^{(i)}$ est la i-ème composante du vecteur x. Le choix d'une fonction logistique paraît pertinent puisque ce genre de fonction permet de définir des probabilités décroissantes en fonction de la distance qui sépare les représentations émetteur 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 pour

les 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 dimension de chaque représentation comme une valeur de biais : $z_u^{(0)}$ et $\omega_v^{(0)}$ modélisent respectivement un a priori sur la tendance générale de u à transmettre de l'information et un a priori sur la tendance générale de v à en recevoir.

3.3. *Algorithme d'apprentissage*

Selon l'ensemble de probabilités de transmission $\mathcal{P} = \{P_{u,v} | (u,v) \in \mathcal{U}^2\}$, la vraisemblance d'un épisode de diffusion D donné peut s'écrire (si l'on se focalise sur les ordres partiels de contamination plutôt que sur les instants d'infection exacts (Lamprier *et al.*, 2015)) :

$$P(D) = \prod_{v \in D(\infty)} P_v^D \prod_{v \in \bar{D}(\infty)} (1 - P_v^D) \quad [4]$$

avec $P_v^D = P(v|D(t^D(v)))$ la probabilité d'observer l'infection de l'utilisateur v connaissant l'ensemble d'utilisateurs infectés avant v dans D. Il est alors possible de considérer la log-vraisemblance d'un ensemble d'épisodes $\mathcal{D}$:

$$\mathcal{L}(\mathcal{P}; \mathcal{D}) = \sum_{D \in \mathcal{D}} \Big(\sum_{v \in D(\infty)} log(P_v^D) + \sum_{v \in \bar{D}(\infty)} log(1 - P_v^D) \Big) \quad [5]$$

La définition de P_v^D rend cette log-vraisemblance difficile à optimiser directement. Cependant, en nous basant sur (Saito *et al.*, 2008), nous pouvons définir un algorithme de type Espérance-Maximisation (EM), basé sur la fonction d'espérance suivante :

$$\mathcal{Q}(\mathcal{P}|\hat{\mathcal{P}}) = \sum_{D \in \mathcal{D}} \left(\Phi^D(\mathcal{P}|\hat{\mathcal{P}}) + \sum_{v \in \bar{D}(\infty)} \sum_{u \in D(\infty)} log(1 - P_{u,v}) \right) \quad [6]$$

où $\Phi^D(\mathcal{P}|\hat{\mathcal{P}})$ correspond à l'espérance de la valeur du premier terme de la fonction de log-vraisemblance de D selon les paramètres courants $\hat{\mathcal{P}}$. Cette valeur est calculée en considérant les probabilités conditionnelles de succès des différentes tentatives de transmission du contenu diffusé : Sachant que v est infecté dans D avec une probabilité $\hat{P_v^D}$ (calculée selon l'équation 1 avec l'estimation courante des probabilités de transmission), la probabilité conditionnelle que la transmission de u vers v ait eu lieu est donnée par $\hat{P_{u,v}}/\hat{P_v^D}$.

$$\Phi^D(\mathcal{P}|\hat{\mathcal{P}}) = \sum_{v \in D(\infty)} \sum_{u \in D(t^D(v))} \frac{\hat{P_{u,v}}}{\hat{P_v^D}} \log(P_{u,v}) + (1 - \frac{\hat{P_{u,v}}}{\hat{P_v^D}}) \log(1 - P_{u,v}) \quad [7]$$

Du fait de l'utilisation d'un espace de représentation vectoriel, les probabilités de transmission entre chaque paire d'utilisateurs ne sont plus libres. Les contraintes géométriques liées à l'espace rendent leurs valeurs interdépendantes. Maximiser une telle log-vraisemblance ne peut alors plus se décomposer en un ensemble de sous-problèmes convexes comme cela peut être le cas avec un modèle IC classique. Néanmoins, il est possible de définir une procédure de montée de gradient stochastique convergeant vers un maximum local efficace, lorsque l'on détermine un nombre de dimensions adapté (25 dimensions semble être un bon compromis, cf. section 4).

L'algorithme 1 détaille la procédure utilisée pour apprendre Z et Ω. Il s'agit d'un algorithme EM où l'étape de maximisation effectue un pas de gradient. Une itération se déroule ainsi :

1) Ligne 7 : tirage uniforme d'un épisode de diffusion D et d'un utilisateur v n'étant pas la source de D ;

2) Lignes 9 à 15 : si v fait partie de D, calcul des estimations courantes $\hat{P_v^D}$ et $\hat{P_{u,v}}$ pour chaque utilisateur u infecté avant v (selon les formules 1 et 2 en utilisant les valeurs courantes de z_u et ω_v) ;

3) Lignes 16 à 25 : mise à jour des valeurs de Z et Ω de façon à augmenter la valeur de $\mathcal{Q}(\mathcal{P}|\hat{\mathcal{P}})$. Si v est infecté dans D, cette mise à jour fait intervenir la dérivée de $\Phi^D(\mathcal{P}|\hat{\mathcal{P}})$ (lignes 19 à 22). Sinon, elle fait intervenir la dérivée de $\log(1 - P_{u,v})$ (lignes 22 à 24). Le pas d'apprentissage ϵ est fixé à 10^{-4}.

4) Lignes 27 à 34 : 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 significativement augmenté depuis le dernier test.

Notons enfin qu'à cause du tirage effectué à la ligne 7, les différentes paires (u, v) ne sont pas associées aux épisodes de $\mathcal{D}$ dans les mêmes proportions que dans la formule 6 : les paires d'utilisateurs apparaissant dans les épisodes plus courts sont tirées plus souvent. Nous calculons donc, en ligne 8, un poids α permettant de corriger ce biais.

$$\alpha = \frac{|\mathcal{D}| \times \mid \bar{D}(1) \mid}{\sum_{D \in \mathcal{D}} \sum_{u \in D(\infty)} |\bar{D}(t^D(u) + 1)|}$$

Ce poids est considéré lors de la mise à jour des paramètres (lignes 16 à 25).

4. Expériences

4.1. *Jeux de données*

Dans nos expériences, les jeux de données suivants sont utilisés :

– **Digg** : corpus constitué à partir du portail d'information collaboratif *Digg*, sur lequel les utilisateurs postaient des messages auxquels d'autres utilisateurs

Algorithme 1 : Apprentissage du modèle IC projeté

Entrées :

$\mathcal{U}$: 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 :

$Z = \{\forall u \in \mathcal{U} : z_u \in \mathbb{R}^d\}$; $\Omega = \{\forall u \in \mathcal{U} : \omega_u \in \mathbb{R}^d\}$;

$nbProbas \leftarrow \sum_{D\in\mathcal{D}} \sum_{u\in D(\infty)} |\bar{D}(t^D(u)+1)|$;
pour $u \in \mathcal{U}$ **faire**
  Tirage uniforme de $z_u \in [-1,1]^d$; Tirage uniforme de $\omega_u \in [-1,1]^d$;
fin
$oldL \leftarrow -\infty$; $it \leftarrow 0$;
tant que *true* **faire**
  Tirage uniforme de $D \in \mathcal{D}$ et $v \in \bar{D}(1)$;
  $\alpha \leftarrow |\mathcal{D}| \times |\ \bar{D}(1)\ | / nbProbas$;
  si $t^D(v) < \infty$ **alors**
    $\hat{P_v^D} \leftarrow 1$;
    pour $u \in D(t^D(v))$ **faire**
      $\hat{P_{u,v}} \leftarrow f(z_u, \omega_v)$; $\hat{P_v^D} \leftarrow \hat{P_v^D} \times (1 - \hat{P_{u,v}})$;
    fin
    $\hat{P_v^D} \leftarrow 1 - \hat{P_v^D}$;
  fin
  pour $u \in D(t^D(v))$ **faire**
    $\xi_u^+ \leftarrow \dfrac{\partial \log f(z_u,\omega_v)}{\partial z_u}$; $\xi_u^- \leftarrow \dfrac{\partial \log(1-f(z_u,\omega_v))}{\partial z_u}$;
    $\xi_v^+ \leftarrow \dfrac{\partial \log f(z_u,\omega_v)}{\partial \omega_v}$; $\xi_v^- \leftarrow \dfrac{\partial \log(1-f(z_u,\omega_v))}{\partial \omega_v}$;
    si $t^D(v) < \infty$ **alors**
      $z_u \leftarrow z_u + \alpha \times \epsilon \times \big(\dfrac{\hat{P_{u,v}}}{\hat{P_v^D}}\xi_u^+ + (1 - \dfrac{\hat{P_{u,v}}}{\hat{P_v^D}})\xi_u^-\big)$;
      $\omega_v \leftarrow \omega_v + \alpha \times \epsilon \times \big(\dfrac{\hat{P_{u,v}}}{\hat{P_v^D}}\xi_v^+ + (1 - \dfrac{\hat{P_{u,v}}}{\hat{P_v^D}})\xi_v^-\big)$;
    sinon
      $z_u \leftarrow z_u + \alpha \times \epsilon \times \xi_u^-$; $\omega_v \leftarrow \omega_v + \alpha \times \epsilon \times \xi_v^-$;
    fin
  fin
  $it \leftarrow it + 1$;
  si $it\ mod\ freq = 0$ **alors**
    $\mathcal{P} \leftarrow \{P_{u,v} | (u,v) \in \mathcal{U}^2 \wedge P_{u,v} = f(z_u, \omega_v)\}$;
    $L \leftarrow$ Calcul de la log-vraissemblance selon (5) avec $\mathcal{P}$;
    si $L \leq oldL$ **alors**
      retourner (Z, Ω);
    fin
    $oldL \leftarrow L$;
  fin
fin

Corpus	Util.	Liens	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

Tableau 1. *Quelques statistiques sur les jeux de données.*

pouvaient ensuite attribuer des "Diggs". Chaque épisode regroupe le message initial et l'ensemble des "Diggs" correspondants.

– **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 et 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 forum particulier.

– **Twitter** : corpus correspondant à l'activité sur Twitter d'un ensemble d'environ 5000 utilisateurs durant 2 semaines. Chaque épisode regroupe l'ensemble des utilisations d'un hashtag particulier.

– **Memetracker** : corpus contenant des millions d'articles de blogs (Leskovec *et al.*, 2009). Chaque épisode regroupe les utilisations d'une des expressions récurrentes extraites par (Leskovec *et al.*, 2009).

– **ICWSM** : corpus contenant plusieurs millions de posts de blogs collectés en 2009. À la manière de (Gomez-Rodriguez *et al.*, 2011), chaque épisode correspond à une composante connexe du graphe des citations entre messages.

Comme expliqué en section 3.2, nous ne disposons *pas* de graphe social sur lequel apprendre les modèles. Pour les modèles de référence basés sur un graphe, une possibilité serait de considérer le graphe complet reliant toutes les paires d'utilisateurs du réseau considéré. Néanmoins, pour réduire la complexité, nous limitons grandement le nombre de liens considérés lors de l'apprentissage en nous appuyant sur le graphe de "co-participations" aux épisodes de diffusion : un lien (u, v) est envisagé si et seulement si il existe au moins un épisode de diffusion D contenant u et v, avec $t^D(u) < t^D(v)$. À noter que cela aboutit pour les modèles de référence envisagés (voir ci-après), après convergence de l'apprentissage, au même ensemble de probabilités que si nous avions considéré un graphe complet, car les liens sans exemples de diffusion obtiennent une probabilité de transmission nulle.

La table 1 donne quelques statistiques sur les jeux de données utilisés : nombre d'utilisateurs, nombre de liens et densité du graphe de co-participation, nombre d'épisodes en apprentissage et en test et enfin taille moyenne des épisodes de diffusion.

4.2. *Modèles de référence*

Nous comparons notre modèle aux approches suivantes :

– **IC** : version du modèle *Independent Cascades* décrite dans (Lamprier *et al.*, 2015).

– **Netrate** : modèle génératif définissant une distribution des délais de transmission sur chaque lien du graphe social de façon à reproduire les temps d'infection observés (Gomez-Rodriguez *et al.*, 2011).

– **CTIC** : extension du modèle IC, dans laquelle chaque lien est associé à deux paramètres contrôlant la probabilité et le délai de transmission entre ses utilisateurs.

4.3. *Prédiction de diffusion*

La principale tâche sur laquelle nous comparons les modèles pour en évaluer les performances est une tâche en prédiction de diffusion : le but est de retrouver, à partir d'une ou plusieurs sources $D(1)$ d'un épisode de diffusion D, l'ensemble des utilisateurs finalement infectés $D(\infty)$. Du fait de la complexité induite par la présence probable de cycles dans les graphes de diffusion appris, il est impossible de calculer directement la probabilité d'infection finale de chaque utilisateur pour un épisode donné. La procédure d'inférence emploie alors un processus de Monte-Carlo pour estimer ces probabilités : à partir d'un ensemble d'utilisateurs initial $D(1)$, nous simulons 1000 processus de diffusion selon chaque modèle, et estimons ainsi pour chaque utilisateur sa probabilité $P_u^{\mathcal{D}}$ d'appartenir à $D(\infty)$.

Les modèles sont ensuite évalués selon plusieurs mesures :

– **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 $\log(0)$

– **MAP** : les utilisateurs sont classés par ordre décroissant de probabilités d'infection, et la liste est évaluée par Mean-Average-Precision.

– **F1** : la mesure classique $F1$, définie comme suit :

$$p = \frac{\sum_{u \in \mathcal{D}(\infty)} P_u^{\mathcal{D}}}{\sum_{u \in \mathcal{U}} P_u^{\mathcal{D}}} \qquad r = \frac{\sum_{u \in \mathcal{D}(\infty)} P_u^{\mathcal{D}}}{|\mathcal{D}(\infty)|} \qquad F1 = \frac{2 * p.r}{p + r}$$

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 2 que les résultats obtenus sur Digg, mais les tendances observées sur les autres

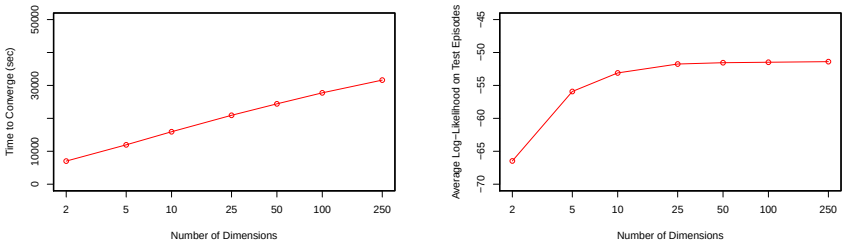

Figure 2. *Durée (en secondes) de l'apprentissage de notre modèle jusqu'à convergence, et log-vraisemblance obtenue en test, sur le jeu de données Digg.*

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.

Les résultats sont moyennés et regroupés dans le tableau 2. Nous pouvons tout d'abord remarquer que notre modèle (Embedded IC) obtient des résultats toujours *au moins aussi bons* que ceux des autres modèles pour une complexité bien inférieure. Les résultats assez mauvais obtenus par CTIC et NetRate peuvent être expliqués par le fait qu'il est difficile d'extraire des régularités sur les délais de transmission. Les bons résultats obtenus par notre approche montrent que nous pouvons projeter correctement les probabilités de transmission dans un espace vectoriel. Il est important de remarquer que cela se fait avec un nombre de paramètres (colonne de droite) largement inférieur au modèle IC, sauf sur le corpus ICWSM, dont le graphe de co-participations est très creux. Nous remarquons également que les mesures se comportent différemment :

– La MSE de notre modèle est la meilleure sur Irvine, Digg et LastFM. Sur ces jeux de données, 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'intérêts. Notre approche par apprentissage de représentations continues parvient à mieux identifier les similarités entre utilisateurs, ce qui conduit à une meilleur MSE (voir section 4.4).

– La log-vraisemblance de notre modèle est meilleure sur Irvine, ICWSM, Digg et Twitter. Ces jeux de données ont en commun qu'ils possèdent un graphe de co-participations bien moins dense que les autres corpus (table 1). Cela implique qu'un modèle classique, basé sur le graphe, ne pourra apprendre qu'un petit nombre de probabilités, et sera donc incapable de prédire la transmission d'information entre la plupart des utilisateurs. Notre approche, en revanche, est capable d'étendre des observations de transmission à des couples non observés.

Corpus	Modèle	MSE	LogVrai.	MAP	F1	nbParams
Irvine	IC	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
	Embedded IC	**14,53***	**-532,5***	0,079	**0,025***	42350
ICWSM	IC	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
	Embedded IC	**0,19**	**-6,14***	**0,78**	**0,651**	113500
MemeTracker	IC	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
	Embedded IC	**32,15**	**-791,3**	**0,23**	**0,0632***	24900
Digg	IC	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
	Embedded IC	**1,79***	**-51,75***	**0,434***	0,198	164750
Twitter	IC	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
	Embedded IC	**5,47***	**-223,15***	**0,056***	**0,013**	142050
LastFM	IC	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
	Embedded IC	**11,62***	**-405**	**0,151***	**0,027**	49300

Tableau 2. *Résultats obtenus sur les différents corpus. Les valeurs marquées d'un astérisque sont significativement meilleures que celles obtenues par IC (Test-t de Student 95%), et celles en gras indiquent le meilleur résultat obtenu sur chaque corpus.*

– Enfin, notre modèle obtient une meilleure MAP sur Digg, Twitter et LastFM. Ces sites web ont également une propriété particulière : ils fonctionnent comme des "portails". Quand un utilisateur se connecte sur un de ces sites, la première chose qu'il trouve est une liste d'items populaires (même Twitter dispose d'une section "Trends"). Les interactions utilisateurs-utilisateurs sont ainsi moins importantes que dans les autres jeux de données. Ainsi, sur ces sites, la diffusion a lieu de manière plus globale : plus une information est populaire, plus elle a de chances d'infecter de nouveaux utilisateurs. Ce phénomène est mieux représenté par notre modèle, où chaque utilisateur a au moins une petite chance d'infecter tous les autres.

Les résultats obtenus nous apprennent 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 sont propre "type" de diffusion (retweets, discussion, recommandation). Cela explique pourquoi les modèles se comportent aussi différemment sur chaque corpus. Néanmoins, d'une façon générale les résultats mettent en avant les propriétés avantageuses

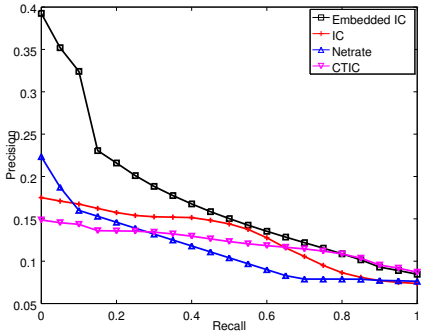

Figure 3. *Précision-Rappel de la detection de relations sur Memetracker.*

de notre approche : moins de paramètres à apprendre, une meilleure généralisabilité et une meilleure capacité à modéliser différents types de diffusion.

4.4. *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. À l'instar de (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. La présence d'un hyperlien entre deux messages rend en effet compte de l'influence qu'a eu le message pointé sur celui qui le référence, et donc de l'influence de l'auteur du premier sur l'auteur du second.

Pour chaque paire (u, v), la probabilité $P_{u,v}$ 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 3. Cette expérience illustre la meilleure capacité de notre modèle à inférer des liens d'influence à partir d'épisodes de diffusion, sa courbe de precision en fonction du rappel se situant clairement au dessus de celle des autres modèles. Le fait de projeter les utilisateurs dans un espace vectoriel pour définir les probabilités de transmission permet au modèle de découvrir des relations sans exemple de diffusion dans les épisodes observés. En outre, la régularisation induite par les contraintes géométriques de l'espace permet de minorer la probabilité de certains liens non pertinents, grâce notamment à l'effet repoussoir de certains utilisateurs projetés proches de leur utilisateur destination selon la représentation récepteur.

5. Conclusion

La projection d'un graphe de relations sociales dans un espace vectoriel permet d'en extraire des régularités, afin de découvrir de nouvelles relations ou au contraire d'identifier des relations peu pertinentes. Cela peut s'avérer très bénéfique pour des tâches autour de la diffusion d'information, puisque permettant de se concentrer sur les canaux principaux de communication du réseau. De plus, l'utilisation d'un espace vectoriel réduit l'impact de la présence importante de bruit dans les données sociales collectées, ainsi que le nombre de paramètres à apprendre et à stocker pour les réseaux denses. Dans cet esprit, notre proposition d'apprendre les probabilités du modèle *Independent Cascades* par projection des utilisateurs dans un espace de représentation nous a permis d'observer de bonnes performances pour les tâches de prédiction de diffusion et de découverte de relations d'influence.

La piste principale de poursuite du travail mené concerne la considération de l'objet des diffusions dans l'apprentissage du modèle. Nous envisageons d'apprendre des opérateurs de transformation de l'espace de représentation en fonction du contenu diffusé. D'autre part, nous envisageons l'application du processus d'apprentissage proposé ici dans un cadre "en ligne", de manière à capturer les évolutions du réseau d'influence au cours du temps.

Remerciements

Ce travail a été effectué avec le support des projets REQUEST (Investissement d'avenir, 2014-2017) et LOCUST (ANR, 2015-2018).

6. Bibliographie

Bakshy E., Rosenn I., Marlow C., Adamic L., « The role of social networks in information diffusion », *WWW'12*, ACM, p. 519-528, 2012.

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

Bengio Y., Courville A. C., Vincent P., « Representation Learning : A Review and New Perspectives », *IEEE TPAMI*, vol. 35, nº 8, p. 1798-1828, 2013.

Bordes A., Usunier N., García-Durán A., Weston J., Yakhnenko O., « Translating Embeddings for Modeling Multi-relational Data », *NIPS'13*, p. 2787-2795, 2013.

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

Bourigault S., Lamprier S., Gallinari P., « Representation Learning for Information Diffusion through Social Networks : an Embedded Cascade Model », *WSDM'16*, ACM, p. 573-582, 2016.

Celma O., *Music Recommendation and Discovery in the Long Tail*, Springer, 2010.

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

Cheng J., Adamic L., Dow P. A., Kleinberg J. M., Leskovec J., « Can Cascades Be Predicted ? », *WWW'14*, ACM, p. 925-936, 2014.

Goldenberg J., Libai B., Muller E., « Talk of the network : A complex systems look at the underlying process of word-of-mouth », *Marketing letters*, vol. 12, nº 3, p. 211-223, 2001.

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

Gomez Rodriguez M., Leskovec J., Krause A., « Inferring networks of diffusion and influence », *KDD'10*, ACM, 2010.

Gruhl D., Guha R., Liben-Nowell D., Tomkins A., « Information Diffusion Through Blogspace », *WWW'04*, ACM, p. 491-501, 2004.

Guille A., Hacid H., « A predictive model for the temporal dynamics of information diffusion in online social networks », *WWW Companion*, ACM, 2012.

Kempe D., Kleinberg J., Tardos E., « Maximizing the spread of influence through a social network », *KDD'03*, ACM, p. 137-146, 2003.

Lagnier C., Denoyer L., Gaussier E., Gallinari P., « Predicting Information Diffusion in Social Networks using Content and User's Profiles », *ECIR '13*, 2013.

Lamprier S., Bourigault S., Gallinari P., « Extracting Diffusion Channels from Real-World Social Data : a Delay-Agnostic Learning of Transmission Probabilities », *ASONAM'15*, IEEE Computer Society, 2015.

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

Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J., « Distributed Representations of Words and Phrases and their Compositionality », *NIPS'13*, 2013.

Mislove A., Marcon M., Gummadi K. P., Druschel P., Bhattacharjee B., « Measurement and Analysis of Online Social Networks », *IMC '07*, ACM, p. 29-42, 2007.

Najar A., Denoyer L., Gallinari P., « Predicting information diffusion on social networks with partial knowledge », *WWW '12 Companion*, ACM, p. 1197-1204, 2012.

Opsahl T., Panzarasa P., « Clustering in weighted networks », *Social networks*, vol. 31, nº 2, p. 155-163, 2009.

Saito K., Kimura M., Ohara K., Motoda H., « Learning Continuous-Time Information Diffusion Model for Social Behavioral Data Analysis », *ACML '09*, Springer-Verlag, p. 322-337, 2009.

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

Saito K., Ohara K., Yamagishi Y., Kimura M., Motoda H., « Learning Diffusion Probability Based on Node Attributes in Social Networks. », *ISMIS'11*, Springer, p. 153-162, 2011.

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

Yang J., McAuley J., Leskovec J., « Detecting Cohesive and 2-mode Communities Indirected and Undirected Networks », *WSDM '14*, ACM, p. 323-332, 2014.