



**HAL**  
open science

# Apprentissage et Inférence Structurale pour l'Extraction de Dynamiques de l'Information

Sylvain Lamprier

► **To cite this version:**

Sylvain Lamprier. Apprentissage et Inférence Structurale pour l'Extraction de Dynamiques de l'Information. Apprentissage [cs.LG]. Sorbonne Université, 2020. tel-03060061

**HAL Id: tel-03060061**

**<https://hal.sorbonne-universite.fr/tel-03060061>**

Submitted on 13 Dec 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## **Habilitation à Diriger des Recherches**

Spécialité

**Informatique**

**Sylvain LAMPRIER**

Apprentissage et Inférence Structurale pour  
l'Extraction de Dynamiques de l'Information :  
Capture et Prédiction dans les Réseaux Sociaux

Soutenue publiquement le 23 Septembre 2020

Devant le jury composé de :

M. Eric Gaussier	Président du jury	Université de Grenoble - LIG,
M. Olivier Pietquin	Rapporteur	Google Brain,
M. Philippe Preux	Rapporteur	Université de Lille - CRISTAL,
M. Shengrui Wang	Rapporteur	Université de Sherbrooke (Québec),
M. Patrick Gallinari	Examineur	Sorbonne Université - LIP6,
M. Nicolas Thome	Examineur	CNAM - CEDRIC



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Capture de données sur les réseaux</b>	<b>7</b>
2.1	Collecte temps réel ciblée . . . . .	8
2.1.1	Sélection dynamique de flux . . . . .	9
2.1.2	Algorithmes de bandits pour la collecte de données dynamique . . . . .	13
2.2	Bandits contextuels pour la sélection active de flux de données . . . . .	19
2.2.1	Algorithmes de bandits contextuels . . . . .	19
2.2.2	Collecte structurée avec profils cachés . . . . .	24
2.2.3	Collecte dynamique contextuelle . . . . .	33
2.2.4	Collecte dynamique relationnelle . . . . .	43
2.3	Conclusions et Perspectives . . . . .	49
2.3.1	Inférence Bayésienne . . . . .	49
2.3.2	Réseaux de neurones probabilistes . . . . .	51
2.3.3	Récompenses dynamiques . . . . .	53
<b>3</b>	<b>Diffusion d'information sur les réseaux</b>	<b>57</b>
3.1	Diffusion Bayésienne : application et extension du modèle IC . . . . .	58
3.1.1	Modèles de cascade . . . . .	60
3.1.2	Inférence de Graphes de Diffusion . . . . .	61
3.1.3	Application et Extension d'IC . . . . .	64
3.2	Apprentissage de représentation pour la diffusion dans les réseaux . . . . .	69
3.2.1	Apprentissage de représentation et diffusion . . . . .	69
3.2.2	Modèle de Cascade à représentations distribuées . . . . .	71
3.3	Modèles neuronaux récurrents pour la diffusion dans les réseaux . . . . .	75
3.3.1	Réseaux de neurones récurrents pour la diffusion . . . . .	76
3.3.2	Modèle de Cascade Récurrent . . . . .	78
3.4	Conclusions et Perspectives . . . . .	85
3.4.1	Modèles graphiques bayésiens . . . . .	85
3.4.2	Modèles propagatifs (plus) profonds . . . . .	91
3.4.3	Modèles en ligne . . . . .	96
<b>4</b>	<b>Évolution temporelle dans les communautés d'auteurs</b>	<b>103</b>
4.1	Modélisation de l'évolution langagière au cours du temps . . . . .	104
4.2	Modèle de Langue Dynamique Récurrent . . . . .	106
4.2.1	Modèle à États Temporels pour la Génération de Documents Textuels . . . . .	107

TABLE DES MATIÈRES

---

4.2.2	Apprentissage par Inférence Variationnelle Temporelle . . . . .	108
4.3	Apprentissage de représentations dynamiques dans les communautés d’auteurs . . . . .	111
4.3.1	Trajectoires Individuelles Déterministes . . . . .	112
4.3.2	Trajectoires Individuelles Stochastiques . . . . .	114
4.4	Conclusions et Perspectives . . . . .	117
4.4.1	Structuration de l’espace latent . . . . .	118
4.4.2	Prédiction stochastique inductive . . . . .	122
4.4.3	Modèles en ligne . . . . .	124
<b>5</b>	<b>Conclusion</b>	<b>127</b>



# Chapitre 1

## Introduction

Les échanges d'informations entre individus sont au cœur de toute organisation de société, puisqu'ils permettent d'en structurer les normes et comportements, en perpétuelle évolution. Ces échanges, qui sont motivés par des besoins de communication de contenus, faits ou opinions, découlent de divers jeux de domination, d'influence et phénomènes d'imitation au sein de la population. Nombre de sociologues, dont Gabriel Tarde [Tarde, 1890], placent l'imitation au principe même de toute activité humaine et conçoivent la société comme un ensemble d'individus qui s'imitent entre eux. Le terme même, introduit pour la première fois dans [Dawkins, 1976], vise à décrire toute "unité d'information contenue dans un cerveau, échangeable au sein d'une société". C'est "un élément d'une culture pouvant être considéré comme transmis par des moyens non génétiques, en particulier par l'imitation" [Jouxte, 2005]. Toute culture est alors constituée d'unités échangeables, véhiculées d'individus à d'autres en subissant d'éventuelles variations. Être à même d'extraire les dynamiques en jeu dans les échanges d'information d'une société est alors crucial pour en comprendre le fonctionnement et l'analyser.

Les phénomènes de propagation d'information dans les populations ont suscité de nombreuses études dès les années 1950, avec notamment des travaux sur la diffusion et l'acceptation d'innovations [Ryan and Gross, 1950], qui montrent l'importance des relations sociales dans l'adoption de nouveaux produits. À tel point que par exemple dans le milieu médical, des études ont montré que les phénomènes de bouche à oreille sont souvent plus efficaces que la publication d'études scientifiques pour convaincre des praticiens à la prescription de nouveaux traitements [Bauer and Wortzel, 1966]! Comprendre les mécanismes en jeu est alors primordial pour analyser la structuration d'une communauté, y favoriser la diffusion d'informations utiles et couper court à la propagation de croyances erronées. Toutefois, les interactions entre individus, ou groupes d'individus, relèvent de relations sociales de diverses natures, souvent difficiles à appréhender. À moins de disposer d'un très grand volume de données, la modélisation des dynamiques de propagation dans la population ne peut se faire qu'à un niveau macroscopique, tel que dans le modèle fondateur de [Bass, 1969], qui définit l'évolution du nombre d'individus adoptant tel ou tel comportement selon un ensemble d'équations différentielles. Cela limite à une compréhension globale des phénomènes de propagation dans une communauté, sans permettre l'analyse plus fine des types d'interaction en jeu et groupes d'individus impliqués dans telle ou telle diffusion de contenu ou de comportement.

L'avènement des réseaux sociaux a constitué une révolution pour de nombreux chercheurs, tant dans le domaine de la sociologie qu'en sciences de l'information, car ils permettent l'enregistrement d'une trace d'humanité qu'il est possible d'exploiter pour extraire diverses informations sur les dyna-

miques de la société. Alors qu'avant leur apparition les études se cantonnaient à des données simples, échangées sur un maillage d'entités relativement limité, la quantité et la richesse des informations échangées sur les media sociaux est telle que les difficultés de constitution de jeux de données sont alors remplacées par de nouvelles problématiques de stockage, filtrage et traitement des masses de données collectées. Être à même de sélectionner les contenus utiles selon un besoin d'analyse donné est un enjeu majeur en fouille de données sociales, lorsque ces données arrivent en flux continu à des fréquences élevées (à titre illustratif, 350 000 messages par minute sont échangés en moyenne sur Twitter), d'autant plus pour les types de tâche à composante relationnelle qui nous intéressent dans ce manuscrit, où chaque contenu ne peut être considéré de manière individuelle mais doit être mis en regard des données précédemment collectées. Au cœur de ce problème de sélection de données se pose également le paradoxe de la rareté des données individuelles dans les très grands volumes de données sociales (*Big-Data paradox* [Liu et al., 2016]) : alors que l'on dispose globalement de très nombreuses informations, les données collectées à l'échelle des individus sont souvent très parcimonieuses, les informations individuelles se trouvant disséminées dans la masse. Des méthodes adaptées doivent alors être développées pour combler ces manques d'information, en exploitant conjointement structure et temporalité des observations.

Depuis mon recrutement en tant que maître de conférences au LIP6 en 2009, ma recherche s'est focalisée sur la proposition de méthodes d'apprentissage statistique pour données structurées, notamment donc pour l'extraction de dynamiques de l'information dans les réseaux sociaux. Mon équipe d'accueil étant spécialisée dans l'apprentissage profond, une large partie des méthodes que j'ai pu développer dans ce cadre repose sur des réseaux de neurones. Ce n'est cependant pas une constante dans mes recherches, puisque l'apprentissage de ce genre de modèle, impliquant généralement des optimisations itératives à base de gradients, peut s'avérer parfois difficile pour la prise de décision en ligne à partir de flux continu de données. Des modèles plus simples, par exemple linéaires, peuvent se révéler plus efficaces dans certains cas. Non, ce qui unifie plus généralement ma recherche est plutôt le cadre bayésien sur lequel la grande majorité de mes travaux repose. La formulation générative de l'apprentissage statistique m'est toujours apparue comme à la fois plus puissante et élégante que sa contre-partie discriminative, permettant une meilleure mise en évidence des hypothèses de travail et fournissant un cadre mieux fondé pour la capture de l'incertitude des modèles. Les réponses fournies par les modèles ne sont, comme dans la vie, jamais des vérités absolues, mais exhibent une part d'incertitude qu'il faut pouvoir considérer. La quasi-totalité des travaux mentionnés dans ce manuscrit repose donc sur la définition de modèles stochastiques génératifs, avec apprentissage de représentations et inférence bayésienne de variables latentes.

Alors que la plupart des travaux de la littérature sur le traitement de données sociales supposent la connaissance a priori de la structure des entités relationnelles qu'ils manipulent, l'ensemble de mes recherches se place dans un contexte de graphe inconnu. Lorsqu'elles sont disponibles, les relations explicites entre entités des réseaux sociaux ne sont en effet souvent que peu représentatives des dynamiques de l'information circulant sur ces media [Huberman et al., 2008]. L'apprentissage de modèles dans ce cadre se fait alors à partir de données très incomplètes, puisqu'en plus de n'avoir à disposition que des informations partielles sur les individus et leurs rôles dans les échanges d'informations sur les réseaux considérés, les relations entre les entités de ces réseaux sont donc inconnues a priori. Néanmoins, un point commun de la plupart des travaux présentés dans ce manuscrit est la prise en compte conjointe des aspects structurels et temporels des données considérées, tous deux primordiaux pour l'extraction des dynamiques d'échanges sur les réseaux, via notamment la modélisation des divers jeux d'influence entre individus ou groupes d'individus, et l'établissement de modèles prédictifs efficaces à l'échelle microscopique. À plusieurs reprises, nous avons donc recours

à des méthodes d'inférence structurelle bayésienne, notamment variationnelle, pour l'estimation de distributions probabilistes de diverses variables cachées, sur lesquelles les algorithmes d'apprentissage développés peuvent se baser.

Ce manuscrit s'articule autour de trois grandes problématiques pour l'extraction de dynamiques de l'information :

- Dans un premier chapitre nous nous intéressons à la sélection dynamique de données. L'objectif est de développer des méthodes capables d'identifier les sources pertinentes de données parmi un ensemble de flux (e.g., identifier les auteurs de contenus à écouter), afin de réaliser une collecte répondant à divers critères pré-définis (e.g., besoins d'information particuliers, communautés cibles, caractéristiques relationnelles des données collectées, etc.). Nous montrons comment ce genre de tâche de sélection en temps réel peut être traitée par divers types d'algorithmes de bandits multi-bras ;
- Le second chapitre se focalise sur la diffusion d'évènements discrets sur les réseaux. L'objectif est de modéliser la propagation de contenus (e.g., url, mème, image, etc.) sur le réseau, pour diverses tâches d'analyse et prédiction, à une échelle microscopique (i.e., à l'échelle des individus ou groupes d'individus). Divers modèles, notamment neuronaux, sont proposés pour répondre à cette problématique, dans un cadre d'apprentissage avec données manquantes.
- Dans le troisième chapitre, nous étendons ce travail de modélisation à des phénomènes de propagation plus diffus sur les réseaux, à savoir les évolutions langagières au sein d'une communauté d'auteurs. L'idée est de capturer les évolutions de la langue sur un groupe d'individus au fil du temps, afin de faire émerger les tendances globales et jeux d'influence dans la communauté, voire d'être à même d'identifier des leaders d'opinion, sans s'appuyer sur des séquences d'évènements binaires comme c'est le cas classiquement dans les travaux d'analyse de diffusion.

Un chapitre de conclusion vient finalement faire le bilan du travail réalisé, et décrit divers travaux annexes développés dans le cadre de mes recherches. Il énonce également un certain nombre de perspectives qui constituent mon projet de recherche pour les années à venir, qui vise à unifier sélection de données et inférence structurelle dans un cadre commun, basé notamment sur de l'apprentissage par renforcement, pour l'extraction "dynamique" de dynamiques de l'information.



## Chapitre 2

# Capture de données sur les réseaux

### Sommaire

---

<b>2.1 Collecte temps réel ciblée</b> . . . . .	<b>8</b>
2.1.1 Sélection dynamique de flux . . . . .	9
2.1.2 Algorithmes de bandits pour la collecte de données dynamique . . . . .	13
<b>2.2 Bandits contextuels pour la sélection active de flux de données</b> . . . . .	<b>19</b>
2.2.1 Algorithmes de bandits contextuels . . . . .	19
2.2.2 Collecte structurée avec profils cachés . . . . .	24
2.2.2.1 Bandit contextuel sur profils cachés . . . . .	25
2.2.2.2 Intervalles de confiance pour profils cachés . . . . .	26
2.2.2.3 Algorithme SampLinUCB . . . . .	30
2.2.2.4 Application à la collecte dynamique de données . . . . .	31
2.2.3 Collecte dynamique contextuelle . . . . .	33
2.2.3.1 Modèle génératif et inférence variationnelle . . . . .	35
2.2.3.2 Algorithme <i>HiddenLinUCB</i> . . . . .	40
2.2.4 Collecte dynamique relationnelle . . . . .	43
2.2.4.1 Modèle relationnel . . . . .	44
2.2.4.2 Modèle à espace latent . . . . .	46
<b>2.3 Conclusions et Perspectives</b> . . . . .	<b>49</b>
2.3.1 Inférence Bayésienne . . . . .	49
2.3.2 Réseaux de neurones probabilistes . . . . .	51
2.3.3 Récompenses dynamiques . . . . .	53

---

Face à l'augmentation croissante des données produites sur le web et les media sociaux, d'importants efforts ont été déployés pour établir des méthodes efficaces pour les exploiter. Alors que des progrès considérables ont été réalisés dans le domaine du stockage, de la parallélisation et du calcul distribué, des méthodes d'apprentissage en ligne sont nécessaires pour intégrer en temps réel ces flux continus de données, potentiellement illimités, dans les modèles. Diverses techniques ont été proposées dans la littérature pour apprendre sans avoir à stocker - ou peu. Dans le cadre du web, le rythme de production de certaines sources de données est tel qu'il n'est en effet clairement pas possible de tout stocker pour intégration future à un modèle d'apprentissage hors-ligne. Néanmoins, le cadre d'apprentissage hors-ligne offre un certain nombre d'avantages par rapport à sa contre-partie en-ligne, notamment pour l'extraction de relations entre entités productrices d'informations, où il est généralement plus simple d'identifier des influences ou corrélations de comportements avec l'ensemble des données d'apprentissage à disposition (les modèles des chapitres suivants ne sont par exemple pas directement applicables dans un cadre d'apprentissage en ligne, bien que cela constitue une perspective de recherche importante). Être à même de réaliser une collecte dynamique ciblée apparaît alors comme un enjeu crucial, afin de permettre l'établissement de jeux de données restreints, répondant à divers critères importants pour la tâche de modélisation ciblée (e.g., ciblage d'une communauté ou d'une thématique spécifique, densité du graphe relationnel des données, diversité des contenus, profils des auteurs, etc.). En outre, l'analyse de données en temps réel est requise pour de nombreuses applications, telles que le traitement de nouvelles journalistiques, la veille stratégique, la surveillance d'entités sensibles, ou pour n'importe quelle autre tâche nécessitant la capture de dynamiques sociales en continu. Pour toutes ces tâches, il est nécessaire de pouvoir s'adapter aux changements structurels et thématiques des réseaux, afin de pouvoir réagir en conséquence.

Ce chapitre traite de la sélection de données en continu à partir des media sociaux, qui constitue un réel challenge dans divers contextes, car il s'agit d'orienter dynamiquement l'attention de l'agent de capture sur les données importantes pour la tâche ciblée. Les flux de données considérés sont massifs, continus et potentiellement illimités. De plus, les distributions des données qu'ils émettent ne sont pas nécessairement stationnaires. Nous parcourons diverses réponses, apportées notamment au cours de la thèse de Thibault Gisselbrecht pour la collecte de données à partir de flux, de la présentation de la tâche et d'une première approche pour traiter le problème en section 2.1, à des méthodes plus évoluées visant à prendre en considération la structure stationnaire des flux en section 2.2.2, les changements de distributions en section 2.2.3 et les relations entre flux de données en section 2.2.4. Nous concluons par une ouverture à diverses perspectives de recherche autour de ces problématiques de collecte d'informations en section 2.3.

## 2.1 Collecte temps réel ciblée

Bien que certains media offrent la possibilité de parcourir leurs bases de données historiques, cela implique souvent un coût très important (i.e., abonnement très onéreux sur des plateformes telles que Twitter) et difficile à maîtriser pour les analyseurs (i.e., difficulté pour quantifier à priori la quantité de calculs nécessaires pour la collecte des données cibles, les abonnements étant souvent définis en terme d'unités de temps de calcul). D'autre part, comme énoncé plus haut, pour diverses applications, capturer l'information au moment de sa publication est primordial pour apporter des réponses en temps réel. Une alternative à l'exploitation de bases de données statiques est rendue possible pour la plupart des media sociaux, par la mise à disposition d'APIs de diffusion en continu des publications postées sur le media, afin de permettre à des applications tierces d'analyser les activités récentes de

leurs utilisateurs (à des fins publicitaires par exemple).

Cette section introductive commence par présenter la tâche générale de collecte à partir de flux traitée dans ce chapitre, en la positionnant par rapport à la littérature connexe du domaine. Nous présentons ensuite une première contribution, basée sur des algorithmes de bandits multi-bras, que nous avons proposée dans [Gisselbrecht et al., 2015c] pour traiter le problème posé (fin de section 2.1.2).

### 2.1.1 Sélection dynamique de flux

Comme le fait justement remarquer [Bechini et al., 2016], la collecte de données à partir des media sociaux est souvent seulement vue comme un problème annexe, préliminaire à l'établissement de modèles, plutôt qu'un processus important devant faire partie intégrante de la réflexion pour le développement d'approches pour la fouille et le traitement de données sociales. Cependant, la difficile tâche de capture temps-réel à partir des media sociaux, avec les diverses contraintes associées, n'a été que peu explorée dans la littérature.

**Fouille de données sociales** Une large part des travaux en fouille de données sociales a d'abord considéré les réseaux sociaux de la même manière que le graphe du Web, avec les domaines du Web remplacés par les utilisateurs des réseaux, et les hyperliens par les connexions sociales connues entre ces utilisateurs. Suivant ce principe, il est possible d'appliquer aux graphes des réseaux sociaux des méthodes de *Web Crawling*, classiquement utilisées pour définir des stratégies de sélection de nœuds à visiter pour l'indexation de pages Web. Le *Crawling* ciblé, initialement introduit dans [Chakrabarti et al., 1999], détermine si un nœud d'un réseau devrait être visité ou non selon la pertinence de son contenu pour un besoin prédéfini. De nombreuses variations autour de ce modèle ont été proposées, telles que [Micarelli and Gasparetti, 2007] qui définit des stratégies adaptatives en fonction du voisinage des nœuds dans le graphe, pouvant être appliquées pour identifier des utilisateurs comme sources d'information utile dans les réseaux. Plus spécifiquement pour le cadre des réseaux sociaux, *Twittomender* [Hannon et al., 2010] utilise des méthodes de filtrage collaboratif (technique de factorisation matricielle classiquement utilisée par les systèmes de recommandation) pour identifier les utilisateurs à suivre en priorité, alors que *Who to Follow* [Gupta et al., 2013] sélectionne les sources à considérer par une marche aléatoire sur le graphe. Cependant, ce genre d'approches nécessite la connaissance globale du réseau considéré, ce qui n'est habituellement pas le cas dans des situations réelles, les relations entre utilisateurs n'étant souvent pas connues a priori. Or, alors que les liens du Web peuvent être découverts au fur et à mesure de l'indexation des pages sans limitation particulière, les réseaux sociaux ne mettent souvent à disposition qu'une portion restreinte du graphe social à travers les APIs proposées. Face à cette contrainte, [Boanjak et al., 2012] a proposé un système distribué, permettant d'interroger l'API *Twitter* à partir de multiples IPs, afin de contourner les restrictions empêchant le *Crawling* ciblé sur ce réseau. Notons que des approches similaires [Catanese et al., 2011] et [Gjoka et al., 2010] ont également été imaginées pour Facebook. Néanmoins, les politiques fixées par ces media sociaux étant de plus en plus restrictives (en terme de fréquence de requêtes autorisées), de telles approches ne paraissent plus possibles aujourd'hui (d'autant que la taille des graphes a explosé dans le même temps). Sur *Twitter* par exemple, un maximum de 180 requêtes par 15 minutes est autorisé. Cela paraît clairement insuffisant pour collecter la totalité des liens du réseau comptant plus 300 millions d'utilisateurs. Et même de manière distribuée : la capture de la totalité de l'activité sur *Twitter* nécessiterait presque 2 millions de clients, chacun avec une adresse IP et une clé d'authentification différents.

**Fouille de flux de données** L'exploitation des flux de données fournis par les media sociaux constitue une alternative intéressante à ces parcours de graphes complexes, qui se révèlent impossibles en pratique dans le cadre des réseaux sociaux de grande taille. La fouille de flux de données (*Data Stream Mining*) s'intéresse à l'extraction de connaissances à partir de flux continus, potentiellement infinis et fournissant des données à haute fréquence. Ce champ de recherche implique le développement d'approches d'apprentissage en ligne, qui ne requièrent donc pas le stockage de la totalité des données, afin de contrôler la complexité des méthodes. De plus, les données issues de ces flux ne sont pas nécessairement stationnaires, ce qui implique de devoir prendre en considération de possibles dérives conceptuelles (*Concept Drift*) dans les modèles. La fouille de flux de données a donné lieu à de nombreux algorithmes, notamment des approches reposant sur un échantillonnage probabiliste des données, tels que les algorithmes VFDT [Domingos and Hulten, 2000] et CVFDT [Zhong et al., 2013] basés sur des arbres de décision de Hoeffding (*Hoeffding Trees* [Hulten et al., 2001, Kirkby, 2007]). Les arbres de Hoeffding, qui utilisent l'inégalité de Hoeffding pour déterminer statistiquement quand séparer les feuilles d'un arbre de décision pour en raffiner la classification à partir d'un sous-ensemble de données collectées, sont un exemple populaire de modèle de décision incrémental et perpétuel, capable d'apprendre à partir de flux de données massifs, mais qui ne permettent pas de considérer la non-stationnarité éventuelle des distributions. D'autres approches, dites de *Sketching*, résument les données du flux via des projections dans des espaces de dimension réduite. Alors que les approches de *Sketching data-oblivious* considèrent des projections fixes aléatoires ou apprises a priori (e.g., LSH [Indyk and Motwani, 1998], MDSH [Weiss et al., 2012] ou encore KLSH [Kulis and Grauman, 2009]), les approches *data-dependent* considèrent des projections qui évoluent en fonction des données entrantes (e.g., PCA [Kargupta et al., 2004], *Laplacian Eigenmaps* [Belkin and Niyogi, 2003] ou *Spherical Hashing* [Lee, 2012]). Enfin, citons des approches à base de fenêtre glissante (ne conservant simplement que les données les plus récentes) [Datar et al., 2002], d'agrégation de données (s'appuyant sur des statistiques suffisantes des distributions des données) [Aggarwal et al., 2003, Aggarwal et al., 2004] ou encore à base d'ondelettes [Papadimitriou et al., 2003]. Toutes ces approches constituent des solutions pour résoudre des tâches d'apprentissage classique (régression, classification, etc.) pour des contextes où donc les exemples arrivent au fil de l'eau, de façon potentiellement rapide, et sans pouvoir être réutilisés. Certains travaux étendent ces approches à la prise en compte de flux de données simultanés, comme par exemple [Hesabi et al., 2015] qui propose une méthode de *clustering* incrémental pour des données issues d'un grand nombre de flux. Dans ce cadre à flux multiples, d'autres approches considèrent les flux comme des séries temporelles, sur lesquelles divers modèles auto-régressifs peuvent être appliqués (modèles statistiques AR, ARMA, ARIMA, etc. ou modèles neuronaux récurrents RNNs en ligne), éventuellement avec apprentissage de corrélations entre séries [DeLasalles et al., 2019c].

**Contexte de travail** Concernant les réseaux sociaux, une très large littérature s'est concentrée au début des années 2000 sur le domaine du *Topic Detection and Tracking* (TDT), introduit dans [Alla et al., 1998], et qui inclut des problématiques de segmentation de flux, de détection d'événements à partir de flux, et de suivi de contenus, possiblement évolutifs, dans les flux. Les travaux de ce domaine reprennent efficacement les principes de la fouille de flux de données classique, en appliquant les approches énoncées ci-dessus, comme par exemple [Guralnik and Srivastava, 1999] pour la détection d'événements dans des flux de données sociales. Cependant, les travaux en TDT considèrent généralement que les flux sont entièrement accessibles, ce qui n'est pas le cas pour la majorité de APIs des media sociaux. Au delà des difficultés techniques qu'engendrerait la prise en compte complète de l'ensemble des flux (e.g., jusqu'à 7000 messages à analyser par seconde sur Twitter), les media sociaux

fixent également des restrictions sur l'utilisation de leurs APIs de *Streaming*, en limitant la délivrance de contenus à un certain ratio maximal de leur activité courante. Par exemple, Twitter<sup>1</sup> offre les 3 APIs de streaming suivantes :

- *Sample Streaming* : fournit des échantillons de messages en temps-réel, correspondant à 1% de l'ensemble des messages postés chaque seconde sur le réseau;
- *Track Streaming* : donne un accès temps-réel à tout message posté qui contient au moins un mot parmi une liste d'au plus 400 mots-clés à définir, dans la limite de 1% de l'activité globale du réseau;
- *Follow Streaming* : permet de suivre l'activité complète d'une liste d'au plus 5000 utilisateurs, dans la limite de 1% de l'activité globale du réseau;

Quelle que soit le mode d'échantillonnage utilisé, de fortes restrictions sont donc appliquées. Pour une très large majorité des travaux exploitant les flux issus des media sociaux, tels que ceux présentés dans [Cataldi et al., 2010], ces contraintes ne posent pas de problème particulier car ils peuvent se contenter d'un accès aux ressources tel que celui fourni par l'API *Sample Streaming* de Twitter. Par exemple, la détection des contenus les plus populaires du moment peut se faire relativement efficacement à partir de seulement 1% de l'activité globale du réseau. Pour des tâches d'analyse de sentiments, l'approche de [Bifet and Frank, 2010] propose également d'exploiter ce flux aléatoire, afin d'apprendre des classificateurs d'opinions basés sur des arbres de Hoeffding. Ce genre d'approche a notamment été appliqué dans le contexte de la campagne présidentielle américaine de 2012 [Wang et al., 2012b]. Cependant, aucun de ces travaux ne se préoccupe de la sélection dynamique de flux dans un environnement contraint comme celui des réseaux sociaux. Cependant, il paraît évident que le flux fourni par l'API *Sample Streaming* n'est pas suffisant pour de nombreuses tâches, notamment des tâches de capture qui nous intéressent, étant données ses limitations par rapport à l'activité globale du trafic. En particulier, son utilisation n'est pas bien adaptée pour un besoin d'information spécifique, avec une très large proportion de contenus non pertinents délivrés, et une forte probabilité de passer à côté de l'information utile.

Les deux autres APIs décrites ci-dessus impliquent de cibler les flux de données pertinents, en sélectionnant les mots-clés ou utilisateurs à suivre. Le travail présenté dans [Li et al., 2013] a par exemple proposé d'exploiter l'API *Track Streaming* de Twitter pour collecter les messages liés aux mots-clés en vogue (*trending words*) sur le réseau. Cependant, cette approche ne permet pas une exploration efficace du réseau pour des tâches de collecte avec besoin prédéfini, ne s'appuyant sur aucun mécanisme d'apprentissage pour s'orienter vers les mots-clés utiles : l'ensemble des mots-clés à suivre est complètement redéfini à chaque nouvelle période d'écoute, en sélectionnant les mots les plus utilisés dans des messages collectés simultanément à partir de l'API *Sample Streaming*. D'autre part, le type d'accès aux ressources fourni par l'API *Track Streaming* de Twitter limite le processus de collecte à des besoins de données pouvant s'exprimer par le biais d'une liste de mots-clés. Alors qu'une exploration dynamique utilisant l'API *Follow Streaming* peut permettre d'identifier les utilisateurs du réseau les plus pertinents pour le besoin considéré, une exploration dynamique basée sur des mots-clés ne peut que rester cantonnée à l'identification des termes les plus fréquents dans les messages pertinents, ce qui paraît bien plus limité. Par exemple, pour des tâches où l'on est intéressé par des données fortement connectées (i.e. avec graphe de relations dense), centrées sur une communauté d'utilisateurs particulière, des modes d'échantillonnages comme celui proposé par l'API *Track Streaming* de Twitter ne

---

1. Dans nos travaux sur la collecte de données, nous nous sommes focalisés sur le media *Twitter*, mais bien d'autres media pourraient être considérés, tels que par exemple *Facebook* qui propose des APIs de Streaming avec restrictions similaires. La documentation complète des APIs Twitter est disponible à l'adresse : <https://dev.twitter.com/streaming/public>

paraissent clairement pas adaptés. Par ailleurs, dans le cas spécifique de Twitter, il est probable que certains mots-clés trop génériques viennent saturer la collecte en atteignant à eux seuls la limite des 1% de l'activité globale du réseau, menant alors à ignorer les messages correspondant à d'autres mots ciblés, pourtant plus discriminants. Ce genre de difficulté est beaucoup moins probable via des collectes selon l'API *Follow Streaming*, aucun utilisateur ne produisant suffisamment de messages sur le réseau pour saturer la collecte.

**Un problème de sélection de capteurs** Dans ce chapitre, nous nous intéressons aux producteurs de contenu, les utilisateurs du réseau, dont le suivi ouvre sur un plus large spectre d'objectifs que le suivi de mots-clés dans les messages publiés. Nous exploitons donc des modes d'accès aux ressources du type de l'API *Follow Streaming* de *Twitter*, pour proposer un processus de capture générique, capable de se concentrer sur les sources de messages pertinents pour un besoin particulier. Une difficulté majeure pour notre tâche de collecte à partir de flux centrés sur des utilisateurs du réseau réside dans le fait que seuls les messages postés par ces utilisateurs ciblés sont observés. Une large part de l'information est alors cachée pour l'agent de collecte. Il est alors crucial de définir des stratégies d'exploration efficaces, pour maximiser la quantité de contenus utiles collectés en re-spécifiant les utilisateurs à suivre pour chaque nouvelle période d'écoute. Un problème connexe est celui de la sélection de capteurs, pour lequel l'objectif est de définir des sous-ensembles de  $k$  capteurs parmi  $N$ , de manière à estimer efficacement diverses mesures sur un environnement, sous contraintes de budget limité [Joshi and Boyd, 2009]. Par exemple, le déploiement de caméras de surveillance peut s'apparenter à un problème de sélection de capteurs où le but est d'obtenir une bonne couverture des activités sensibles, tout en minimisant le nombre de caméras requises [Spaan and Lima, 2009]. Ce problème est particulièrement complexe dans des environnements dynamiques, où les objets mesurés évoluent au cours du temps, comme c'est le cas sur les réseaux sociaux. Dans [Colbaugh and Glass, 2011], le but est d'identifier les blogs d'un réseau qui re-publient la plupart des contenus les plus populaires. L'approche cherche ainsi à identifier les sources de données utiles, mais cela est réalisé de manière statique, selon des modèles de diffusion appris hors-ligne à partir de données collectées sur l'ensemble des nœuds du réseau. Son applicabilité est alors limitée à des réseaux de petite taille, pour lesquelles un volume de données suffisant peut être collecté pour chaque nœud, et pour des distributions stationnaires de diffusion de l'information. Des approches de modélisation de processus de diffusion de l'information en ligne ont été proposées [Taxidou, 2013, Taxidou and Fischer, 2014b], mais requièrent la connaissance du graphe de relations entre utilisateurs, à l'instar des méthodes issues du Web décrites ci-dessus. Dans [De Choudhury et al., 2010], les auteurs proposent diverses stratégies d'échantillonnage des utilisateurs à suivre pour mesurer la diffusion de contenus sur le réseau, mais ne proposent pas de mécanisme d'apprentissage de sélection de ces utilisateurs. Enfin, les travaux présentés dans [Bao et al., 2015] développent une approche de couverture d'ensemble (*Set Cover problem*), dans laquelle l'objectif est de déterminer les  $k$  utilisateurs à suivre pour obtenir la meilleure couverture des différents sujets abordés sur le réseau, en supposant que l'écoute d'un utilisateur permette de récupérer les contenus postés par tous ses contacts (ce qui n'est pas le cas pour la plupart des réseaux). Là encore, l'approche requiert la connaissance du réseau, ce qui est le cas de la plupart des approches de sélection de capteurs (que ce soient des relations de localisation géographiques ou des relations sociales dans le cas des media sociaux).

**Formalisation de la tâche** Nous considérons donc un problème de collecte dynamique à partir de flux centrés sur des utilisateurs du réseau, où chaque flux fournit en temps réel les messages postés

par l'utilisateur sur lequel il est positionné, sous contraintes restrictives sur le nombre d'utilisateurs qui peuvent être suivis simultanément. L'objectif est de collecter des données utiles pour un besoin identifié. Puisque aucune connaissance n'est fournie a priori, ce problème requiert de faire évoluer dynamiquement l'ensemble d'utilisateurs suivis à chaque période, afin d'explorer l'environnement et permettre de s'orienter vers les sources de données les plus susceptibles de produire du contenu utile sur la prochaine période d'écoute. Cette problématique générale nous est apparue en discutant avec des industriels nous affirmant réaliser cette tâche manuellement pour divers clients ayant des objectifs divers (l'un des clients avait par exemple des problématiques d'estimation de son audimat). Ainsi, les industriels en question réorientaient au moins quotidiennement, pour chaque client, ses écoutes des réseaux sociaux en modifiant la liste des utilisateurs suivis pour la journée via l'API *Follow Streaming*, ce qui correspond à un travail colossal et relativement peu efficace. Cette tâche peut en effet difficilement être réalisée manuellement, pour deux raisons principales. Premièrement, les critères choisis pour sélectionner les utilisateurs à écouter peuvent être durs à définir manuellement, sans connaissance de la diversité des messages postés par les utilisateurs du réseau. Deuxièmement, la quantité de données à analyser est trop importante et des ré-orientations fréquentes ne sont pas envisageables, à moins d'impliquer un coût humain très important. Étant donnée une fonction de récompense qui nous permet d'évaluer l'utilité des messages collectés par rapport à un besoin de données, notre idée a alors été de réfléchir à une politique efficace de ré-orientation automatique des utilisateurs suivis. Cette politique, qui doit être apprise en ligne, vise à maximiser un gain cumulé au cours du temps de collecte, selon les scores attribués par la fonction de récompense aux messages publiés par l'ensemble des utilisateurs suivis à chaque période d'écoute du processus. Ce genre d'approche aurait l'avantage d'être suffisamment générique pour s'appliquer pour n'importe quel besoin en données, sous réserve que ce besoin puisse être exprimé par une fonction associant une récompense à chaque message collecté. Diverses fonctions de récompense peuvent être imaginées, pour des besoins différents tels que capturer des contenus d'actualités sur des thèmes spécifiques, identifier des leaders d'opinion ou encore collecter des messages qui tendent à satisfaire un panel de clients donné. Notons que, alors que les travaux connexes présentés dans [Hannon et al., 2010] ou [Gupta et al., 2013] s'intéressent à la précision des messages collectés (taux de messages collectés pertinents), nous nous intéressons plutôt à la maximisation du volume de données pertinentes collectées, dans une perspective de collecte automatique. Formellement, selon un ensemble  $\mathcal{K}$  de  $K$  utilisateurs du réseau, nous considérons le problème d'optimisation suivant :

$$\max_{(\mathcal{K}_t \subset \mathcal{K})_{t=1..T}} \sum_{t=1}^T \sum_{i \in \mathcal{K}_t} r_{i,t} \quad (2.1)$$

avec  $T \in \mathbb{N}$  le nombre de périodes de collecte,  $\mathcal{K}_t \subset \mathcal{K}$  l'ensemble des  $k$  utilisateurs ( $k < K$ ) suivis pendant la période  $t$  et  $r_{i,t} \in \mathbb{R}$  la somme des récompenses associées aux contenus postés par l'utilisateur  $i$  pendant cette période. Dans nos expérimentations, nous nous sommes toujours restreints à des fonctions de récompense stationnaires, où un même message obtient toujours le même score quel que soit l'avancement de la collecte, mais des fonctions évolutives pourraient être envisagées, afin de permettre par exemple de donner plus de crédit à des contenus nouveaux, ou de favoriser la collecte d'un graphe de données plus ou moins dense (voire les perspectives à ce sujet en section 2.3).

### 2.1.2 Algorithmes de bandits pour la collecte de données dynamique

Puisque pour notre tâche il est question de définir un agent maximisant une récompense cumulée, sans que les décisions successives ne modifient les distributions de récompenses associées aux

différents utilisateurs candidats, l'utilisation d'algorithmes de bandits multi-bras paraît naturelle. Les bandits multi-bras [Bubeck and Cesa-Bianchi, 2012], originellement introduits dans [Lai and Robbins, 1985], réfèrent à une classe de problèmes d'exploitation/exploration dans les processus de décision séquentiels où, à chaque étape de décision, un agent doit choisir une action parmi un ensemble de  $K$  actions candidates, avec pour but de maximiser la somme des récompenses récoltées tout au long du processus. Ces problèmes entrent dans le cadre plus général de l'apprentissage par renforcement, qui manipule des processus de décision markoviens (MDP), mais présentent la spécificité de travailler sur des MDP à un seul état (i.e., les décisions de l'agent ne modifient pas l'environnement). Dans l'instance traditionnelle du bandit stochastique, chaque action (appelée bras dans le cadre des bandits) est associée à une distribution stationnaire de récompenses. Les récompenses de chaque action  $i \in \mathcal{K}$  sont alors supposées être échantillonnées indépendamment et identiquement, selon une distribution de moyenne  $\mu_i$ . L'objectif est alors de minimiser le pseudo-regret cumulé défini tel que :

$$\hat{R}_T = T\mu_{i^*} - \sum_{t=1}^T \mu_{i_t} \quad (2.2)$$

où  $i^*$  correspond au bras de meilleure espérance de récompense et  $i_t$  correspond au bras actionné à l'étape  $t$ .

**Algorithmes de bandits** L'approche la plus simple pour traiter ce genre de problème est certainement l'algorithme  $\epsilon$ -greedy [Auer et al., 2002], qui exploite l'action de meilleure moyenne estimée avec une probabilité de  $1 - \epsilon$  et explore une autre action tirée uniformément parmi les candidates avec une probabilité de  $\epsilon$ , où le paramètre  $\epsilon$  peut possiblement décroître avec le temps (décroissance de  $\epsilon$  nécessaire pour assurer la convergence de l'algorithme, mais sa planification est difficile dans le cadre général). Une autre famille d'algorithmes pour traiter les problèmes de bandits est celle des algorithmes de type UCB (*Upper Confidence Bound*), qui considèrent l'intervalle de confiance des estimateurs d'espérance de récompense des différents bras. Ces algorithmes sont dits optimistes car ils travaillent en faisant l'hypothèse que l'espérance de chaque bras se situe au niveau de la borne supérieure de l'intervalle de confiance de son estimateur empirique. Choisir à chaque étape le bras de meilleur score UCB permet de garantir un pseudo-regret sous linéaire. De nombreuses extensions de l'algorithme UCB initial [Auer et al., 2002] ont été proposées pour le bandit stochastique, utilisant d'autres inégalités de concentration que l'inégalité de Hoeffding utilisée dans [Auer et al., 2002], ou d'autres manière d'exploiter les intervalles de confiance pour assurer le compromis exploitation/exploration (voir par exemple UCBV dans [Audibert et al., 2009], MOSS dans [Audibert and Bubeck, 2009] ou encore KL-UCB dans [Cappé et al., 2013]). Comme nous le verrons dans les prochaines sections, des extensions contextuelles de ces algorithmes ont également été proposées, tels que le fameux LinUCB [Chu et al., 2011], permettant de prendre en compte des observations sur le contexte de décision et/ou des informations sur la structure des bras. Une autre grande famille d'approches pour traiter le problème du bandit stochastique s'appuie sur l'algorithme Thompson Sampling, initialement proposé dans [Thompson, 1933]. Plutôt que de choisir de manière déterministe le bras à actionner en fonction de la borne supérieure de l'intervalle de confiance de son estimateur, l'idée est d'échantillonner les paramètres de la distribution de ses récompenses en fonction de leur distribution postérieure. Cette approche bayésienne permet elle aussi de garantir un pseudo-regret sous linéaire (voir [Kaufmann et al., 2012] et [Agrawal and Goyal, 2012]). Notons que des versions contextuelles ont également été développées dans le cadre des algorithmes de Thompson Sampling [Chapelle and Li, 2011, Agrawal and Goyal, 2013].

**Bandits sur le Web** Les algorithmes de bandits ont déjà été très largement utilisés pour des tâches liées au Web et aux réseaux sociaux, à commencer par la publicité/recommandation de produits en ligne [Kohli et al., 2013, Guillou et al., 2016, Nuara et al., 2018], qui constitue leur principale utilisation à large échelle par les industriels. Pour cette application, on note par exemple l’approche *Gang of Bandit* présentée dans [Cesa-Bianchi et al., 2013], qui émet l’hypothèse que des utilisateurs connectés ont tendance à avoir des comportements similaires (i.e., homophilie), et exploite le graphe des relations connues entre utilisateurs pour fournir des recommandations efficaces. Cette idée est reprise plus tard dans [Gentile et al., 2014], qui fait une hypothèse similaire mais où les dépendances entre bras sont détectées automatiquement par clustering, plutôt que de s’appuyer sur un graphe de relations (pas toujours) connu. D’autres applications concernent par exemple la maximisation d’audience dans [Lage et al., 2013], où l’algorithme *Thompson Sampling* est employé pour sélectionner quels messages un utilisateur devrait publier pour maximiser son impact sur le réseau. Dans [Bnaya et al., 2013], une approche de bandit générique est proposée pour réaliser du *crawling* ciblé dans les réseaux sociaux avec graphe de relations connus. Dans [Vaswani and Lakshmanan, 2015], les auteurs traitent une tâche de maximisation d’influence via une approche de bandit stochastique. L’algorithme, basé sur le modèle de diffusion *Independent Cascade* (IC, voir section 3.1), cherche à découvrir les utilisateurs les plus influents du réseau, en s’appuyant sur le graphe des relations sociales mais sans connaissance des probabilités d’influence a priori. Pour un objectif similaire, mais sans connaissance du graphe de relations, [Carpentier and Valko, 2016] propose d’employer un algorithme de bandit relationnel sur un sous-graphe d’utilisateurs extrait au cours d’une phase d’exploration pure, afin d’identifier les nœuds les plus influents. Tous ces exemples d’application montrent la bonne aptitude des approches de bandit pour traiter des tâches en ligne sur les réseaux sociaux, pour lesquelles pas (ou peu) d’information n’est disponible sur l’environnement a priori. Dans ce chapitre, nous explorons leur application pour notre problème de capture de données dynamique.

**Application pour la collecte à partir de flux** Dans notre approche préliminaire présentée dans [Gisselbrecht et al., 2015c], nous formalisons notre problème de capture sous la forme d’un problème de bandit stochastique combinatoire [Chen et al., 2013, Combes et al., 2015], dans lequel l’agent doit choisir un ensemble  $\mathcal{K}_t \subset \mathcal{K}$  de  $k > 1$  bras à actionner à chaque pas de temps  $t$ . En considérant que ces bras à actionner correspondent chacun à un utilisateur du réseau à suivre pendant la prochaine période de temps (entre  $t$  et  $t + 1$ ), on cherche alors la politique  $\pi$  qui maximise la formulation du problème donné par l’équation 2.1, avec  $\mathcal{K}_t$  déterminé par  $\pi$ . Dans notre problème, les récompenses obtenues dépendent des messages collectés pour chaque individu, nous avons donc accès aux récompenses individuelles (et pas uniquement à la récompense commune associée au super-bras  $\mathcal{K}_t$ ). Cela nous place dans le cadre des semi-bandits combinatoires, selon la définition donnée dans [Audibert et al., 2014], un problème de bandit combinatoire strict n’ayant accès qu’à la récompense commune associée à  $\mathcal{K}_t$ . L’algorithme CUCB [Chen et al., 2013] propose une extension de l’algorithme de bandit stochastique UCB aux problèmes à sélection multiple pour ce cadre de semi-bandit. Dans notre cas, selon l’équation 2.1, nous restreignons l’aspect combinatoire à une simple somme des récompenses individuelles, mais d’autres fonctions de récompenses combinatoires seraient envisageables en s’appuyant sur CUCB, qui reste valide pour toute récompense commune correspondant à une fonction lipschitzienne des  $k$  récompenses individuelles des bras actionnés.

Pour notre tâche de capture dynamique, une difficulté majeure réside dans le fait que nous ne connaissons pas *a priori* l’ensemble complet des utilisateurs du réseau. Ainsi, en partant d’un ensemble d’utilisateurs initiaux, il s’agit de concevoir un processus alimentant l’ensemble des sources possibles, de façon incrémentale au fur et à mesure que de nouveaux utilisateurs sont rencontrés,

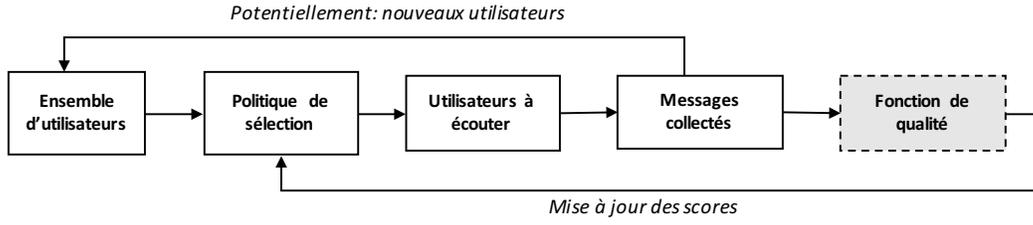


FIGURE 2.1 – Capture dynamique de données sur les réseaux sociaux.

afin d’orienter la collecte vers des zones non connues du réseau. Par exemple, on peut s’appuyer sur des mentions des utilisateurs écoutés à d’autres utilisateurs du réseau dans les messages collectés. Typiquement, sur Twitter, les utilisateurs peuvent se répondre entre eux ou republier des messages d’autres utilisateurs, ce qui nous offre la possibilité de découvrir de nouvelles sources sans utilisation de ressources externes. Une autre option serait d’utiliser l’API *Sample Streaming* en parallèle pour alimenter le pool d’utilisateurs  $\mathcal{K}$  à considérer. Le processus général, présenté dans la figure 2.1 peut être décrit de la façon suivante :

1. Sélection d’un sous ensemble d’utilisateurs  $\mathcal{K}_t \subset \mathcal{K}$  relativement à la politique courante  $\pi$ ;
2. Écoute de ces utilisateurs sources pendant une période de temps  $[t, t + 1]$ , avec chaque pas de temps  $t$  séparé d’un délai  $\Delta_t$  du pas précédent  $t - 1$ ;
3. Alimentation de l’ensemble des sources disponibles  $\mathcal{K}$  en fonction des nouveaux utilisateurs référencés dans les messages enregistrés;
4. Évaluation des données collectées en fonction de leur pertinence pour la tâche à résoudre;
5. Mise à jour de la politique de sélection  $\pi$  en fonction des scores  $r_{i,t}$  obtenus pour chaque utilisateur  $i \in \mathcal{K}_t$  pendant la période d’écoute  $[t, t + 1]$ .

Le fait que tous les utilisateurs ne sont pas connus à l’initialisation place notre problème dans le cadre du *sleeping bandit* [Kleinberg et al., 2008], dans lequel l’ensemble des actions disponibles change d’une itération à l’autre. Pour ce cadre, il est possible d’appliquer des algorithmes de bandit stationnaire classiques à la différence près que chaque action doit être actionnée à sa première apparition afin d’initialiser ses estimateurs. Selon les principes des approches type UCB, et l’algorithme CUCB sur lequel nous nous appuyons, nous travaillons de manière gloutonne avec une politique qui choisit à chaque étape  $t$  l’ensemble des  $k$  utilisateurs maximisant un score  $s_{i,t}$  défini pour chaque utilisateur  $i \in \mathcal{K}$  à l’instant  $t$  :

$$s_{i,t} = \begin{cases} \hat{\mu}_{i,t-1} + B_{i,t} & \text{si } N_{i,t-1} > 0 \\ +\infty & \text{si } N_{i,t-1} = 0 \end{cases} \quad (2.3)$$

avec  $N_{i,t-1} = \sum_{s=1}^{t-1} \mathbb{1}_{\{i \in \mathcal{K}_s\}}$  le nombre de fois où l’utilisateur  $i$  a été sélectionné avant l’étape  $t$ ,  $\hat{\mu}_{i,t-1} = \frac{1}{N_{i,t-1}} \sum_{s=1}^{t-1} \mathbb{1}_{\{i_s=i\}} r_{i,s}$  la moyenne empirique de l’action  $i$  et  $B_{i,t} = \sqrt{\frac{2 \log(t)}{N_{i,t-1}}}$  le terme d’exploration classique dans les algorithmes UCB. Le score  $s_{i,t}$  correspond à la borne supérieure de l’intervalle de confiance pour l’utilisateur à l’étape  $t$ , selon l’inégalité de concentration de Hoeffding en fonction de l’estimateur empirique  $\hat{\mu}_{i,t-1}$ . Il représente bien un compromis exploitation/exploration, avec un premier terme correspondant à l’estimation courante de l’utilité de l’utilisateur  $i$  (exploitation) et un second terme décroissant avec le nombre de fois où l’utilisateur  $i$  est sélectionné (exploration). Le

score des utilisateurs non écoutés au moins une fois (i.e.,  $N_{i,t-1} = 0$ ) est fixé à  $+\infty$  de manière à forcer le système à les sélectionner pour initialiser leurs estimations avec une première écoute (tant qu'une limite sur  $|\mathcal{K}|$  n'est pas atteinte). Le processus de collecte générique proposé est détaillé dans l'algorithme 1, où  $\omega_{i,t}$  correspond à la concaténation des messages postés par l'utilisateur  $i$  pendant la période d'écoute  $t$ .

---

**Algorithme 1** : Algorithme de collecte - hypothèse stationnaire
 

---

**Input** :  $\mathcal{K}_{init}$

- 1 **for**  $t = 1..T$  **do**
- 2     **for**  $i \in \mathcal{K}$  **do**
- 3         Calculer  $s_{i,t}$  selon l'équation 2.3
- 4     **end**
- 5     Sélectionner  $\mathcal{K}_t$  selon  $\arg \max_{\mathcal{K}_t \subset \mathcal{K}} \sum_{i \in \mathcal{K}_t} s_{i,t}$ ;
- 6     Écouter en parallèle tous les utilisateurs  $i \in \mathcal{K}_t$  et observer  $\omega_{i,t}$ ;
- 7     **for**  $i \in \mathcal{K}_t$  **do**
- 8         Recevoir la récompense  $r_{i,t}$  associée à  $\omega_{i,t}$ ;
- 9         Alimenter  $\mathcal{K}$  avec les nouveaux utilisateurs  $j$ ,  $j \notin \mathcal{K}$
- 10    **end**
- 11 **end**

---

Dans notre tâche de collecte, la récompense attribuée aux utilisateurs après une période donnée est basée sur la pertinence du contenu qu'ils ont produit pendant cette période. Une difficulté pour notre problème d'exploration provient des fortes variations qui peuvent être observées sur la fréquence de publication des utilisateurs écoutés. La plupart du temps, les utilisateurs ne produisent aucun contenu. Avec la politique CUCB présentée ci-dessus (i.e., avec  $B_{i,t} = \sqrt{\frac{2 \ln(t)}{N_{i,t-1}}}$ ), le score  $s_{i,t}$  peut tendre à pénaliser des utilisateurs très pertinents, ne produisant que peu de contenu. Des utilisateurs produisant beaucoup de contenu, mais de qualité médiocre, peuvent se trouver favorisés, car obtenant une récompense non nulle à chaque écoute. Afin d'éviter ce genre de problème, une possibilité est de considérer une estimation de la variance des récompenses de chaque utilisateur. L'idée est d'inciter l'algorithme à reconsidérer plus régulièrement les utilisateurs à forte variance, dont l'estimation d'utilité est plus incertaine. Ceci nous a conduit à proposer une extension de l'algorithme UCBV, introduit dans [Audibert et al., 2009] pour prise en compte de la variance des actions, dans le cadre des problèmes à sélection multiple. Notre algorithme CUCBV suit comme CUCB l'algorithme 1, mais avec un score  $s_{i,t}$  défini selon (2.3) en utilisant le terme d'exploration  $B_{i,t}$  suivant :

$$B_{i,t} = \sqrt{\frac{2V_{i,t-1} \log(t)}{N_{i,t-1}}} + 3 \frac{\log(t)}{N_{i,t-1}} \quad (2.4)$$

où  $V_{i,t-1} = \frac{1}{N_{i,t-1}} \sum_{s=1}^{t-1} (\mathbb{1}_{\{i_t=i\}} r_{i,s} - \hat{\mu}_{i,t-1})^2$  est la variance empirique de l'action  $i$  (i.e., utilisateur  $i$  pour notre tâche). Dans [Gisselbrecht et al., 2015c], nous prouvons la sous-linéarité de cet algorithme, en majorant le pseudo-regret, défini pour le cas à sélection multiple par  $\hat{R}_T = \mathbb{E} \left[ \sum_{t=1}^T \left( \sum_{i \in \mathcal{K}^*} \mu_i - \sum_{i \in \mathcal{K}_t} \mu_i \right) \right]$  avec  $\mathcal{K}^*$  l'ensemble des  $k$  actions de plus forte espérance, selon le théorème 1 ci dessous.

**Théorème 1** *En considérant l'ensemble complet des actions  $\mathcal{K}$  connu a priori, le pseudo-regret de l'algorithme CUCBV est majoré par :*

$$\hat{R}_T \leq \ln(T) \sum_{i \in \mathcal{K}^*} \left( C + 8 \left( \frac{\sigma_i^2}{\delta_i^2} + \frac{2}{\delta_i} \right) \right) \Delta_i + D \quad (2.5)$$

où  $C$  et  $D$  sont des constantes.  $\Delta_i = \bar{\mu}^* - \mu_i$  est la différence entre  $\bar{\mu}^*$ , la moyenne des espérances de récompense des actions dans  $\mathcal{K}^*$ , et  $\mu_i$ , l'espérance de récompense pour  $i$ .  $\delta_i = \underline{\mu}^* - \mu_i$  est la différence entre  $\underline{\mu}^*$ , l'espérance associée à la moins bonne action de  $\mathcal{K}^*$ , et  $\mu_i$ .  $\sigma_i^2$  est la variance de l'action  $i$ .

Bien que ce résultat ne soit valide que pour le cas où l'ensemble complet des utilisateurs est connu *a priori*, il permet d'affirmer que lorsqu'un utilisateur optimal (avec une moyenne parmi les  $k$  meilleures) entre dans l'ensemble  $\mathcal{K}$ , le processus converge de façon logarithmique vers une politique le reconnaissant comme tel. Par ailleurs, ce résultat n'est valide que si l'hypothèse de stationnarité est vérifiée. Bien que ce ne soit pas le cas dans notre contexte de réseaux sociaux, cette preuve de convergence nous indique que si des sources sont utiles pendant une période de temps suffisamment longue, notre algorithme est en mesure de les détecter. Ceci est vérifié dans les expérimentations reportées dans la thèse de Thibault Gisselbrecht [Gisselbrecht, 2017].

**Résultats** La figure 2.2 présente un échantillon des résultats obtenus dans le cadre de cette contribution. Afin de pouvoir comparer divers algorithmes, les principales évaluations sont réalisées sur des bases hors-ligne, bien que l'objectif final reste bien entendu la collecte en situation réelle, pour laquelle des courbes de résultats sont données dans la figure 2.7. La figure 2.2 concerne une base de messages, nommée *USElections* dans [Gisselbrecht, 2017], collectés via l'API *Follow Streaming* de Twitter pendant la campagne électorale américaine de 2012, en suivant les 5000 premiers utilisateurs à avoir utilisé les mots-clés "Obama", "Romney" ou le hashtag "#USElections" au cours d'une collecte préliminaire selon l'API *Track Streaming*. Le corpus contient donc  $K = 5000$  utilisateurs pour un total de 3 587 961 messages s'étalant sur 10 jours. La fonction de récompense pour un message dépend de son score de classification en tant que message *Politique* selon un classifieur SVM, appris sur le jeu de données 20 Newsgroups<sup>2</sup>, pondéré par le nombre de fois où le message a été reposté par d'autres utilisateurs sur la période d'écoute. Les périodes d'écoute durent chacune  $\Delta_t = 100$  secondes, soit  $T = 8070$  pas de temps sur l'ensemble du corpus. Les courbes à gauche de la figure 2.2, obtenues avec  $k = 100$ , montrent le très bon comportement de l'algorithme CUCBV par rapport à divers concurrents tels que *CUCB* ou des extensions, pour sélection multiple, des algorithmes de bandit populaires *UCB- $\delta$*  [Abbasi-Yadkori et al., 2011] et *MOSS* [Audibert and Bubeck, 2009]. Le fait que tous ces algorithmes soient bien plus performants qu'une sélection aléatoire (courbe *Random*) conforte également la pertinence des approches de bandit pour notre tâche. D'autre part, la prise en compte de la variance dans CUCBV apparaît particulièrement efficace pour notre tâche, où la récompense obtenue dépend grandement de l'activité des utilisateurs. Le fait de donner plus de crédit aux profils à haute variabilité permet de reconsidérer des utilisateurs moins actifs mais très pertinents (et éviter par exemple de se focaliser sur des bots qui postent en continu des contenus de plus faible qualité). Le graphique à droite de la figure 2.2 montre la dépendance des résultats cumulés en fonction de  $k$ , nombre d'utilisateurs écoutés simultanément. Sans surprise les performances augmentent avec  $k$ , mais on note la supériorité de CUCBV (en vert) quel que soit ce nombre  $k$ .

2. Disponible à l'adresse <http://qwone.com/jason/20Newsgroups/>.

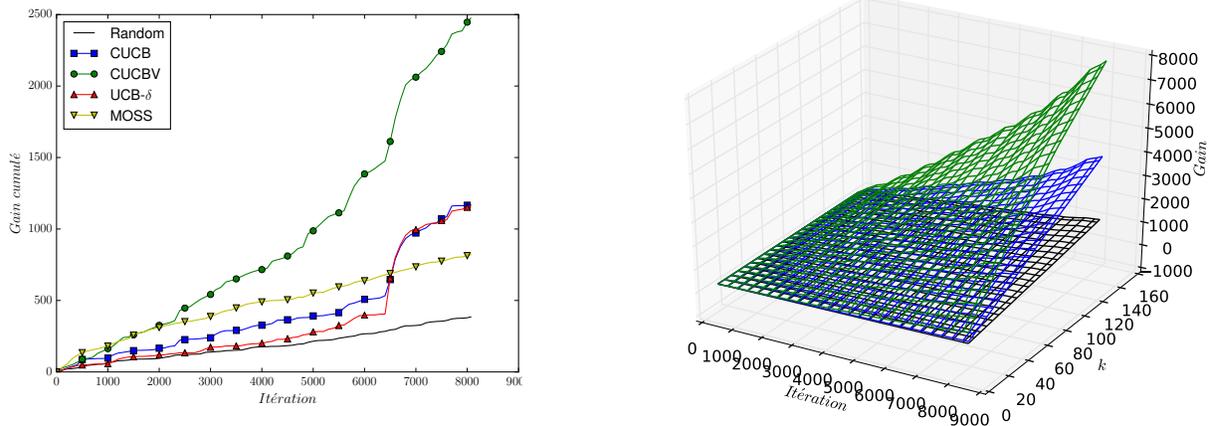


FIGURE 2.2 – Récompenses cumulées sur le jeu de données USElections.

**Publications associées :**

- Gisselbrecht, T., Denoyer, L., Gallinari, P., and Lamprier, S. (2015c). Whichstreams: A dynamic approach for focused data capture from large social media. In *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26-29, 2015*, pages 130–139
- Gisselbrecht, T., Denoyer, L., Gallinari, P., and Lamprier, S. (2015a). Apprentissage en temps réel pour la collecte d’information dans les réseaux sociaux. *Document Numérique*, 18(2-3):39–58
- Gisselbrecht, T., Denoyer, L., Gallinari, P., and Lamprier, S. (2015b). Apprentissage en temps réel pour la collecte d’information dans les réseaux sociaux. In *CORIA 2015 - Conférence en Recherche d’Informations et Applications - 12th French Information Retrieval Conference, Paris, France, March 18-20, 2015.*, pages 7–22

**2.2 Bandits contextuels pour la sélection active de flux de données**

Alors qu’à la section précédente on se limitait à des observations de récompense immédiate pour orienter la collecte, on considère maintenant la possibilité d’exploiter des informations auxiliaires obtenues au cours du processus. Cela est réalisé via des algorithmes de bandits contextuels, que cette section commence par introduire. Trois applications de ces algorithmes sont ensuite proposées pour répondre à notre problème de collecte de données dynamique à partir des flux des réseaux sociaux, pour exploiter la structure des utilisateurs et anticiper des variations d’utilité au cours du temps.

**2.2.1 Algorithmes de bandits contextuels**

Le problème du bandit contextuel fait l’hypothèse qu’il existe un contexte décisionnel qui sous-tend les distributions de récompense des différentes actions. Lorsque ce contexte est disponible, il est possible d’exploiter ces informations auxiliaires pour améliorer l’exploration des actions et éventuellement anticiper leurs évolutions d’utilité. Dans le problème du bandit contextuel, on considère

Contextes		Paramètres	
		Globaux : $\beta$	Individuels : $(\theta_1, \dots, \theta_K)$
Constants	Individuels : $x_i$	$x_i^\top \beta$	$x_i^\top \theta_i$
Variables	Globaux : $x_t$		$x_t^\top \theta_i$
	Individuels : $x_{i,t}$	$x_{i,t}^\top \beta$	$x_{i,t}^\top \theta_i$

TABEAU 2.1 – Différentes instances du bandit contextuel à K bras, avec formulation linéaire de l’espérance conditionnelle de l’utilité des bras.

qu’un contexte est présenté à l’agent décisionnel avant chaque décision. Ces observations permettent à l’agent d’apprendre la fonction - supposée linéaire dans le bandit contextuel classique étudié ici - liant contextes et récompenses. La structure induite dans l’espace des récompenses permet de mutualiser l’apprentissage et de s’orienter plus efficacement vers les actions les plus prometteuses à un instant donné. Le bandit contextuel peut se décliner sous différentes instances, comme synthétisé dans le tableau 2.1 :

- Avec contexte global (état général du monde au moment de la décision) ou individuel (chaque action a des caractéristiques qui lui sont propres)
- Avec contexte constant ou variable (i.e., le contexte évolue à chaque pas de temps  $t$ ).
- Avec prise en compte globale (paramètres partagés) ou individuelle (l’utilité espérée de chaque bras dépend différemment du contexte).

La variabilité des contextes (communs ou individuels) peut permettre d’appréhender des variations d’utilité pour chaque action. En outre, l’observation de contextes individuels (même fixes) peut permettre de mieux explorer les actions en exploitant la structure des actions dans l’espace de leurs caractéristiques. Notons que le cas du contexte global fixe n’a pas d’intérêt (même contexte pour toutes les actions, donc ne permet pas de discrimination entre elles, et stationnaire, donc ne permet pas d’appréhender des variations d’utilité espérée). Notons également le cas du contexte global variable avec prise en compte commune qui ne présente pas d’intérêt non plus (case grisée du tableau 2.1), puisque les utilités espérées sont dans ce cas les mêmes pour tous les bras. Différentes hybridations de ces instances peuvent être considérées [Li et al., 2011].

Dans ce qui suit, nous discutons des méthodes dans le cadre du bandit à contextes individuels variables avec paramètres partagés et hypothèse classique de linéarité suivante :

$$\exists \beta \in \mathbb{R}^d \text{ tel que } r_{i,t} = x_{i,t}^\top \beta + \eta_{i,t} \quad (2.6)$$

avec  $\eta_{i,t}$  un bruit sous-gaussien de moyenne nulle et de constante caractéristique  $R$ , c’est à dire :  $\forall \lambda \in \mathbb{R} : \mathbb{E}[e^{\lambda \eta_{i,t}} | \mathcal{H}_{t-1}] \leq e^{\lambda^2 R^2 / 2}$  avec  $\mathcal{H}_{t-1} = \{(i_s, x_{i_s, s}, r_{i_s, s})\}_{s=1..t-1}$  pour tout  $i \in \mathcal{K}$ .

Étant donné un ensemble  $\mathcal{K}$  de  $K$  actions, les algorithmes de bandit contextuel procèdent typiquement de la façon suivante à chaque itération  $t \in \{1, \dots, T\}$  :

1. Pour tout  $i \in \{1, \dots, K\}$ , observer  $x_{i,t} \in \mathbb{R}^d$ , le vecteur de contexte de chaque action  $i$  ;
2. Selon l’estimation courante de  $\beta$ , sélectionner l’action  $i_t$  et recevoir la récompense  $r_{i_t, t}$  ;
3. Améliorer la stratégie de décision en considérant la nouvelle observation  $(i_t, x_{i_t, t}, r_{i_t, t})$ .

Dans ce cadre, on s'intéresse généralement à trouver le vecteur  $\beta$  minimisant le pseudo-regret suivant :

$$\hat{R}_T = \sum_{t=1}^T x_{i_t^*, t}^\top \beta - x_{i_t, t}^\top \beta \quad (2.7)$$

avec  $i_t^* = \operatorname{argmax}_{i \in \mathcal{K}} x_{i, t}^\top \beta$  le bras de meilleure utilité espérée à l'instant  $t$ .

Une des approches de bandit contextuel les plus populaires est certainement l'algorithme LinUCB, introduit dans [Li et al., 2011], mais dont nous présentons ici la formulation de [Chu et al., 2011], qui correspond à une instance avec contextes individuels et paramètres communs. Cet algorithme est basé sur la résolution du problème de régression linéaire suivant à chaque étape  $t$  du processus, qui a pour but de minimiser l'erreur quadratique entre espérance conditionnelle connaissant le contexte  $\mathbb{E}[r_{i,s} | x_{i,s}] = x_{i,s}^\top \beta$  et récompense observée correspondante, pour toutes les paires contexte-récompense de l'historique  $\mathcal{H}_t$  :

$$\begin{aligned} \hat{\beta}_t &= \operatorname{argmin}_{\beta} \sum_{s=1}^t (r_{i_s, s} - x_{i_s, s}^\top \beta)^2 + \lambda \|\beta\|^2 = \operatorname{argmin}_{\beta} \|c_t - D_t \beta\|^2 + \lambda \|\beta\|^2 \\ &= (D_t^\top D_t + \lambda I)^{-1} D_t^\top c_t = V_t^{-1} b_t \end{aligned} \quad (2.8)$$

où  $\lambda$  est un hyper-paramètre de régularisation L2 (correspondant à un prior gaussien centré et de précision  $\lambda I$  pour le paramètre  $\beta$ ).  $V_t = D_t^\top D_t + \lambda I$  est une matrice  $d \times d$  et  $D_t = \begin{pmatrix} x_{i_s, s}^\top \\ \vdots \\ x_{i_t, t}^\top \end{pmatrix}_{s=1..t}$  correspond à la matrice de taille  $t \times d$  des contextes correspondant aux actions sélectionnées jusqu'au temps  $t$ .  $b_t = D_t^\top c_t$  est un vecteur de taille  $d$ , avec  $c_t = (r_{i_s, s})_{s=1..t}$  le vecteur de taille  $t$  des récompenses reçues jusqu'au temps  $t$ . Le théorème 2 ci-dessous définit un intervalle de confiance pour l'espérance de récompense de chaque action  $i$  au temps  $t$  selon  $x_{i,t}$  en fonction de l'estimateur  $\hat{\beta}_{t-1}$  de  $\beta$ .

**Théorème 2** [Li et al., 2011] Soit  $0 < \delta < 1$ , alors avec une probabilité au moins égale à  $1 - \delta$  on a, avec  $\alpha = 1 + \sqrt{\frac{\log(2/\delta)}{2}}$ , pour toute action  $i$  et à chaque instant  $t \geq 1$  :

$$|x_{i,t}^\top \hat{\beta}_{t-1} - x_{i,t}^\top \beta| \leq \alpha \sqrt{x_{i,t}^\top V_{t-1}^{-1} x_{i,t}} \quad (2.9)$$

À la manière des méthodes UCB classiques, cela permet de définir un score  $s_{i,t}$  pour chaque action, correspondant à la borne supérieure de l'intervalle de confiance de niveau  $1 - \delta$  pour l'espérance de récompense de l'action  $i$  à l'instant  $t$  :  $s_{i,t} = x_{i,t}^\top \hat{\beta}_{t-1} + \alpha \sqrt{x_{i,t}^\top V_{t-1}^{-1} x_{i,t}}$ . L'algorithme 2 décrit le fonctionnement complet de LinUCB. On note qu'un avantage majeur de cet algorithme est de ne pas avoir à retenir l'historique des observations, puisque les mises à jour des paramètres se font par ajouts successifs dans les structures  $V$  et  $b$  (lignes 11 et 12 de l'algorithme 2).

Bien que l'algorithme LinUCB classique soit extrêmement performant d'un point de vue expérimental [Li et al., 2011], son analyse théorique pose des difficultés techniques, car l'estimation du paramètre partagé  $\beta$  à chaque itération  $t$  dépend de l'historique des décisions précédant  $t$ . Pour outrepasser cette difficulté, les auteurs de [Chu et al., 2011] proposent une modification de LinUCB, appelée SupLinUCB, garantissant une estimation des paramètres indépendante des décisions passées. Bien que bien moins performant que LinUCB du fait de la dispersion des observations sur différents ensembles d'apprentissage, SupLinUCB permet de garantir un pseudo-regret sous-linéaire.

---

**Algorithme 2** : LinUCB [Chu et al., 2011]
 

---

**Input** :  $\alpha > 0, \lambda > 0$   
 1  $V = \lambda I$   
 2  $b = 0$   
 3 **for**  $t = 1..T$  **do**  
 4      $\hat{\beta} = V^{-1}b$   
 5     **for**  $i = 1..K$  **do**  
 6         Observer le contexte  $x_{i,t}$   
 7         Calculer  $s_{i,t} = x_{i,t}^\top \hat{\beta} + \alpha \sqrt{x_{i,t}^\top V^{-1} x_{i,t}}$   
 8     **end**  
 9     Sélectionner  $i_t = \arg \max_{i \in \{1, \dots, K\}} s_{i,t}$   
 10     Recevoir récompense  $r_{i_t}$   
 11      $V = V + x_{i_t,t} x_{i_t,t}^\top$   
 12      $b = b + r_{i_t,t} x_{i_t,t}$   
 13 **end**

---

L'algorithme OFUL (Optimism in the Face of Uncertainty Linear) [Abbasi-Yadkori et al., 2011], initialement proposé pour traiter un problème de bandit contextuel avec actions continues, peut être aisément adapté au cas discret qui nous intéresse. Basé sur la théorie des processus auto-normalisés de [de la Peña et al., 2009], [Abbasi-Yadkori et al., 2011] montre par le théorème ci-dessous qu'à chaque instant  $t$ , le paramètre  $\beta$  appartient avec une certaine probabilité à un ellipsoïde  $\mathcal{C}_t$  centré sur l'estimateur  $\hat{\beta}_t$ , calculé selon l'équation 2.8.

**Théorème 3** [Abbasi-Yadkori et al., 2011] Soient  $\delta > 0$  et  $\lambda > 0$ . Si  $\|\beta\| \leq S$ , alors avec une probabilité au moins égale à  $1 - \delta$  :

$$\beta \in \mathcal{C}_t = \left\{ \tilde{\beta} \in \mathbb{R}^d : \|\hat{\beta}_t - \tilde{\beta}\|_{V_t} \leq R \sqrt{2 \log \left( \frac{\det(V_t)^{1/2} \det(\lambda I)^{-1/2}}{\delta} \right)} + \lambda^{1/2} S \right\} \quad (2.10)$$

avec  $\|x\|_A = \sqrt{x^\top A x}$  la norme induite par une matrice définie positive  $A$

Dans le cas à actions continues, l'idée de [Abbasi-Yadkori et al., 2011] est alors de calculer un estimateur optimiste  $\tilde{\beta}_{t-1} = \arg \max_{\tilde{\beta} \in \mathcal{C}_{t-1}} (\max_{x \in \mathcal{D}_t} (x^\top \tilde{\beta}))$  puis de choisir l'action  $x_t = \arg \max_{x \in \mathcal{D}_t} x^\top \tilde{\beta}_{t-1}$ , avec  $\mathcal{D}_t$  le domaine des actions à considérer à l'instant  $t$ . Pour le cas discret avec un nombre d'actions fini, cela nous mène exactement au même algorithme que *LinUCB* (algorithme 2), en prenant :

$$s_{i,t} = x_{i,t}^\top \hat{\beta}_{t-1} + \alpha_{t-1} \sqrt{x_{i,t}^\top V_{t-1}^{-1} x_{i,t}} \text{ avec } \alpha_{t-1} = R \sqrt{2 \log \left( \frac{\det(V_{t-1})^{1/2} \det(\lambda I)^{-1/2}}{\delta} \right)} + \lambda^{1/2} S$$

où  $R$  est la constante caractéristique du bruit sous-gaussien des récompenses (voir (2.6)). Ce score ressemble très fortement au score utilisé par *LinUCB*, à ceci près que le paramètre d'exploration  $\alpha_{t-1}$  varie au cours du temps. Cela permet d'établir la borne du pseudo-regret du théorème ci-dessous, du même ordre que celle de *SupLinUCB*, avec un algorithme aussi efficace que *LinUCB*.

**Théorème 4** [Abbasi-Yadkori et al., 2011] Soit  $0 < \delta < 1$ . Sous les hypothèses  $\|\beta\| \leq S$  et  $|x^\top \beta| \leq 1$ , avec une probabilité au moins égale à  $1 - \delta$ , le pseudo-regret de l'algorithme OFUL est tel que<sup>3</sup> :

$$\hat{R}_T = \mathcal{O} \left( d \log(T) \sqrt{T} + \sqrt{dT \log \left( \frac{T}{\delta} \right)} \right) \quad (2.11)$$

Des algorithmes de Thompson sampling ont également été proposés pour le cadre contextuel [Chapelle and Li, 2011, Agrawal and Goyal, 2013]. Ceux-ci supposent un modèle bayésien des données, où l'on considère généralement :

- Un prior gaussien pour le paramètre  $\beta$  :  $p(\beta) = \mathcal{N}(\beta; 0, \sigma^2 \mathbf{1})$ ;
- Une vraisemblance gaussienne pour les récompenses :  $p(r_{i,t} | x_{i,t}, \beta) = \mathcal{N}(r_{i,t}; x_{i,t}^\top \beta, \sigma^2)$ .

On peut alors déterminer  $p(\beta | \mathcal{H}_{t-1})$  en fonction des observations réalisées jusqu'au temps  $t - 1$  :

$$\begin{aligned} p(\beta | \mathcal{H}_{t-1}) &\propto \prod_{s=1}^{t-1} \exp \left( -\frac{1}{2\sigma^2} (r_{i,s} - x_{i,s}^\top \beta)^2 \right) \exp \left( -\frac{\beta^\top \beta}{2\sigma^2} \right) \\ &\propto \exp \left( -\frac{1}{2\sigma^2} ((\beta - V_{t-1}^{-1} b_{t-1})^\top V_{t-1} (\beta - V_{t-1}^{-1} b_{t-1})) \right) \end{aligned} \quad (2.12)$$

On reconnaît la distribution postérieure gaussienne  $\mathcal{N}(\beta; V_{t-1}^{-1} b_{t-1}, \sigma^2 V_{t-1}^{-1})$ , de laquelle il est possible d'échantillonner un paramètre  $\tilde{\beta}$  à chaque étape de l'algorithme. L'action choisie à l'étape  $t$  est alors celle qui maximise  $x_{i,t}^\top \tilde{\beta}$ . La sous-linéarité du pseudo-regret de cet algorithme a été montrée dans [Agrawal and Goyal, 2013].

Notons que bien d'autres algorithmes existent pour diverses instances du bandit contextuel, tels que par exemples [Li et al., 2017] pour des modèles linéaires généralisés (e.g., problèmes de régression logistique), [Agrawal et al., 2014] pour des problèmes de classification avec coûts asymétriques, [Kazerouni et al., 2016] pour des problèmes de bandits contextuels avec garanties de robustesse, [Agrawal et al., 2015] pour des bandits avec contraintes de budget, ou encore [Slivkins, 2009] pour des bandits contextuels avec informations de similarité entre actions.

Plus proche de ce qui nous intéresse dans la suite, le problème du bandit contextuel avec sélections multiples a été étudié et appliqué à un scénario de recommandation dans [Qin et al., 2014]. Dans ce travail, les auteurs proposent l'algorithme  $\mathcal{C}^2\text{UCB}$  (Contextual Combinatorial UCB) et une borne (non optimale) associée. Il s'agit d'une extension de l'algorithme OFUL au cas des sélections multiples. Dans notre cas il s'agit de sélectionner à chaque instant les  $k$  actions ayant les meilleurs scores. Le pseudo-regret est défini de la façon suivante :

$$\hat{R}_T = \sum_{t=1}^T \left( \sum_{i \in \mathcal{K}_t^*} x_{i,t}^\top \beta - \sum_{i \in \mathcal{K}_t} x_{i,t}^\top \beta \right) \quad (2.13)$$

avec  $\mathcal{K}_t^*$  l'ensemble des  $k$  actions  $i \in \mathcal{K}$  de plus grande espérance de récompense  $x_{i,t}^\top \beta$ .

3. Classiquement,  $f = \mathcal{O}(g)$  implique que  $f$  est dominé par  $g$ . Il existe une constante strictement positive  $C$  telle qu'asymptotiquement :  $|f| \leq C|g|$ .

### 2.2.2 Collecte structurée avec profils cachés

Dans cette section, nous présentons une première application des bandits contextuels pour notre tâche de sélection de flux pour la collecte de données sur les réseaux sociaux. On se place dans le cadre d'un problème de bandit contextuel basé sur des profils  $\mu_i \in \mathbb{R}^d$  définis pour chaque action  $i$ , correspondant à des contextes individuels constants. L'hypothèse linéaire donnée par l'équation (2.6) devient alors :

$$\exists \beta \in \mathbb{R}^d \text{ tel que } r_{i,t} = \mu_i^\top \beta + \eta_{i,t} \quad (2.14)$$

avec  $\beta$  un vecteur de paramètres inconnus. Puisque les contextes sont constants, on reste donc dans un cadre stochastique stationnaire comme à la section 2.1.2, où l'objectif est de s'orienter rapidement vers les utilisateurs d'utilité moyenne maximale. L'idée est d'utiliser les profils des utilisateurs pour explorer plus efficacement l'espace, en faisant l'hypothèse que des utilisateurs aux profils proches tendent à produire des contenus pertinents pour les mêmes besoins de données. L'aspect contextuel n'est pas ici utilisé pour anticiper des variations d'utilité, mais permet d'améliorer l'exploration avec un paramètre  $\beta$  permettant de définir des zones d'intérêt dans l'espace de profils des utilisateurs. Les observations réalisées sur des utilisateurs informent alors de l'utilité espérée des utilisateurs de profil similaire. Cette instance de bandit structuré a déjà été investigué avec succès dans [Filippi et al., 2010]. Le pseudo-regret considéré est celui de la formule 2.13 en remplaçant  $x_{i,t}$  par  $\mu_i$  pour chaque action  $i$ .

Pour notre tâche de capture, un problème majeur est que l'on ne dispose d'aucune information a priori sur les utilisateurs du réseau (i.e., les profils des utilisateurs ne sont pas connus). Tout ce dont on dispose à chaque itération  $t$ , ce sont des observations pour un sous-ensemble d'utilisateurs  $\mathcal{O}_t$ , telles que pour tout  $i \in \mathcal{O}_t$ , un échantillon  $x_{i,t}$  d'une variable que l'on suppose centrée sur le profil  $\mu_i$  est révélé. Pour notre problème de collecte sur les réseaux,  $x_{i,t}$  correspond à du contenu publié par l'utilisateur  $i$  durant la période de  $t-1$  à  $t$ , que l'on suppose centré sur  $\mu_i$  (voir section 2.2.2.4). Notre instance du problème diffère donc des approches existantes de bandit structuré par deux aspects principaux :

1. Les profils des actions ne sont pas directement disponibles, seuls des échantillons centrés sur ces profils sont observables;
2. À chaque étape  $t$ , ces échantillons ne sont observés que pour un sous-ensemble d'actions  $\mathcal{O}_t$ .

À notre connaissance, cette instance du problème n'avait pas encore été étudiée dans la littérature. Nous avons alors proposé une nouvelle approche pour la traiter dans [Gisselbrecht et al., 2016d]. Parmi les travaux connexes, nous notons tout de même l'approche de [Wang et al., 2016], publiée simultanément à [Gisselbrecht et al., 2016d], qui adopte un esprit similaire de profils cachés, mais seulement partiellement. L'idée est de supposer que seule une partie du vecteur de profil est observée, mais que les récompenses observées s'expliquent aussi par d'autres facteurs latents. L'objectif est d'éviter des biais d'apprentissage, en supposant qu'une dépendance linéaire des récompenses ne peut être extraite des observations seules, mais que cette dépendance existe sur les profils complets. On peut aussi mentionner par exemple le travail plus récent de [Gutowski et al., 2018], qui augmente les profils des actions en fonction des erreurs de prédiction réalisées par l'agent sur les itérations passées. Par son aspect *clustering* des actions, ce travail se rapproche de [Gentile et al., 2014], dont l'objectif est d'exploiter des relations entre espérances d'actions.

Dans la suite de cette section, nous commençons par présenter une approche générique pour notre problème de bandit basé sur des profils cachés pour le cas  $k = 1$  (une seule action choisie à chaque itération). La section 2.2.2.1 formalise le problème, la section 2.2.2.2 établit des intervalles

de confiance pour les estimateurs d'espérance des différents bras, la section 2.2.2.3 propose notre algorithme de bandit contextuel `SamplInUCB` basé sur ces intervalles. Enfin, la section 2.2.2.4 décrit l'application de `SamplInUCB` à notre tâche de capture de données, en étendant l'algorithme au cas du bandit à sélections multiples.

### 2.2.2.1 Bandit contextuel sur profils cachés

Pour notre cas de bandit contextuel avec profils cachés, on suppose donc que pour toute action  $i$ , tous les échantillons  $x_{i,t} \in \mathbb{R}^d$  sont iid selon une distribution de moyenne  $\mu_i \in \mathbb{R}^d$ . On suppose également qu'il existe  $L > 0$  tel que  $\|x_{i,t}\| \leq L$  pour tout  $i \in \mathcal{K}$  et  $t \in \mathbb{N}$ . Enfin, on suppose qu'il existe une constante  $S > 0$  tel que  $\|\beta\| \leq S$ , avec  $\|v\| = \sqrt{v^\top v}$  la norme L2 d'un vecteur  $v$ .

La relation de linéarité de l'équation 2.14 peut se ré-écrire de la manière suivante pour tout temps  $t$ , afin d'y introduire les échantillons de profils observés :

$$\forall s \leq t : r_{i,s} = \mu_i^\top \beta + \eta_{i,s} = \hat{x}_{i,t}^\top \beta + (\mu_i - \hat{x}_{i,t})^\top \beta + \eta_{i,s} = \hat{x}_{i,t}^\top \beta + \epsilon_{i,t}^\top \beta + \eta_{i,s} = \hat{x}_{i,t}^\top \beta + \eta'_{i,t,s} \quad (2.15)$$

où  $\epsilon_{i,t} = \mu_i - \hat{x}_{i,t}$ ,  $\hat{x}_{i,t} = \frac{1}{n_{i,t}} \sum_{s \in \mathcal{T}_{i,t}^{obs}} x_{i,s}$ , avec  $\mathcal{T}_{i,t}^{obs} = \{s \leq t, i \in \mathcal{O}_s\}$  et  $n_{i,t} = |\mathcal{T}_{i,t}^{obs}|$ .  $n_{i,t}$  correspond donc au nombre de fois qu'un échantillon a été obtenu pour l'action  $i$  jusqu'au temps  $t$  et  $\hat{x}_{i,t}$  correspond à la moyenne empirique de ces échantillons.  $\epsilon_{i,t}$  correspond au biais d'estimation du profil  $\mu_i$  par l'estimateur  $\hat{x}_{i,t}$ . On peut montrer que la variable  $\eta'_{i,t,s} = \epsilon_{i,t}^\top \beta + \eta_{i,s}$  est conditionnellement sous-gaussienne de constante  $R_{i,t} = \sqrt{R^2 + \frac{L^2 S^2}{n_{i,t}}}$  (voir la proposition 1 de [Lamprier et al., 2018]). On pourrait alors de manière naïve utiliser des algorithmes de bandit classique considérant des estimations  $\hat{x}_{i,t}$  des profils à chaque étape  $t$ .

Néanmoins, l'incertitude liée à ces estimateurs doit être prise en compte dans les choix des bras à actionner, additionnellement à l'incertitude sur l'estimation du paramètre  $\beta$  classiquement considérée dans les approches de bandit contextuel, renforcée par ailleurs par le fait que cette estimation repose alors sur des entrées estimées plutôt que sur les vrais profils (sur laquelle l'hypothèse de linéarité est exprimée). La figure 2.3 illustre notre instance de problème. Contrairement au cas classique où les profils sont connus, nous ne possédons à chaque instant que de zones de confiance, représentées par des cercles, centrées sur la moyenne des observations pour chaque action correspondante  $i$  (croix bleues). Selon la loi des grands nombres, plus on collecte d'observations pour une action donnée  $i$ , plus la moyenne de ses observations tend vers son vrai profil (profils représentés par des croix noires dans la figure 2.3). La surface des zones de confiance réduit donc avec le nombre d'observations collectées. La couleur de fond de la figure représente le score de sélection selon un algorithme tel que *OFUL* à une itération donnée  $t$ , pour un exemple de processus avec  $k = 4$  et  $d = 2$ . Les régions vertes correspondent à de forts scores de sélection pour les actions s'y trouvant (zones mal connues ou de fort potentiel), les rouges à de faibles scores (zones bien connues et de faible potentiel). Les variations de couleur rendent alors compte de la structure apprise par l'algorithme de bandit<sup>4</sup>. La meilleure action selon les scores de sélection est l'action 1,  $\mu_1$  se situant dans une zone vert foncé. Cependant, les vrais profils  $\mu$  sont inconnus, on ne peut donc pas choisir les actions sur cette base. Si on se base naïvement sur les estimations  $\hat{x}_{i,t}$  pour chaque  $i$ , c'est l'action 2 sous-optimale qui est choisie (et risque de l'être pour toutes les itérations futures). Nous proposons de considérer l'incertitude sur les estimations, en choisissant pour chaque action  $i$  la position à l'intérieur de sa zone de confiance

4. Notons que la figure 2.3 n'est qu'une illustration du principe général, en pratique la surface des scores de sélection devrait aussi prendre en compte le fait que  $\beta$  est estimé à partir d'entrées possiblement biaisées, et être différente pour chaque bras selon son nombre d'observations, comme nous le verrons par la suite.

pour laquelle son score de sélection  $s_{i,t}$  est maximisé. Cela permet de définir une politique optimiste qui sélectionnerait l'action prometteuse 1 dans l'exemple de la figure 2.3 (i.e., la zone de l'action 1 contient des positions plus vertes que les autres).

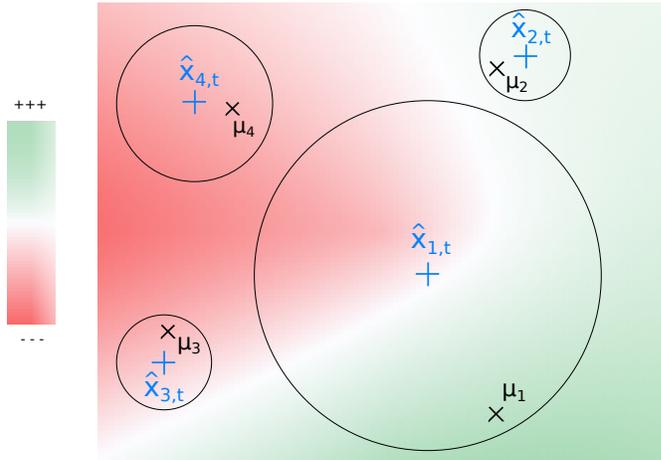


FIGURE 2.3 – Zones de confiance sur une surface de scores type *OFUL*, pour 4 actions à une itération donnée  $t$ .

Dans la suite, nous proposons un algorithme adapté à ce cas, dans un contexte de processus de révélation des profils générique, que l'on déclinera pour les trois cas suivants :

- **Cas 1** : Toutes les actions délivrent un échantillon à chaque pas de temps  $t : \forall t, \mathcal{O}_t = \mathcal{K}$  ;
- **Cas 2** : Chaque action  $i$  possède une probabilité  $p_i$  de délivrer un échantillon : à chaque pas de temps  $t$ , chaque action  $i$  est incluse dans  $\mathcal{O}_t$  avec une probabilité  $p_i$  ;
- **Cas 3** : À chaque pas de temps  $t$ , seule l'action sélectionnée à l'itération précédente délivre un échantillon :  $\forall t, \mathcal{O}_t = \{i_{t-1}\}$ .

Le premier cas correspond au cas le plus simple, où la seule différence avec un problème de bandit contextuel traditionnel provient du fait que les observations de contexte sont bruitées. Le second cas introduit une difficulté supplémentaire au sens où, à chaque instant, toutes les actions n'ont pas le même nombre d'observations, entraînant ainsi une différence d'incertitude entre ces dernières. Finalement, le dernier cas, à notre sens le plus utile pour des applications réelles comme notre tâche de collecte dynamique sur les réseaux sociaux, est le plus difficile puisque les décisions à chaque étape affectent non seulement notre connaissance des distributions de récompenses de chaque bras, mais aussi celle de leurs profils associés. Pour ce cas, il paraît primordial de prendre en compte l'incertitude sur les profils dans la politique de sélection, afin de garantir une convergence vers les actions optimales.

### 2.2.2.2 Intervalles de confiance pour profils cachés

Selon un processus de sélection donné, tel que celui d'*OFUL* [Abbasi-Yadkori et al., 2011] appliqué au cas à actions discrètes sur lequel nous construisons notre proposition, les actions choisies succes-

sivement, qui forment la séquence  $(i_s)_{s=1}^\infty$ , suivent des dépendances difficiles à manipuler. À l'instar de [Abbasi-Yadkori et al., 2011], nous relaxons le problème afin de définir des intervalles de confiance pour un problème plus général, valides pour n'importe quelle séquence  $(i_s)_{s=1}^\infty$ , qui ne dépendraient pas de sa construction. De la même manière, pour simplifier le problème, on va s'intéresser dans un premier temps à des estimations de profil fixées, dont on disposerait à l'issue d'observations provenant d'un processus annexe au processus de décision, indépendant des choix d'actions successifs. On note  $\hat{x}_{i_s}$  l'estimation, fixe, dont on dispose pour l'action sélectionnée au temps  $s$  du processus, dont on sait qu'elle correspond à une moyenne de  $n_{i_s}$  échantillons issus d'une distribution centrée sur le profil  $\mu_{i_s}$ .

Afin d'être à même d'utiliser la théorie des processus auto-normalisés [de la Peña et al., 2009], et notamment d'appliquer le théorème 1 de [Abbasi-Yadkori et al., 2011] sur les bornes auto-normalisées, il nous faut travailler avec des erreurs de prédictions  $(\eta'_s)_{s=1}^\infty$  sous-gaussiennes. Nous avons vu à l'occasion de l'équation 2.15, que pour les estimations  $(\hat{x}_{i_s})_{s=1}^\infty$  et des récompenses  $(r_s)_{s=1}^\infty$  associées, elles le sont bien pour un paramètre  $\beta$  donné si tant est que les écarts  $(\eta_s)_{s=1}^\infty$  en fonction des profils  $(u_{i_s})_{s=1}^\infty$  le sont. Cependant leurs constantes caractéristiques ne sont pas toutes les mêmes, cela dépend des nombres d'échantillons sur lesquels se basent les estimations de profil correspondantes. Cela empêche d'appliquer directement le théorème 1 de [Abbasi-Yadkori et al., 2011]. Nous nous employons dans la suite à normaliser ces constantes, en définissant notamment l'estimateur de  $\beta$  suivant, qui pondère chaque élément en fonction de l'incertitude associée à l'estimateur de profil utilisé :

$$\hat{\beta}_t = \arg \min_{\beta} \sum_{s=1}^t \frac{1}{R_{i_s}} (\beta^\top \hat{x}_{i_s} - r_s)^2 + \lambda \|\beta\|^2 \quad (2.16)$$

avec  $R_{i_s} = \sqrt{R^2 + \frac{L^2 S^2}{n_{i_s}}}$  la constante caractéristique de la variable d'écart  $\eta'_s$  (voir paragraphe sous (2.15)). La solution analytique de cet estimateur est donnée par :

$$\hat{\beta}_t = (D_t^\top A_t D_t + \lambda I)^{-1} D_t^\top A_t c_t \quad (2.17)$$

avec  $D_t$  la matrice de taille  $t \times d$  des  $t$  estimateurs de profil correspondant aux  $t$  actions sélectionnées dans les  $t$  premiers pas de temps du processus,  $c_t$  le vecteur des récompenses de taille  $t$  et  $A_t$  la matrice diagonale de taille  $t \times t$  contenant les facteurs de normalisation considérés (i.e.,  $A_t(s, s) = 1/R_{i_s}$  pour tout  $s$  de 1 à  $t$ ).

En utilisant une méthode similaire à celle du théorème 2 de [Abbasi-Yadkori et al., 2011], on obtient :

$$\|\hat{\beta}_t - \beta\|_{V_t} \leq \|D_t^\top A_t \eta'_{1:t}\|_{V_t^{-1}} + \lambda \|\beta\|_{V_t^{-1}}$$

où  $V_t = \lambda I + D_t^\top A_t D_t$ ,  $\|x\|_{V_t^{-1}} = \sqrt{x^\top V_t^{-1} x}$  est la norme du vecteur  $x$  induite par la matrice  $V_t^{-1}$  et  $\eta'_{1:t}$  est le vecteur des  $t$  écarts  $\eta'_s$  pour  $s$  de 1 à  $t$ . Puisque  $\|\beta\| \leq S$  et  $\|\beta\|_{V_{t-1}^{-1}}^2 \leq \|\beta\|^2 / \lambda_{\min}(V_{t-1}) \leq \|\beta\|^2 / \lambda$ , on obtient que  $\lambda \|\beta\|_{V_{t-1}^{-1}} \leq \sqrt{\lambda S}$ . Le premier terme est équivalent à  $\|S_t\|_{V_t^{-1}}$  avec  $S_t = \sum_{s=1}^t (\eta'_s / R_{i_s}) \hat{x}_{i_s}$ . On commence par noter que, puisque les variables  $\eta'_s$  sont centrées sur 0 (sous-gaussianité), la séquence  $(S_s)_{s=0}^\infty$  est une martingale par rapport à toute filtration  $(\mathcal{F}_s)_{s=0}^\infty$  telle que  $\hat{x}_{i_s}$ ,  $\eta'_s$  et  $R_{i_s}$  sont  $\mathcal{F}_s$ -mesurables pour tout  $s \geq 1$ . On fait alors appel à la méthode de mixtures de [de la Peña et al., 2009], utilisée également dans [Abbasi-Yadkori et al., 2011]. En notant que les variables  $(\eta'_s / R_{i_s})$  sont toutes sous-gaussiennes de constante caractéristique 1, on peut montrer que l'espérance de la quantité  $M_\tau^\gamma = \exp(\langle \gamma, S_\tau \rangle - (1/2) \|\gamma\|_{\bar{V}_\tau}^2)$  pour tout temps  $\tau \geq 1$ , avec  $\bar{V}_\tau = V_\tau - \lambda I$  et pour tout vecteur  $\gamma$ ,

est inférieure ou égale à 1 (car la séquence  $(M_s^Y)_{s=0}^\infty$  est une super-martingale par rapport à la filtration  $(\mathcal{F}_s)_{s=0}^\infty$ ). L'idée est ensuite de considérer l'espérance  $M_\tau$  selon un vecteur  $\gamma$  qui serait issu d'une gaussienne centrée de variance  $\bar{V}_t^{-1}$ . Après divers arrangements calculatoire détaillés dans la preuve du théorème 1 de [Abbasi-Yadkori et al., 2011], on obtient l'inégalité  $\mathbb{P}\left(\|S_\tau\|_{(V_\tau)^{-1}}^2 > 2 \log\left(\frac{\det(V_\tau)^{1/2}}{\delta \det(\lambda I)^{1/2}}\right)\right) \leq \mathbb{E}[M_\tau] \delta \leq \delta$ . La dernière inégalité est obtenue car  $\mathbb{E}[M_\tau] \leq 1$ , sachant que l'on a vu que pour tout  $\gamma$ ,  $\mathbb{E}[M_\tau^Y] \leq 1$ . Cela permet de borner la probabilité qu'un mauvais évènement (i.e.,  $\|S_\tau\|_{(V_\tau)^{-1}}^2$  dépasse une certaine valeur d'intérêt) arrive au temps  $\tau$  selon un paramètre  $\delta \in ]0; 1[$ . Sachant que cette borne est vraie pour tout temps  $\tau > 0$ , on peut l'utiliser pour borner la probabilité que ce mauvais évènement arrive pour un temps  $\tau < \infty$  (*stopping time construction* [Freedman, 1975]). Cela permet alors de borner  $\|D_t^\top A_t \eta'_{1:t}\|_{V_t^{-1}}$  pour tout temps  $t$  simultanément avec une probabilité au moins égale à  $1 - \delta$  et donc d'être à même de définir un ellipsoïde de confiance pour  $\beta$  selon le théorème 5 ci-dessous.

**Théorème 5** *Pour tout  $0 < \delta < 1$ , avec une probabilité au moins égale à  $1 - \delta$ , l'estimateur  $\hat{\beta}_{t-1}$  vérifie pour tout  $t \geq 0$  :*

$$\|\hat{\beta}_t - \beta\|_{V_t} \leq \sqrt{2 \log\left(\frac{\det(V_t)^{1/2} \det(\lambda I)^{-1/2}}{\delta}\right)} + \sqrt{\lambda} S \triangleq \alpha_t \quad (2.18)$$

$$\text{avec } V_t = \lambda I + D_t^\top A_t D_t = \lambda I + \sum_{s=1}^t \frac{\hat{x}_{i_s} \hat{x}_{i_s}^\top}{R_{i_s}}$$

Cette borne de l'erreur d'estimation de  $\beta$  est très similaire à celle donnée dans [Abbasi-Yadkori et al., 2011]. Cependant, une différence notable réside dans la définition de la matrice  $V_t$ , dans laquelle la matrice diagonale de poids  $A_t$  est appliquée pour parer au différences de confiance associée aux différents estimateurs de profil considérés. De cette borne d'erreur, on en vient facilement à l'inégalité suivante, valide pour tout  $i \in \mathcal{K}$  et pour tout  $t \geq 0$ , avec une probabilité supérieure à  $1 - \delta$  :

$$\beta^\top \mu_i \leq \hat{\beta}_{t-1}^\top \mu_i + \alpha_{t-1} \|\mu_i\|_{V_{t-1}^{-1}} \quad (2.19)$$

Cependant cette borne supérieure de récompense espérée pour tout bras  $i$  ne peut pas être directement utilisée dans notre cas pour définir une politique de type *UCB* car les profils  $\mu_i$  sont inconnus. Nous devons donc considérer également des ellipsoïdes de confiance pour les estimateurs de profil, pour lesquels nous utilisons l'inégalité de concentration de Hoeffding. Pour tout  $i \in \mathcal{K}$  et tout  $t > 0$ , avec une probabilité supérieure à  $1 - \delta/t^2$ , on a :

$$\|\hat{x}_{i,t} - \mu_i\| \leq \min(L \sqrt{\frac{2d}{n_{i,t}} \log\left(\frac{2dt^2}{\delta}\right)}, 2L) \triangleq \rho_{i,t,\delta} \quad (2.20)$$

Contrairement à la borne sur la déviation de l'estimateur de  $\beta$  qui était valide pour tout temps simultanément (grâce à la construction sur temps d'arrêt - *stopping time construction* [Freedman, 1975] - valide pour les martingales), celle sur les estimateurs de profil n'est valide que pour chaque pas de temps séparément. Pour obtenir une borne valide pour tous les pas de temps simultanément, ce qui est important pour la dérivation d'une borne du regret, nous utilisons le principe de borne uniforme. Pour toute action  $i$ , on a :  $\mathbb{P}(\forall t, \|\hat{x}_{i,t} - \mu_i\| \leq \rho_{i,t,\delta}) = 1 - \mathbb{P}(\exists t, \|\hat{x}_{i,t} - \mu_i\| \geq \rho_{i,t,\delta}) \geq 1 - \sum_t \mathbb{P}(\|\hat{x}_{i,t} - \mu_i\| \geq \rho_{i,t,\delta}) \geq 1 - \sum_t \delta/t^2$ . Cela justifie l'introduction du terme  $t^2$  dans la borne, permettant de définir une probabilité uniforme sur tous les pas de temps :  $\mathbb{P}(\forall t, \|\hat{x}_{i,t} - \mu_i\| \leq \rho_{i,t,\delta}) \geq 1 - \delta - \sum_{t=2}^\infty \delta/t^2 = 1 - \delta - \delta(\pi^2/6 - 1) \geq 1 - 2\delta$ .

Maintenant que nous avons défini des bornes probabilistes d'écart pour nos différents estimateurs, on peut les utiliser conjointement pour définir un intervalle de confiance pour l'espérance de récompense de chaque bras  $i \in \mathcal{K}$ , dans l'état des connaissances à chaque temps  $t$ .

**Théorème 6** *Pour tout  $i \in \mathcal{K}$  et tout  $t > 0$ , avec une probabilité supérieure à  $1 - \delta/t^2 - \delta$ , on a :*

$$\beta^\top \mu_i \leq \hat{\beta}_{t-1}^\top (\hat{x}_{i,t} + \bar{\epsilon}_{i,t}) + \alpha_{t-1} \|\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}\|_{V_{t-1}^{-1}} \quad (2.21)$$

avec :

$$\bar{\epsilon}_{i,t} = \frac{\rho_{i,t} \delta \hat{\beta}_{t-1}}{\|\hat{\beta}_{t-1}\|} \quad \tilde{\epsilon}_{i,t} = \frac{\rho_{i,t} \delta \hat{x}_{i,t}}{\sqrt{\lambda} \|\hat{x}_{i,t}\|_{V_{t-1}^{-1}}}$$

$$\beta_t = V_t^{-1} b_t \quad V_t = \lambda I + \sum_{s=1}^t \frac{\hat{x}_{i_s,t} \hat{x}_{i_s,t}^\top}{R_{i_s,t}} \quad b_t = \sum_{s=1}^t \frac{r_{i_s,t} \hat{x}_{i_s,t}^\top}{R_{i_s,t}}$$

Notons que nous sommes revenus dans un cadre où on considère des estimations de profil qui évoluent au cours du temps, en fonction des observations d'échantillons réalisées à chaque pas de temps, cadre que nous avons quitté pour simplifier l'établissement des bornes sur l'estimateur de  $\beta$ . Pour une meilleure efficacité de l'apprentissage, lorsqu'une nouvelle estimation  $\hat{x}_{i,t}$  est disponible pour une action  $i$ , on reconsidère toutes les itérations où cette action a été sélectionnée dans le passé, afin de faire profiter pleinement l'apprentissage de  $\beta$  de cette nouvelle connaissance. Cela peut se faire efficacement selon les règles de mise à jour suivantes, appliquées à chaque pas de temps pour chaque action  $i \in \mathcal{O}_t$  :

$$V_t \leftarrow V_t + N_{i,t} \left( \frac{\hat{x}_{i,t} \hat{x}_{i,t}^\top}{R_{i,t}} - \frac{\hat{x}_{i,t-1} \hat{x}_{i,t-1}^\top}{R_{i,t-1}} \right); \quad b_t \leftarrow b_t + B_{i,t} \left( \frac{\hat{x}_{i,t}}{R_{i,t}} - \frac{\hat{x}_{i,t-1}}{R_{i,t-1}} \right) \quad (2.22)$$

où  $\hat{x}_{i,t+1}$  et  $\hat{x}_{i,t}$  correspondent respectivement à l'ancienne et la nouvelle estimation pour  $i$ , avec  $N_{i,t}$  le nombre de fois où l'action  $i$  apparaît dans  $V_t$  (le nombre de fois où elle a été sélectionnée avant  $t$ ) et  $B_{i,t}$  la somme des récompenses associées. Ces mises à jour, réalisées avant tout choix d'action à l'itération  $t$ , permettent de conserver des versions des structures  $V$  et  $b$  à jour (selon les définitions données au théorème 6) sans avoir à stocker l'historique des actions-récompenses passées. Après le choix de l'action  $i_t$ ,  $\frac{1}{R_{i_t,t}} \hat{x}_{i_t,t} \hat{x}_{i_t,t}^\top$  et  $\frac{r_{i_t,t}}{R_{i_t,t}} \hat{x}_{i_t,t}$  sont respectivement ajoutés à  $V_t$  et  $b_t$ .

Les détails de la preuve du théorème 6 sont donnés dans [Lamprier et al., 2018]. Mentionnons simplement le fait que pour chaque  $i$ ,  $\bar{\epsilon}_{i,t}$ , qui est colinéaire à  $\hat{\beta}_{t-1}$ , permet la translation de  $\hat{x}_{i,t}$  dans la zone de confiance de  $\mu_i$ , de manière à ce que  $\hat{\beta}_{t-1}^\top \mu_i$  soit majoré probablement.  $\tilde{\epsilon}_{i,t}$  est quant à lui colinéaire à  $\hat{x}_{i,t}$ . Ce terme est utilisé pour déplacer l'estimateur  $\hat{x}_{i,t}$  de manière à ce que la norme  $\|\mu_i\|_{V_{t-1}^{-1}}$  soit également majorée. Ainsi, l'incertitude sur les profils est intégrée dans les deux termes exploitation-exploration de notre score de sélection, ce qui permet de définir une politique optimiste. Mentionnons également que la borne du théorème 2.18 reste valide malgré les mises à jour des estimations de (2.22) dans  $V_t$  et  $b_t$ . Pour s'en assurer, on note tout d'abord que, si tant est que  $S_t$  et  $V_t$  soient  $\mathcal{F}_t$  mesurables pour tout  $t$  par rapport à la filtration  $(\mathcal{F}_t)_{t=0}^\infty$  considérée (i.e., la filtration  $(\mathcal{F}_t)_{s=0}^\infty$  repose sur des  $\sigma$ -algèbres contenant pour chaque  $t$  les échantillons observés  $((x_{i,s})_{i \in \mathcal{O}_s})_{s=0}^t$ , les actions jouées  $(i_s)_{s=0}^t$  et les écarts de prédiction  $(\eta'_s)_{s=0}^t$  jusqu'à  $t$ ),  $(\|S_t\|_{V_t^{-1}}^2)_{t=0}^\infty$  reste une martingale par rapport à  $(\mathcal{F}_t)_{t=0}^\infty$ , même si  $S_t$  et  $V_t$  sont possiblement entièrement redéfinis à chaque étape en fonction des nouveaux échantillons. Puisque  $\mathbb{E}[\mu_i | \hat{x}_{i,t}] = \hat{x}_{i,t}$ , et que les nouveaux échantillons pour  $i$  sont centrés sur  $\mu_i$ , on a  $\mathbb{E}[\hat{x}_{i,t+1} | \hat{x}_{i,t}] = \hat{x}_{i,t}$ . Autrement dit en espérance les nouvelles observations ne modifient pas les contenus historiques de  $S_t$  et  $V_t$ . On a toujours bien  $\mathbb{E}[\|S_{t+1}\|_{V_{t+1}^{-1}}^2 | \mathcal{F}_t] = \|S_t\|_{V_t^{-1}}^2$ ,

car par ailleurs  $S_t$  est une somme de variables d'espérance nulle (estimations de profils pondérés par des écarts sous-gaussiens). On peut alors borner chaque  $\|S_t\|_{(V_t)^{-1}}^2$  en fonction de  $V_t$  comme précédemment en considérant comme estimations fixes les estimations courantes à  $t$ , puis utiliser les propriétés de la martingale  $(\|S_t\|_{V_t}^2)_{t=0}^\infty$  pour affirmer que la borne du théorème 6 est valide pour tout  $t > 0$  simultanément, avec une probabilité d'au moins  $1 - \delta$ , même dans notre cas avec mises à jour successives des estimations au fur et à mesure du processus.

### 2.2.2.3 Algorithme SampLinUCB

L'idée est alors de se servir de la borne définie au théorème 6, afin de définir un score de sélection optimiste que l'on peut utiliser dans un algorithme de bandit contextuel, que l'on nomme *SampLinUCB*. À chaque pas de temps  $t$  on sélectionne alors l'action  $i \in \mathcal{K}$  qui maximise :

$$s_{i,t} = (\hat{x}_{i,t} + \bar{e}_{i,t})^\top \hat{\beta}_{t-1} + \alpha_{t-1} \|\hat{x}_{i,t} + \bar{e}_{i,t}\|_{V_{t-1}^{-1}} \quad (2.23)$$

Pour les cas 2 et 3 de délivrance de contexte (voir fin de section 2.2.2.1), il est possible de ne disposer d'aucun échantillon de profil (i.e.,  $n_{i,t} = 0$ ) en début de processus, ce qui est gênant pour le calcul du score. Pour le cas 2, dans lequel on n'est pas actif sur l'observation des contextes, ceci peut être contourné en ignorant simplement ces actions jusqu'à ce qu'un premier échantillon soit observé pour elles. Pour le cas 3 cependant, les échantillons ne sont obtenus que pour les actions sélectionnées. Dans ce cas, nous devons, comme pour l'algorithme présenté en section 2.1.2, forcer la sélection prioritaire des actions  $i$  telles que  $n_{i,t} = 0$ , en fixant  $s_{i,t} = +\infty$ .

Le score de sélection défini par (2.23) correspond à la borne supérieure de l'intervalle de confiance de l'espérance de récompense pour chaque action, comme c'est le cas dans toutes les politiques de type *UCB*. Intuitivement, ce score conduit l'algorithme à choisir des actions pour lesquelles l'estimation de profil est soit dans une zone de fort potentiel, soit est suffisamment incertaine pour considérer que l'action peut être quand même potentiellement utile. L'objectif est d'écarter rapidement les "mauvaises" actions, dont l'ellipsoïde de confiance ne permet pas d'inclure des positions de l'espace intéressantes au sens de  $\beta$ . On peut montrer que le score  $s_{i,t}$  peut se ré-écrire de la façon suivante :

$$s_{i,t} = \hat{x}_{i,t}^\top \hat{\beta}_{t-1} + \alpha_{t-1} \|\hat{x}_{i,t}\|_{V_{t-1}^{-1}} + \rho_{i,t,\delta} \left( \|\hat{\beta}_{t-1}\| + \frac{\alpha_{t-1}}{\sqrt{\lambda}} \right) \quad (2.24)$$

avec  $\rho_{i,t,\delta}$  défini selon (2.20). Cette nouvelle formulation permet de voir sous un jour différent le comportement de l'algorithme *SampLinUCB*. La première partie du score  $\hat{x}_{i,t}^\top \hat{\beta}_{t-1} + \alpha_{t-1} \|\hat{x}_{i,t}\|_{V_{t-1}^{-1}}$  est similaire au score que l'on aurait utilisé dans l'algorithme OFUL (bien que définie selon une construction différente de  $V_t$ ). Mais le score exhibe une partie additionnelle  $\rho_{i,t,\delta} \left( \|\hat{\beta}_{t-1}\| + \frac{\alpha_{t-1}}{\sqrt{\lambda}} \right)$ , dont le facteur  $\rho_{i,t,\delta}$  permet de garantir l'exploration nécessaire liée à l'incertitude sur les estimations de profils. Cela met en évidence la prime accordée aux actions les moins connues. Notons que pour le cas 1, cette partie additionnelle est la même pour toutes les actions. Elle pourrait donc être retirée du score puisqu'elle ne permet aucune discrimination d'une action par rapport à une autre. Cependant, ce terme d'exploration additionnel est particulièrement important pour le cas 3, dans lequel le processus de délivrance d'échantillons est directement connecté à la politique de sélection, en empêchant par exemple d'écarter des actions optimales qui auraient souffert d'échantillons peu prometteurs dans les premières étapes du processus.

L'utilisation de ce score de sélection permet de garantir une borne supérieure du pseudo-regret sous-linéaire. À partir d'une borne générique, il est possible d'établir les bornes spécifiques suivantes,

correspondant aux trois cas de délivrance de contexte considérés, valides avec une probabilité d'au moins  $1 - 3\delta$  :

- **Cas 1** :  $\hat{R}_T = \mathcal{O} \left( d \log \left( \frac{T}{\delta} \right) \sqrt{T \log(T)} \right)$ ;
- **Cas 2** :  $\hat{R}_T = \mathcal{O} \left( d \log \left( \frac{T}{\delta} \right) \sqrt{T \frac{\log(T)}{p}} \right)$ , pour  $T \geq 2 \log(1/\delta)/p^2$ , avec  $p$  la probabilité de délivrance d'un échantillon pour chaque action à chaque étape;
- **Cas 3** :  $\hat{R}_T = \mathcal{O} \left( d \log \left( \frac{T}{\delta} \right) \sqrt{TK \log \left( \frac{T}{K} \right)} \right)$

Quel que soit le cas, `SampLinUCB` garantit donc bien un pseudo-regret cumulé sous-linéaire. La borne pour le cas 2 possède une dépendance additionnelle en  $p$ , qui est la probabilité pour chaque bras de recevoir un nouvel échantillon à chaque étape. De manière évidente, plus cette probabilité est élevée, plus l'incertitude sur les profils diminue rapidement. Notons cependant que cette borne pour le cas 2 n'est valide qu'à partir d'un certain nombre d'itérations, inversement proportionnel à  $p^2$ . La borne pour le cas 3 dépend du nombre d'actions disponibles  $K$ . Ceci vient du fait que seuls l'action sélectionnée obtient un échantillon à chaque étape du processus, ce qui ré-introduit la nécessité de considérer chaque action régulièrement, comme c'est le cas pour `UCB` par exemple, afin de reconsidérer leurs estimations. Cependant, l'utilisation d'une structure des actions, qui permet un apprentissage mutualisé des distributions de récompenses, tend à améliorer significativement les performances par rapport aux approches classiques, notamment pour notre tâche de collecte comme détaillé ci-dessous.

#### 2.2.2.4 Application à la collecte dynamique de données

Pour notre tâche de collecte de données ciblée sur les media sociaux, nous devons étendre l'algorithme précédent au cas à sélection multiple. Cela peut se faire de manière directe, en sélectionnant les  $k$  meilleures actions au sens de  $s_{i,t}$ , défini par (2.24), à chaque pas de temps  $t$ . Des bornes sous-linéaires du pseudo-regret cumulé peuvent être dérivées pour ce cadre où  $k \geq 1$  [Lamprier et al., 2018].

Nous considérons que le profil d'un utilisateur  $i$  correspond à la moyenne des contenus qu'il poste sur le réseau :  $\mu_i = \lim_{T \rightarrow \infty} \frac{1}{T} \sum_{t=1}^T x_{i,t}$ , où  $x_{i,t}$  correspond au contenu que  $i$  a posté entre les pas de temps  $t-1$  et  $t$ . Dans nos expérimentations, on utilise une représentation *bag-of-words TF* (*Term Frequency*) sur laquelle on applique une variante de la méthode LDA (*Latent Dirichlet Allocation*) adaptée aux textes courts [Hong and Davison, 2010]. L'objectif de l'utilisation de LDA est de réduire la dimension de l'espace des échantillons  $d$ , qui est le facteur principal de complexité de `SampLinUCB` (du fait des inversions régulières de matrices  $d \times d$ ). On fixe  $d = 30$  pour nos expérimentations. Nous reportons en figure 2.4 un échantillon des résultats expérimentaux obtenus sur un jeu de données et une tâche similaires à ce que l'on a considéré en section 2.1.2 pour l'approche `CUCBV`.

En plus des approches de bandit stochastique détaillées en section 2.1.2, la figure 2.4 propose des résultats pour deux baselines de bandit contextuel appliquées à notre contexte, confrontées au cas de délivrance d'échantillons 3, dans lequel les échantillons ne sont donnés que pour les utilisateurs écoutés :

- `LinUCB` : application naïve d'un algorithme de bandit linéaire contextuel classique, dans lequel on exploite directement les échantillons observés comme contextes. À chaque pas de temps, des vecteurs nuls sont donnés comme contexte aux utilisateurs non sélectionnés au pas de temps précédent (donc pour lesquels on n'a pas observé d'échantillon sur la période d'écoute) ;

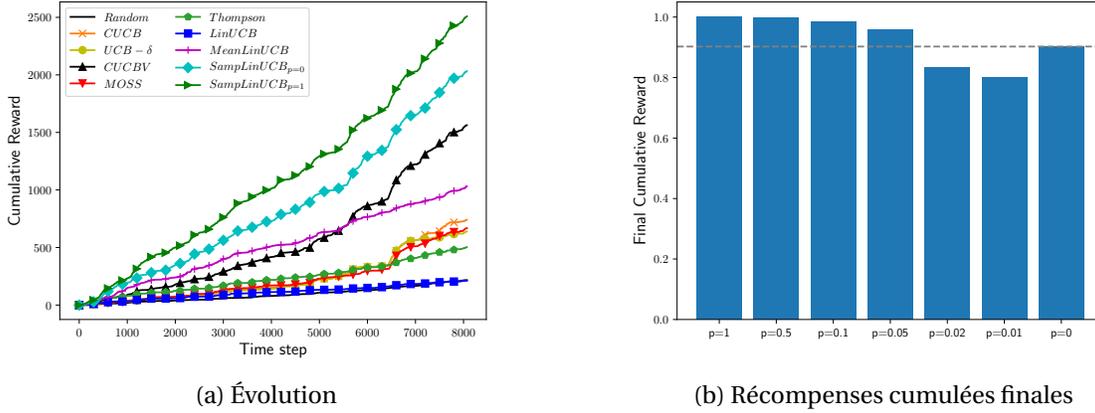
- **MeanLinUCB** : cette baseline correspond à notre approche *SampLinUCB* mais sans le terme d'exploration sur les estimations de profil (i.e.,  $\rho_{i,t,\delta}$  est fixé à 0 pour tout  $i$  et tout  $t$ ). Les estimations courantes sont alors considérées comme de vrais profils à chaque étape du processus. Une telle baseline ne peut garantir un regret sous-linéaire. Cependant, son évaluation empirique est utile pour comprendre les bénéfices de l'approche *SampLinUCB*.

Plusieurs instances de *SampLinUCB* sont considérées. La figure dénote *SampLinUCB*<sub>p=<p></sub> une instance de l'approche dans laquelle chaque utilisateur a une probabilité  $p$  d'obtenir des échantillons à chaque pas de temps. Alors que  $p = 1$  correspond au cas 1 de délivrance d'échantillons, on note  $p = 0$  l'instance du cas 3 qui nous intéresse plus particulièrement, pour laquelle on ne dispose que des contenus postés par les utilisateurs sélectionnés. Notons que, par souci de clarté et d'analyse, nous ne considérons pas ces contenus d'utilisateurs sélectionnés pour les estimations de profils pour les instances avec  $p > 0$  (ce qui correspond alors exactement au cas 2 décrit plus haut pour  $p \in ]0; 1[$ ). On fait remarquer que seul le cas 3 (i.e.,  $p = 0$ ) nous intéresse réellement pour notre tâche de collecte (ou une hybridation des cas 3 et 2 en alliant flux provenant de l'API *Filter streaming* pour le cas 3 et de l'API *Sample streaming* pour le cas 2, mais avec un  $p$  très faible en pratique du fait de la limitation à 1% de l'activité et du nombre d'utilisateurs total), les autres instances n'étant considérées que pour analyse.

La figure 2.4a trace les courbes de l'évolution des récompenses cumulées pour les différentes approches. Une observation importante est la dominance des approches *SampLinUCB*, dont les performances dépassent celles de toutes les autres politiques. La baseline *LinUCB*, qui se base directement sur les contenus postés à la période d'écoute passée pour définir ses contextes, obtient de manière non surprenante des résultats proches de *Random*, les contextes obtenus étant trop bruités et trop peu nombreux (majorité de vecteurs nuls). La baseline *MeanLinUCB* qui considère directement les moyennes empiriques courantes comme vrais profils ne permet pas de collecter plus efficacement qu'une approche comme *CUCBV* qui n'utilise pourtant aucune information supplémentaire que les récompenses observées. Cela montre le rôle crucial du terme d'exploration pour la découverte de profils  $\rho$ , qui permet à *SampLinUCB*<sub>p=0</sub> d'exploiter les contenus observés, pourtant en quantité similaire, bien plus efficacement. *SampLinUCB*<sub>p=0</sub>, bien que ne disposant que des contenus des utilisateurs sélectionnés, suit une tendance proche de *SampLinUCB*<sub>p=1</sub> qui a accès à la totalité des contenus à chaque instant. Cela démontre la bonne capacité de l'algorithme à explorer activement l'espace. Cela est confirmé par la figure 2.4b qui donne les récompenses cumulées finales relatives pour les différentes instances de *SampLinUCB* et montre que *SampLinUCB*<sub>p=0</sub> est capable de collecter quasiment aussi efficacement que *SampLinUCB*<sub>p=0.05</sub>, qui pourtant observe un nombre bien plus important de contenus (250 sur 5000 à chaque étape pour *SampLinUCB*<sub>p=0.05</sub> contre seulement 100 pour *SampLinUCB*<sub>p=0</sub>). Alors que les instances avec  $p > 0$  sont favorisées par le fait qu'elles n'ont pas besoin de sélectionner les utilisateurs pour en acquérir tout de même des échantillons, *SampLinUCB*<sub>p=0</sub> est actif non seulement sur l'extraction des paramètres de prédiction  $\beta$ , mais aussi sur le choix des profils à découvrir, ce qui lui permet de concentrer ses choix d'échantillons sur les utilisateurs prometteurs.

### Publications associées :

- Lamprier, S., Gisselbrecht, T., and Gallinari, P. (2018). Profile-based bandit with unknown profiles. *Journal of Machine Learning Research*, 19:53:1–53:40
- Gisselbrecht, T., Lamprier, S., and Gallinari, P. (2016d). Linear bandits in unknown environments. In *Machine Learning and Knowledge Discovery in Databases - European Conference*,


 FIGURE 2.4 – Performances de *SampLinUCB*.

*ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II, pages 282–298*

### 2.2.3 Collecte dynamique contextuelle

Dans les sections précédentes, nous faisons l'hypothèse que les utilisateurs sont associés à des distributions de récompense de moyenne constante. Quel que soit le type de fonction de récompense considéré, cette hypothèse est d'évidence très forte, car les utilisateurs peuvent changer de sujet d'intérêt et les dynamiques du réseau évoluer. Dans cette section, on émet donc une hypothèse différente, qui considère que l'activité d'un utilisateur durant une période donnée peut permettre d'anticiper son activité pour la période suivante. À chaque étape  $t$  du processus, un vecteur de contexte  $x_{i,t} \in \mathbb{R}^d$  est associé à chaque utilisateur  $i \in \mathcal{K}$ , avec  $x_{i,t}$  une représentation de son activité (i.e., des messages qu'il a postés) sur la période  $[t-1; t]$ . Nous considérons encore une relation linéaire entre contexte et espérance de récompense, mais cette fois avec un contexte variable pour chaque utilisateur  $i$ , plutôt qu'un profil fixe  $\mu_i$  comme c'était le cas dans la section précédente. Formellement, on suppose :

$$\exists \beta \in \mathbb{R}^d \text{ tel que } \forall t > 0 \forall i \in \mathcal{K} : \mathbb{E}[r_{i,t} | x_{i,t}] = x_{i,t}^\top \beta \quad (2.25)$$

Lorsque les contextes  $x_{i,t}$  sont disponibles pour tous les agents  $i \in \mathcal{K}$ , ils peuvent donc être exploités pour anticiper les variations d'utilité des utilisateurs. Cependant dans notre cas de collecte sur les réseaux, il n'est bien sûr pas possible d'obtenir ces contextes pour tous les utilisateurs à chaque instant  $t$ , car cela supposerait d'avoir un accès total aux messages postés sur le réseau. Comme dans la section précédente, on considère que les contextes sont disponibles uniquement pour un sous-ensemble d'utilisateurs  $\mathcal{O}_t \subset \mathcal{K}$  à chaque instant  $t$ . Dans ce cas, des suppositions doivent être réalisées sur les contextes restant cachés de l'agent. L'idée est alors de sélectionner les utilisateurs à écouter  $\mathcal{K}_t$  à chaque étape  $t$  selon :

$$\mathcal{K}_t = \arg \max_{\substack{\mathcal{K}' \subset \mathcal{K}, \\ |\mathcal{K}'| = k}} \sum_{i \in \mathcal{K}'} \max_{(\hat{\beta}, \hat{x}_{i,t}) \in \mathcal{C}_t \times \mathcal{X}_{i,t}} \mathbb{E}[r_{i,t} | \hat{x}_{i,t}, \hat{\beta}] \quad (2.26)$$

où  $C_t$  correspond à une ellipsoïde de confiance pour  $\beta$  et  $\mathcal{X}_{i,t}$  est soit une ellipsoïde de confiance pour le contexte moyen de l'utilisateur  $i$  au temps  $t$  si  $i \notin \mathcal{O}_t$ , soit est égal à  $\{x_{i,t}\}$  si  $i \in \mathcal{O}_t$ . L'idée est de définir une politique hybride qui est capable d'exploiter les contextes disponibles pour capturer des variations d'utilité espérée, tout en permettant des choix parmi les utilisateurs sans contextes en se basant sur leur profil moyen.

Contrairement à la section précédente, nous allons nous appuyer sur une formulation bayésienne, équivalente, du problème d'apprentissage de  $\beta$ . Si on s'intéresse au cas où tous les contextes sont observés à chaque pas de temps  $t$  (i.e.,  $\mathcal{O}_t = \mathcal{K}$ ), cela nous ramène à un problème classique de bandit contextuel, dans lequel on peut supposer un prior gaussien  $\mathcal{N}(m_0, v_0^2 \mathbf{I})$  pour  $\beta$ , avec  $m_0$  un vecteur de taille  $d$  et  $v_0$  un paramètre de variance scalaire, ainsi qu'une vraisemblance gaussienne des récompenses observées  $\mathcal{N}(x_{i,t}^\top \beta, \sigma_i^2)$ , avec  $\sigma_i^2$  la variance de  $r_{i,t}$ . Pour simplifier les notations, on considère dans la suite que  $m_0 = (0)_{1..d}$ ,  $v_0 = 1$  et  $\sigma_i = 1$  pour tout bras  $i$ . Dans ce cas, selon (2.12), on en arrive à une distribution postérieure  $p(\beta | \mathcal{H}_{t-1}) = \mathcal{N}(V_{t-1}^{-1} b_{t-1}, V_{t-1}^{-1})$ , avec  $\mathcal{H}_{t-1} = (\mathcal{F}_{i,t-1}, c_{i,t-1}, D_{i,t-1})_{i \in \mathcal{K}}$  l'historique d'observations au temps  $t$ , où  $\mathcal{F}_{i,t-1} = \{s : s \leq t-1, i \in \mathcal{K}_s\}$  est l'ensemble des itérations auxquelles  $i$  a été sélectionné,  $c_{i,t-1} = (r_{i,s})_{s \in \mathcal{F}_{i,t-1}}$  est le vecteur de récompenses pour  $i$  à ces itérations et  $D_{i,t-1} = (x_{i,s}^\top)_{s \in \mathcal{F}_{i,t-1}}$  la matrice  $N_{i,t-1} \times d$  de contextes associés.

Puisque  $\mathbb{E}[r_{i,t} | x_{i,t}] = x_{i,t}^\top \beta$ , on a alors  $\mathbb{E}[r_{i,t} | x_{i,t}] \sim \mathcal{N}(x_{i,t}^\top \hat{\beta}_{t-1}, x_{i,t}^\top V_{t-1}^{-1} x_{i,t})$ . Soit  $\alpha = \Phi^{-1}(1 - \delta/2)$ , où  $\Phi^{-1}$  correspond à la réciproque de la fonction de répartition de la loi normale, on en déduit facilement que pour tout  $\delta \in ]0; 1[$ , pour tout  $t > 0$  et tout  $i \in \mathcal{O}_t$  :

$$\mathbb{P}\left(|\mathbb{E}[r_{i,t} | x_{i,t}] - x_{i,t}^\top \hat{\beta}_{t-1}| \leq \alpha \sqrt{x_{i,t}^\top V_{t-1}^{-1} x_{i,t}}\right) \geq 1 - \delta \quad (2.27)$$

On peut alors utiliser le score de sélection LinUCB classique :  $s_{i,t} = x_{i,t}^\top \hat{\beta}_{t-1} + \alpha \sqrt{x_{i,t}^\top V_{t-1}^{-1} x_{i,t}}$  dans ce cas. Jusque là rien de nouveau. Cependant, si  $\mathcal{O}_t \neq \mathcal{K}$ , alors certains contextes sont cachés à chaque étape  $t$ . On ne peut alors pas directement appliquer un algorithme de bandit contextuel classique pour deux raisons :

1. Les scores  $s_{i,t}$  pour les utilisateurs  $i \notin \mathcal{O}_t$  ne peuvent pas être calculés car leur contexte  $x_{i,t}$  est inconnu. Une approche naïve serait d'éliminer ces utilisateurs des choix possibles à chaque étape  $t$ , mais à certaines étapes cela peut revenir à écarter des utilisateurs optimaux. Selon le processus de délivrance des contextes, cela peut également empêcher toute exploration : par exemple pour le cas d'intérêt  $\mathcal{O}_t = \mathcal{K}_{t-1}$  (cas 3 de la section précédente), aucun changement d'utilisateur suivi ne pourrait être opéré d'une itération à l'autre. Il faut donc être à même de donner des scores de sélection aux utilisateurs pour lesquels le contexte est caché.
2. Seules les récompenses  $r_{i,t}$  collectées pour des utilisateurs  $i$  dont on connaît le contexte au temps  $t$  peuvent être directement utilisées pour l'apprentissage de  $\beta$ . Autrement dit, on ne peut apprendre à chaque  $t$  qu'à partir des observations des utilisateurs dans  $\mathcal{K}_t \cap \mathcal{O}_t$ . Or, rien ne garantit que cette intersection soit non vide. Afin d'assurer un apprentissage efficace, la méthode doit pouvoir extraire de l'information à chaque étape  $t$ , pour chaque récompense collectée, même si le contexte correspondant est absent.

Pour répondre à ces deux problèmes, nous établissons en section 2.2.3.1 un modèle probabiliste des données, permettant l'inférence des contextes cachés. La section 2.2.3.2 présente ensuite notre algorithme HiddenLinUCB, qui s'appuie sur ce modèle pour une sélection efficace des utilisateurs à suivre à chaque étape, tenant compte de leurs variations d'utilité espérée.

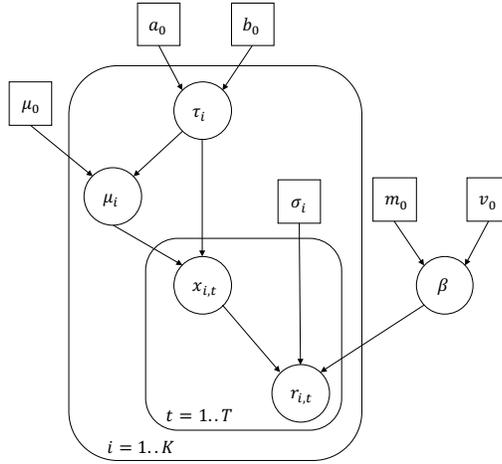


FIGURE 2.5 – Représentation graphique du modèle génératif considéré.

### 2.2.3.1 Modèle génératif et inférence variationnelle

La figure 2.5 donne une représentation graphique du modèle génératif probabiliste que l'on considère dans la suite, dans laquelle les cercles correspondent aux variables aléatoires et les carrés aux hyper-paramètres du modèle (*Plate Diagram*). Les flèches correspondent aux dépendances entre variables. On y retrouve la dépendance des récompenses selon les contextes et le paramètre  $\beta$ , décrite ci-dessus. On y suppose également des dépendances probabilistes pour les contextes  $x_{i,t}$  en fonction d'un profil  $\mu_i$  et d'une précision (i.e., variance inverse)  $\tau_i$ , telles que pour tout  $i \in \mathcal{K}$  et  $t > 0$  :  $x_{i,t} \sim \mathcal{N}(\mu_i, \tau_i^{-1}\mathbf{I})$ , avec  $\mu_i$  un vecteur de taille  $d$  de prior gaussien  $\mathcal{N}(\mu_0, \tau_i^{-1}\mathbf{I})$ , où  $\mu_0$  est un paramètre de contexte moyen pour toutes les actions, et  $\tau_i$  une variable scalaire issue d'une distribution  $\text{Gamma}(a_0, b_0)$ , où  $a_0$  et  $b_0$  sont deux paramètres de prior pour  $\tau_i$ . Pour simplifier, on considère dans la suite  $\sigma_i = 1$  pour tout  $i$ ,  $m_0 = (0)_{1..d}$ ,  $v_0 = 1$  et  $\mu_0 = (0)_{1..d}$ .

Le modèle décrit donc non seulement le mode de génération des récompenses et du paramètre  $\beta$ , mais aussi des contextes des utilisateurs à chaque instant. Notons que l'on a choisi d'utiliser des distributions de contexte sphériques (i.e., avec matrice de précision scalaire). Nous aurions également pu considérer des matrices de covariance diagonales, avec possiblement des valeurs différentes sur la diagonale (impliquant une distribution Gamma différente pour chaque composante), ou même des matrices pleines avec co-variances non-nulles entre dimensions des contextes (requérant alors l'utilisation d'une distribution normal-Wishart comme prior de précision), mais ceci au prix d'une complexité accrue.

À chaque instant  $t$  du processus, les observations dont on dispose sont les récompenses  $(r_{i,s})_{i \in \mathcal{K}, s=1..t-1}$  et les contextes  $(x_{i,s})_{i \in \mathcal{O}, s=1..t-1}$ . Les variables cachées qui nous intéressent, pour lesquelles nous cherchons à inférer les distributions bayésiennes postérieures, sont alors  $\beta$ ,  $(\mu_i)_{i=1..K}$ ,  $(\tau_i)_{i=1..K}$  et les contextes non-observés associés à des récompenses observées  $(x_{i,s})_{i \in \mathcal{K}, s \in \bar{\mathcal{O}}, s=1..t-1}$ . Notons que les récompenses correspondant à des contextes observés sont ignorées, car elles n'impactent pas les distributions utiles à la dérivation des bornes supérieures de récompense espérée à chaque temps  $t$  (i.e., les postérieures pour  $\beta$  et  $(\mu_i)_{i=1..K}$ , voir section 2.2.3.2). Ceci nous amène à considérer la distribution jointe conditionnelle suivante dans notre problème d'inférence :

$$p(\beta, (\mu_i)_{i=1..K}, (\tau_i)_{i=1..K}, (x_{i,s})_{i=1..K, s \in \mathcal{B}_{i,t-1}} | (r_{i,s})_{i=1..K, s \in \mathcal{A}_{i,t-1} \cup \mathcal{B}_{i,t-1}}, (x_{i,s})_{i=1..K, s \in \mathcal{C}_{i,t-1}}) \quad (2.28)$$

où, pour chaque utilisateur  $i$  au temps  $t$ ,  $\mathcal{A}_{i,t-1} = \{s: 1 \leq s \leq t-1 \wedge i \in \mathcal{K}_s \cap \mathcal{O}_s\}$ ,  $\mathcal{B}_{i,t-1} = \{s: 1 \leq s \leq t-1 \wedge i \in \mathcal{K}_s \cap \bar{\mathcal{O}}_s\}$  et  $\mathcal{C}_{i,t-1} = \{s: 1 \leq s \leq t-1 \wedge i \in \mathcal{O}_s\}$  correspondent aux ensembles d'itérations antérieures à  $t$  telles que, respectivement,  $i$  a été sélectionné en connaissant son contexte,  $i$  a été sélectionné sans connaître son contexte, le contexte de  $i$  a été observé.

Calculer de manière exacte cette distribution est cependant inenvisageable d'un point de vue complexité algorithmique. Par ailleurs, aucune forme close ne peut de toutes façons être obtenue. Dans la suite, nous proposons de nous appuyer sur une approche d'approximation variationnelle [Bishop, 2006]. L'inférence variationnelle (VI) regroupe les approches basées sur le calcul des variations pour approximer des distributions de probabilité avec dépendances complexes. Elle se pose en alternative efficace aux méthodes MCMC (*Monte Carlo Markov Chains*), tel que l'échantillonnage de *Gibbs*. Alors que les méthodes MCMC fournissent des approximations numériques via échantillonnages successifs de la vraie distribution jointe, l'inférence variationnelle détermine une approximation analytique, localement optimale de cette distribution. En pratique l'utilisation d'approches MCMC pour notre tâche de bandit avec contextes cachés induirait des calculs trop lourds à chaque étape, et probablement une convergence trop lente et incertaine du fait de la variance liée à l'échantillonnage.

L'idée de l'inférence variationnelle est d'approximer la distribution cible par une distribution plus simple, en relaxant des dépendances entre variables. Soient  $X$  un ensemble de variables observées et  $Z$  un ensemble de variables cachées. Soit la distribution postérieure  $p(Z|X)$  que l'on cherche à approximer via la distribution variationnelle  $q(Z)$ . L'objectif est d'amener  $q$  au plus proche de la distribution d'intérêt  $p$ , au sens d'une mesure de divergence entre distributions. Nous considérons classiquement la divergence de Kullback-Leibler (KL) de  $q$  par rapport à  $p$  :

$$\text{KL}_X(q||p) = \int q(Z) \log \left( \frac{q(Z)}{p(Z|X)} \right) dZ \quad (2.29)$$

Notons que, comme discuté dans [Winn and Bishop, 2005], cette utilisation "exclusive" de la KL conduit à l'obtention d'une distribution  $q$  incluse dans  $p$ . Cela mène à ignorer des modes de  $p$ , mais exclut l'attribution de masse de probabilité à des régions de faible densité selon  $p$ , contrairement à ce qu'une minimisation d'une version "inclusive" (i.e.,  $\text{KL}(p||q)$ ) fournirait.

L'équation (2.29) peut être ré-écrite de la manière suivante pour faire apparaître l'évidence des données observées selon  $p$  (voir par exemple dans [Winn and Bishop, 2005]) :

$$\log p(X) = \text{KL}_X(q||p) + \mathcal{L}_X(q) \text{ avec } \mathcal{L}_X(q) \triangleq \int q(Z) \log \left( \frac{p(X,Z)}{q(Z)} \right) dZ \quad (2.30)$$

où, étant donné que la divergence KL est une quantité positive,  $\mathcal{L}_X(q)$  correspond alors à une borne inférieure du logarithme de l'évidence  $p(X)$ , ce qui lui vaut le nom de *ELBO* (pour *Evidence Lower Bound*). Puisque la quantité  $\log p(X)$  est indépendante de  $q$ , maximiser  $\mathcal{L}_X(q)$  revient à minimiser la divergence  $\text{KL}_X(q||p)$ , ce qui permet d'obtenir une approximation de la distribution conditionnelle  $p(Z|X)$ . Pour notre problème, nous proposons de considérer une distribution variationnelle  $q$  suivant la classique approximation des champs moyens, qui se factorise selon des partitions de  $Z$  supposées indépendantes dans  $q$  :

$$q(\beta, (\mu_i)_{i=1..K}, (\tau_i)_{i=1..K}, (x_{i,s})_{i=1..K, s \in \mathcal{B}_{i,t-1}}) = q_\beta(\beta) \prod_{i=1}^K \left( q_{\mu_i}(\mu_i) q_{\tau_i}(\tau_i) \prod_{s \in \mathcal{B}_{i,t-1}} q_{x_{i,s}}(x_{i,s}) \right) \quad (2.31)$$

où  $q_\beta$ ,  $q_{\mu_i}$ ,  $q_{\tau_i}$  et  $q_{x_{i,s}}$  correspondent à des distributions spécifiques pour chaque facteur indépendant de  $q$ . L'objectif est alors de trouver les distributions optimales pour tous ces facteurs, telles que  $\mathcal{L}_X(q)$

soit maximisé. L'idée est de faire tendre la distribution jointe  $q$  vers la probabilité conditionnelle de (2.28), afin d'obtenir des intervalles de confiance pour  $\beta$  et  $\mu_i$  pour tout  $i$ , sur lesquels baser notre politique de bandit.

Selon le calcul des variations, qui correspond à un ensemble de méthodes mathématiques pour la minimisation de fonctionnelles, il peut être montré que  $\mathcal{L}_X(q)$  est maximisé en prenant, pour chaque facteur indépendant  $q_j$  de  $q$ , une distribution variationnelle  $q_j^*(Z_j)$  qui satisfait :

$$q_j^*(Z_j) = \frac{e^{\mathbb{E}_{i \neq j}[\log p(Z, X)]}}{\int e^{\mathbb{E}_{i \neq j}[\log p(Z, X)]} dZ_j} \quad (2.32)$$

avec  $\mathbb{E}_{i \neq j}[\log p(Z, X)]$  l'espérance, prise sur tous les facteurs  $i$  de  $Z$  sauf  $j$  selon leurs distributions  $q_i$  respectives, du logarithme de la probabilité jointe des valeurs observées pour les variables  $X$  et l'instanciation  $Z$  des variables cachées. En pratique, il est plus simple de passer au logarithme :

$$\log q_j^*(Z_j) = \mathbb{E}_{i \neq j}[\log p(Z, X)] + \text{constante} \quad (2.33)$$

où la constante correspond au terme de normalisation de la distribution, qui ne dépend pas de la valeur de  $Z_j$ . Tous les priors de notre modèle étant les distributions conjuguées des vraisemblances correspondantes<sup>5</sup>, cela permet de dériver les propositions 1, 2, 3 et 4 (dont les preuves sont données dans [Lamprier et al., 2019]), qui déterminent respectivement les distributions pour  $\beta$ ,  $x_{i,s}$ ,  $\mu_i$  et  $\tau_i$ .

**Proposition 1** À l'étape  $t$ , la distribution  $q_\beta^*$  est une gaussienne multivariée  $\mathcal{N}(\hat{\beta}_{t-1}, V_{t-1}^{-1})$ , avec :

$$\hat{\beta}_{t-1} = V_{t-1}^{-1} \left( \sum_{i=1}^K \left[ \sum_{s \in \mathcal{A}_{i,t-1}} x_{i,s} r_{i,s} + \sum_{s \in \mathcal{B}_{i,t-1}} \mathbb{E}[x_{i,s}] r_{i,s} \right] \right), \quad V_{t-1} = I + \sum_{i=1}^K \left[ \sum_{s \in \mathcal{A}_{i,t-1}} x_{i,s} x_{i,s}^\top + \sum_{s \in \mathcal{B}_{i,t-1}} \mathbb{E}[x_{i,s} x_{i,s}^\top] \right]$$

Où  $\mathbb{E}[x_{i,s} x_{i,s}^\top] = \text{Cov}(x_{i,s}) + \mathbb{E}[x_{i,s}] \mathbb{E}[x_{i,s}]^\top$ , avec  $\mathbb{E}[x_{i,s}]$  et  $\text{Cov}(x_{i,s})$ , respectivement l'espérance et la covariance de  $x_{i,s}$ , données en proposition 2.

**Proposition 2** À l'étape  $t$ , pour tout  $i \in \{1, \dots, K\}$  et tout  $s \in \mathcal{B}_{i,t-1}$ , la distribution optimale  $q_{x_{i,s}}^*$  est une gaussienne multivariée  $\mathcal{N}(\hat{x}_{i,s}, W_{i,s}^{-1})$ , avec :

$$\hat{x}_{i,s} = W_{i,s}^{-1} (\mathbb{E}[\beta] r_{i,s} + \mathbb{E}[\tau_i] \mathbb{E}[\mu_i]), \quad W_{i,s} = \mathbb{E}[\beta \beta^\top] + \mathbb{E}[\tau_i] I$$

Où  $\mathbb{E}[\beta \beta^\top] = \text{Cov}(\beta) + \mathbb{E}[\beta] \mathbb{E}[\beta]^\top$ , avec  $\mathbb{E}[\beta]$  et  $\text{Cov}(\beta)$ , respectivement l'espérance et la covariance de  $\beta$ , données en proposition 1.  $\mathbb{E}[\tau_i]$  et  $\mathbb{E}[\mu_i]$  correspondent respectivement à l'espérance de  $\tau_i$ , donnée en proposition 4<sup>6</sup>, et l'espérance de  $\mu_i$ , donnée en proposition 3, pour toute action  $i$ .

**Proposition 3** À l'étape  $t$ , pour tout  $i \in \{1, \dots, K\}$ , la distribution optimale  $q_{\mu_i}^*$  est une gaussienne multivariée  $\mathcal{N}(\hat{\mu}_{i,t-1}, \Sigma_{i,t-1}^{-1})$ , avec :

$$\hat{\mu}_{i,t-1} = \frac{\sum_{s \in \mathcal{C}_{i,t-1}} x_{i,s} + \sum_{s \in \mathcal{B}_{i,t-1}} \mathbb{E}[x_{i,s}]}{1 + n_{i,t-1}}, \quad \Sigma_{i,t-1} = (1 + n_{i,t-1}) \mathbb{E}[\tau_i] I$$

5. En théorie bayésienne, si la distribution postérieure d'une variable aléatoire est de la même famille que son prior, on dit que le prior est le prior conjugué de la vraisemblance considérée. La page *Wikipedia* [https://en.wikipedia.org/wiki/Conjugate\\_prior](https://en.wikipedia.org/wiki/Conjugate_prior) recense un grand nombre de distributions de vraisemblances avec leur prior conjugué.

6.  $q_{\tau_i}^*$  étant une distribution Gamma de forme  $a_{i,t-1}$  et d'échelle  $b_{i,t-1}$ , son espérance  $\mathbb{E}[\tau_i]$  vaut  $\frac{a_{i,t-1}}{b_{i,t-1}}$ .

Avec  $n_{i,t-1} = |\mathcal{B}_{i,t-1}| + |\mathcal{C}_{i,t-1}|$  correspondant au nombre d'itérations antérieures à  $t$  telles que  $i$  a été sélectionné ou son contexte a été observé.  $\mathbb{E}[\tau_i]$  et  $\mathbb{E}[x_{i,s}]$  correspondent respectivement à l'espérance de  $\tau_i$ , donnée en proposition 4, et à l'espérance de  $x_{i,s}$ , donnée en proposition 2, pour toute action  $i$  et pas de temps  $s < t$ .

**Proposition 4** À l'étape  $t$ , pour tout  $i \in \{1, \dots, K\}$ , la distribution optimale  $q_{\tau_i}^*$  est une distribution Gamma de forme  $a_{i,t-1}$  et d'échelle  $b_{i,t-1}$ , avec :

$$a_{i,t-1} = a_0 + \frac{d(1 + n_{i,t-1})}{2}, \quad b_{i,t-1} = b_0 + \frac{1}{2}(1 + n_{i,t-1})\mathbb{E}[\mu_i^\top \mu_i] - \mathbb{E}[\mu_i]^\top \left( \sum_{s \in \mathcal{C}_{i,t-1}} x_{i,s} + \sum_{s \in \mathcal{B}_{i,t-1}} \mathbb{E}[x_{i,s}] \right) + \frac{1}{2} \sum_{s \in \mathcal{C}_{i,t-1}} x_{i,s}^\top x_{i,s} + \frac{1}{2} \sum_{s \in \mathcal{B}_{i,t-1}} \mathbb{E}[x_{i,s}^\top x_{i,s}]$$

Où  $\mathbb{E}[\mu_i^\top \mu_i] = \text{Trace}(\text{Cov}(\mu_i)) + \mathbb{E}[\mu_i]^\top \mathbb{E}[\mu_i]$ , avec  $\mathbb{E}[\mu_i]$  et  $\text{Cov}(\mu_i)$ , respectivement l'espérance et la covariance de  $\mu_i$ , donnés en proposition 3.  $\mathbb{E}[x_{i,s}]$  correspond à l'espérance de  $x_{i,s}$ , donné en proposition 2 pour toute action  $i$  et pas de temps  $s$ .

Ces propositions définissent des dépendances circulaires entre les distributions, ce qui suggère l'emploi d'un algorithme itératif jusqu'à convergence des estimateurs. En pratique, on détermine un nombre maximal de passes d'optimisation  $nbIt$ . L'algorithme 3 décrit notre procédure d'inférence variationnelle.

---

**Algorithme 3**: Processus d'Inférence Variationnelle
 

---

```

1  for  $It = 1..nbIt$  do
2      Calcul de  $\hat{\beta}_{t-1}$  et  $V_{t-1}$  suivant la proposition 1;
3      for  $i = 1..K$  do
4          Calcul de  $\hat{\mu}_{i,t-1}$  et  $\Sigma_{i,t-1}$  suivant la proposition 3 ;
5          Calcul de  $a_{i,t-1}$  et  $b_{i,t-1}$  suivant la proposition 4 ;
6          for  $s \in \mathcal{B}_{i,t-1}$  do
7              Calcul de  $\hat{x}_{i,s}$  et  $W_{i,s}$  suivant la proposition 2 ;
8          end
9      end
10 end
    
```

---

Cet algorithme, appliqué à chaque étape  $t$  du processus de collecte, permet dans la section suivante d'établir des intervalles de confiance pour  $\beta$  et les profils  $\mu_i$ . Cependant, se pose le problème de la complexité croissante de cette procédure d'inférence variationnelle. En effet à chaque étape  $t$ , pour chaque action  $i$ , il est nécessaire d'estimer les distributions de chaque contexte caché  $x_{i,s}$  tel que  $s \in \mathcal{B}_{i,t-1}$ , ensemble qui tend à croître avec  $t$ . Selon les cas, le nombre de contextes cachés à considérer peut croître rapidement (possiblement  $k$  supplémentaires à chaque étape), ce qui rend le processus incompatible avec l'aspect en ligne des algorithmes de bandit. Cela mène à des quantités d'informations à inférer incalculables. Pour dépasser cette limitation, nous proposons de supposer que les postérieures au temps  $t$  peuvent être efficacement approximées en ne recalculant à chaque étape que les distributions des contextes cachés appartenant à des itérations d'un passé proche  $\mathcal{B}_{i,t-1} \setminus \mathcal{B}_{i,t-1-H}$ , avec  $H$  la taille de la fenêtre glissante considérée, ce qui garantit une complexité constante. Pour

chaque étape  $t > H$ , cela revient à considérer la factorisation suivante :

$$q^{(t)}(\beta, (\mu_i)_{i=1..K}, (\tau_i)_{i=1..K}, (x_{i,s})_{i=1..K, s \in \mathcal{B}_{i,t-1}}) \approx q_{\beta}^{(t)}(\beta) \times \prod_{i=1}^K \left( q_{\mu_i}^{(t)}(\mu_i) q_{\tau_i}^{(t)}(\tau_i) \prod_{s \in \mathcal{B}_{i,t-1} \setminus \mathcal{B}_{i,t-1-H}} q_{x_{i,s}}^{(t)}(x_{i,s}) \prod_{s \in \mathcal{B}_{i,t-1-H}} q_{x_{i,s}}^{(s+H)}(x_{i,s}) \right) \quad (2.34)$$

où  $q^{(t)}$  correspond à la distribution variationnelle calculée à l'étape  $t$ . Cette formulation distingue deux ensembles de contextes cachés : ceux appartenant à un passé proche (i.e.,  $s \in \mathcal{B}_{i,t-1} \setminus \mathcal{B}_{i,t-1-H}$ ) dont les distributions sont mises à jour selon les autres facteurs et les nouvelles observations, et ceux plus anciens (i.e.,  $s \in \mathcal{B}_{i,t-1-H}$ ) dont les distributions sont figées à leur état obtenu à l'étape  $s + H$ . Les distributions des contextes cachés deviennent alors constantes à partir du moment où ils sortent de la fenêtre glissante de taille  $H$ . Concrètement, cela revient dans l'algorithme 3 à limiter la boucle interne aux éléments de  $\mathcal{B}_{i,t-1} \setminus \mathcal{B}_{i,t-1-H}$ . Le biais induit par cette approximation supplémentaire est supposé diminuer avec les itérations, au fur et à mesure que de nouvelles connaissances sur les récompenses et les utilisateurs viennent rendre insignifiantes les erreurs induites par les distributions biaisés des contextes historiques.

Notons qu'une alternative aurait été de définir un processus d'inférence variationnelle stochastique (SVI) [Hoffman et al., 2012], spécifiquement adapté à l'inférence dans les flux de données. Plutôt que de réaliser l'inférence des variables globales  $\beta$ ,  $\mu_i$  et  $\tau_i$  en considérant l'ensemble des données à chaque étape  $t$ , SVI "oublie" les informations trop anciennes en fonction d'un ratio d'oubli  $\rho_t$ . Alors que les paramètres des variables locales  $x_{i,t}$  sont inférés de la même manière que dans notre proposition 2 ci dessus, SVI met à jour les distributions des variables globales en travaillant sur leur paramètre naturel en fonction des nouvelles observations. Dans le contexte d'une fenêtre glissante de taille  $H$ , cela revient à fixer  $\eta_{q_x}^t = (1 - \rho_t)\eta_{q_x}^{(t-1)} + \frac{\rho_t}{H} \sum_{s=t-H}^{t-1} \mathbb{E}_{x \setminus}[\eta_{p_x}(t, s)]$  pour toute variable globale  $x$  à chaque étape  $t$ , où  $\eta_{q_x}^t$  correspond au paramètre naturel de la distribution variationnelle de  $x$  à  $t$ ,  $\mathbb{E}_{x \setminus}$  correspond à une espérance calculée selon la distribution variationnelle de tous les facteurs sauf  $x$  et  $\eta_{p_x}(t, s)$  correspond au paramètre naturel de la distribution postérieure de  $x$  considérant  $t$  répliques des échantillons du pas de temps  $s$ . Par exemple pour  $\beta$ ,  $\eta_{p_x}(t, s)$  est donné par :

$$\eta_{p_{\beta}}(t, s) = \left( t \sum_{i \in \mathcal{K}_s} \mathbb{E}[x_{i,s}] r_{i,s}, -\frac{1}{2} (I + t \sum_{i \in \mathcal{K}_s} \mathbb{E}[x_{i,s} x_{i,s}^{\top}]) \right)$$

où  $\mathbb{E}[x_{i,s}]$  est égal à  $x_{i,s}$  si  $i \in \mathcal{O}_s$  et correspond à la moyenne de sa distribution variationnelle  $x_{i,s}$  (proposition 2) sinon. Cela garantit une convergence vers des paramètres variationnels proches de ce qui aurait été obtenu en travaillant sur l'ensemble des données historiques à chaque étape. Cependant, le ratio d'oubli (et son éventuelle évolution) est difficile à régler. Une valeur trop faible induit une convergence trop lente (mises à jour trop conservatives), alors qu'une valeur trop forte risque de provoquer des pertes d'information importantes (*catastrophic forgetting*). Des expérimentations complémentaires devraient être effectuées pour s'en assurer mais aucune amélioration significative des performances n'a été observée dans des expérimentations préliminaires utilisant ce processus de SVI. Nous nous en tenons dans les résultats présentés à la section suivante à l'algorithme 3 avec fenêtre glissante de taille  $H$ .

En utilisant l'approximation donnée par (2.34), la complexité à chaque étape  $t$  ne dépend donc plus de  $t$  : les éléments dépendant des contextes à sommer dans  $V_{t-1}$ ,  $\hat{\beta}_{t-1}$ ,  $\hat{\mu}_{i,t-1}$  et  $b_{i,t-1}$  sont tous constants, à l'exception de ceux issus des itérations des  $H$  derniers pas de temps. La complexité de

l'algorithme dépend principalement des inversions matricielles réalisées pour chaque estimation. À chaque itération du processus et chaque passe d'optimisation, on a une matrice  $d \times d$  à inverser pour  $\beta$  et une matrice  $d \times d$  à inverser pour chaque contexte caché de la fenêtre glissante. Puisqu'une inversion matricielle via la méthode d'élimination de Gauss-Jordan possède une complexité en  $n^3$  pour une matrice  $n \times n$ , la complexité de notre processus d'inférence est donc en  $\mathcal{O}(nb1t \times H(K - k + 1)d^3)$  à chaque pas de temps de la collecte. Cette complexité est largement supérieure à celle d'un *UCB* qui a seulement des mises à jour de moments à réaliser à chaque étape, mais c'est le prix d'une exploitation d'informations auxiliaires partielles observées au cours du processus. Nous verrons à la section suivante que cela se justifie au regard des performances de collecte. De plus, dans de nombreux contextes comme celui de la capture sur les réseaux sociaux, cette complexité additionnelle n'est pas limitante, une large part de l'optimisation pouvant être effectuée en tâche de fond pendant les période de capture, ce qui n'induit aucune latence pour le processus de décision. Dans les autres cas, la complexité pourrait être en outre réduite par l'utilisation de techniques de mise à jour dans des matrices inverses (e.g., formule de Sherman–Morrison). Enfin, le processus d'inférence variationnelle peut être facilement parallélisé sur des groupes d'utilisateurs, la synchronisation étant seulement requise à la fin de chaque passe d'optimisation.

### 2.2.3.2 Algorithme *HiddenLinUCB*

L'algorithme d'inférence variationnelle permet donc de définir des distributions postérieures pour  $\beta$  et  $\mu_i$  pour tout  $i$ , pour les cas où des contextes sont cachés, ce qui est toujours le cas sur les réseaux sociaux. Dans cette section, nous nous appuyons sur cette inférence pour établir une politique de bandit, nommée *HiddenLinUCB*, basée sur des intervalles de confiance des distributions variationnelles obtenues. À chaque étape  $t$ , deux cas sont possibles pour chaque utilisateur  $i$  candidat. Soit le contexte de  $i$  est disponible (i.e.,  $i \in \mathcal{O}_t$ ) et dans ce cas,  $\mathbb{E}[r_{i,t}|x_{i,t}]$  peut directement être considéré comme une variable aléatoire suivant une distribution gaussienne d'espérance  $x_{i,t}^\top \hat{\beta}_{t-1}$  et de variance  $x_{i,t}^\top V_{t-1}^{-1} x_{i,t}$ . Il est alors possible de considérer un intervalle de confiance selon (2.27) et donc un score *LinUCB* classique associé pour  $i$ . À noter cependant que les valeurs de  $\hat{\beta}_{t-1}$  et  $V_{t-1}^{-1}$  utilisées dépendent du processus d'inférence variationnelle de la section précédente. Soit le contexte de  $i$  n'est pas visible (i.e.,  $i \notin \mathcal{O}_t$ ) et dans ce cas il n'est pas possible d'appliquer directement (2.27) car  $x_{i,t}$  est inconnu. Dans ce qui suit, nous dérivons un intervalle de confiance pour  $\mathbb{E}[r_{i,t}]$  pour ce cas. On a pour tout  $i$  et tout  $t$  :

$$\begin{aligned} \mathbb{E}[r_{i,t}] &= \int_{x_{i,t}} \mathbb{E}[r_{i,t}|x_{i,t}] p(x_{i,t}) dx_{i,t} = \int_{x_{i,t}} x_{i,t}^\top \beta p(x_{i,t}) dx_{i,t} = \beta^\top \int_{x_{i,t}} x_{i,t} p(x_{i,t}) dx_{i,t} \\ &= \beta^\top \mathbb{E}[x_{i,t}] = \beta^\top \mu_i = \beta^\top \hat{\mu}_{i,t-1} + \beta^\top (\mu_i - \hat{\mu}_{i,t-1}) \end{aligned} \quad (2.35)$$

où  $\hat{\mu}_{i,t-1}$  correspond à l'espérance de  $\mu_i$  selon la distribution postérieure de  $\mu_i$  (voir proposition 3). En s'appuyant sur le fait que les postérieures de  $\beta$  et de  $\mu_i$  sont toutes les deux des gaussiennes dont on connaît les paramètres grâce au processus d'inférence de la section précédente et en utilisant notamment l'inégalité de Boole, on peut alors montrer le théorème ci-dessous pour l'établissement d'un intervalle de confiance pour  $\mathbb{E}[x_{i,t}]$  (preuves données dans [Lamprier et al., 2019]).

**Théorème 7** Soient  $\delta_1 \in ]0; 1[$  et  $\delta_2 \in ]0; 1[$ . À chaque étape  $t > 0$  et pour chaque utilisateur  $i \in \mathcal{K}$ , on a :

$$\mathbb{P} \left( \left| \mathbb{E}[r_{i,t}] - \hat{\beta}_{t-1}^\top \hat{\mu}_{i,t-1} \right| \leq \alpha_1 \sqrt{\hat{\mu}_{i,t-1}^\top V_{t-1}^{-1} \hat{\mu}_{i,t-1}} + \alpha_2 \sqrt{\frac{b_{i,t-1}}{a_{i,t-1}(1 + n_{i,t-1})}} \right) \geq 1 - \delta_1 - \delta_2 \quad (2.36)$$

où  $\alpha_1 = \Phi^{-1}(1 - \delta_1/2)$ , avec  $\Phi^{-1}$  la réciproque de la fonction de répartition de la loi normale standard, et  $\alpha_2 = S\sqrt{\Psi^{-1}(1 - \delta_2)}$ , avec  $\Psi^{-1}$  la réciproque de la fonction de répartition de la loi  $\chi^2$  avec  $d$  degrés de liberté et  $S$  une borne supérieure pour  $\|\beta\|$ .

L'inégalité du théorème 7, dont les éléments  $\hat{\mu}_{i,t-1}$ ,  $V_{t-1}$ ,  $\hat{\beta}_{t-1}$ ,  $a_{i,t-1}$  et  $b_{i,t-1}$  sont issus du processus d'inférence donné par l'algorithme 3, donne une borne supérieure de l'espérance de récompense en absence de contexte observé. Nous utilisons cette borne pour définir le score de sélection de tout  $i \notin \mathcal{O}_t$  à chaque instant  $t$  du processus :

$$s_{i,t} = \hat{\mu}_{i,t-1}^\top \hat{\beta}_{t-1} + \alpha_1 \sqrt{\hat{\mu}_{i,t-1}^\top V_{t-1}^{-1} \hat{\mu}_{i,t-1}} + \alpha_2 \sqrt{\frac{b_{i,t-1}}{a_{i,t-1}(1 + n_{i,t-1})}} \quad (2.37)$$

Notons que la partie de gauche de ce score est similaire à un score `LinUCB` dans lequel on utiliserait  $\hat{\mu}_{i,t-1}$  plutôt que  $x_{i,t}$ . Intuitivement, cela revient à approximer  $x_{i,t}$  par sa moyenne empirique sur les échantillons observés pour les itérations  $t$  telles que  $i \in \mathcal{O}_t$ . L'utilisation d'un estimateur plutôt que la vraie espérance  $\mu_i$  induit une incertitude additionnelle, qui est prise en compte par le second terme d'exploration du score proposé. Pour mettre en place un algorithme équitable, il faut que les intervalles de confiance utilisés pour le score de sélection dans les cas observés (2.27) et non-observés (2.36) soient de même niveau. Cela implique alors de fixer  $\delta$ ,  $\delta_1$  et  $\delta_2$  de manière à ce que  $\delta = \delta_1 + \delta_2$ . Dans nos expérimentations nous utilisons  $\delta_1 = \delta_2 = \delta/2$ , puisqu'il n'y a pas de raison particulière à favoriser l'exploration vis à vis du paramètre  $\beta$  par rapport à celle sur  $\mu$  et inversement.

Du fait des approximations variationnelles effectuées, il n'est pas possible de donner une borne supérieure du regret comme c'était le cas dans la section précédente. Cependant, les termes d'exploration définis garantissent de ne pas s'enfermer dans des configurations de paramètres sous-optimales, puisque menant à reconsidérer régulièrement les estimations réalisées. Selon le processus de délivrance des contextes, et particulièrement pour le cas qui nous intéresse où  $\mathcal{O}_t = \mathcal{K}_{t-1}$ , le terme d'exploration additionnel est crucial pour ne pas écarter des utilisateurs à fort potentiel dont les contextes observés dans les premiers pas de temps de la collecte n'étaient pas bons. Notons que ce terme décroît avec  $1/\sqrt{n_{i,t-1}}$ . Au bout d'un certain nombre d'étapes avec un nombre fini d'utilisateurs, il est alors garanti de converger vers un score classique de `LinUCB` qui s'appuie sur  $\hat{\mu}_{i,t-1}$  et dont l'estimateur de  $\beta$  est obtenu par inférence variationnelle, tenant à la fois compte des contextes observés et cachés de l'historique de la collecte.

**Résultats** La figure 2.7 à gauche donne les résultats comparés, en terme de récompenses cumulées finales, pour les différentes approches abordées jusqu'ici pour notre tâche de collecte, sur une base hors-ligne et une tâche similaire aux expérimentations reportées en figure 2.4. `resHiddenLinUCB $\mathcal{K}_{t-1}$`  correspond au cas où  $\mathcal{O}_t = \mathcal{K}_{t-1}$ , les versions `resHiddenLinUCB $_p$`   $= < p >$  correspondent, comme dans la section précédente, à des instances où à chaque temps  $t$ , chaque utilisateur possède une probabilité  $p$  de figurer dans  $\mathcal{O}_t$ . On observe l'efficacité de l'approche `HiddenLinUCB`, qui permet d'améliorer significativement les performances de collecte, y compris par rapport à `SamplLinUCB`, grâce à sa capacité d'adaptation aux variations d'utilité. On note comme pour `SamplLinUCB` les bonnes performances de l'approche même dans le cas où l'exploration des contextes est liée au processus de sélection (i.e., quand  $\mathcal{O}_t = \mathcal{K}_{t-1}$ ), `resHiddenLinUCB $\mathcal{K}_{t-1}$`  obtenant des résultats supérieurs à certaines instances avec délivrance de contextes indépendante. Notons par ailleurs que nous observons un bon taux de renouvellement des utilisateurs sélectionnés, avec une moyenne d'environ 45% d'utilisateurs sélectionnés en connaissant leur contexte à chaque étape. L'équilibre entre les deux

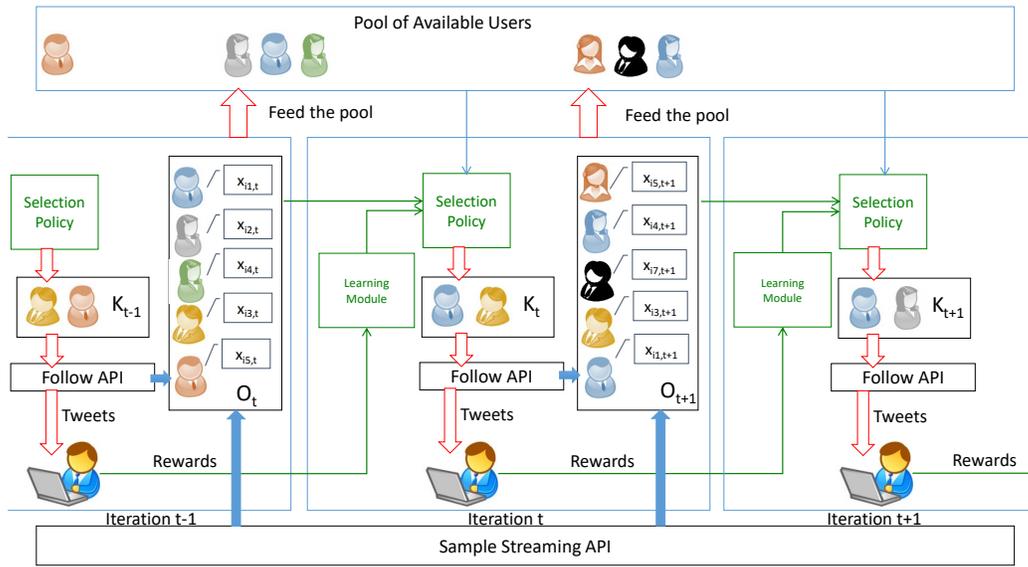


FIGURE 2.6 – Processus de collecte avec observation de contextes simultanément à partir de l’API *Follow Streaming* et de l’API *Sample Streaming*.

scores de sélection considérés semble donc bien assuré, l’algorithme étant capable de se fixer sur des utilisateurs utiles sur une période donnée en fonction de leur contexte, tout en réservant une partie de ses capacités d’écoute pour l’exploration. La figure 2.7 à droite donne l’évolution des récompenses cumulées en situation réelle de collecte sur Twitter, où ici les contextes observés sont issus à la fois des utilisateurs écoutés sur la période précédente (i.e., API *Follow Streaming* de Twitter) et de messages collectés aléatoirement sur l’ensemble du réseau (i.e., API *Sample Streaming* de Twitter), comme décrit dans l’illustration de l’architecture générale de l’approche de collecte donnée en figure 2.6.

**Publications associées :**

- Lamprier, S., Gisselbrecht, T., and Gallinari, P. (2019). Contextual bandits with hidden contexts: a focused data capture from social media streams. *Data Min. Knowl. Discov.*, 33(6):1853–1893

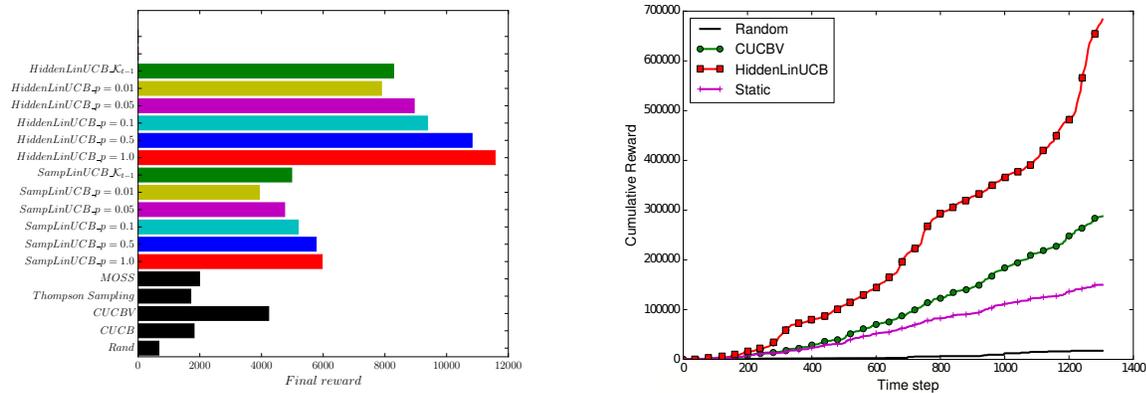


FIGURE 2.7 – Performances de *HiddenLinUCB*.

- Gisselbrecht, T., Lamprier, S., and Gallinari, P. (2016c). Dynamic data capture from social media streams: A contextual bandit approach. In *Proceedings of the Tenth International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016.*, pages 131–140
- Gisselbrecht, T., Lamprier, S., and Gallinari, P. (2016b). Collecte ciblée à partir de flux de données en ligne dans les médias sociaux: une approche de bandit contextuel. *Document Numérique*, 19(2-3):11–30
- Gisselbrecht, T., Lamprier, S., and Gallinari, P. (2016a). Bandit contextuel pour la capture de données temps réel sur les médias sociaux. In *CORIA 2016 - Conférence en Recherche d'Informations et Applications- 13th French Information Retrieval Conference. CIFED 2016 Colloque International Francophone sur l'Ecrit et le Document, Toulouse, France, March 9-11, 2016, Toulouse, France, March 9-11, 2016.*, pages 57–72

#### 2.2.4 Collecte dynamique relationnelle

Dans la section précédente, nous avons proposé une approche pour anticiper efficacement les variations d'utilité des utilisateurs des réseaux, par une approche basée sur les contenus observés à chaque période d'écoute. Cependant, le fait qu'une majorité des contextes ne soit pas accessible en raison des restrictions imposées par les API de Twitter, constitue une contrainte forte. Chaque contenu observé ne permet d'anticiper que l'utilité de son auteur sur le prochain pas de temps, ce qui impose à l'algorithme de reconsidérer fréquemment les utilisateurs individuellement pour connaître leur potentiel courant (le contenu observé à  $t$  pour  $i$  n'est exploitable que pour  $i$  à  $t + 1$ ). Une autre possibilité, qui ne nécessite pas l'utilisation d'informations auxiliaires, est de modéliser des relations de dépendance temporelle entre utilisateurs, afin de mieux capter les variations d'utilité des récompenses associées. L'idée est d'exploiter les récompenses observées sur les utilisateurs sélectionnés, pour anticiper les utilités de l'ensemble des utilisateurs sur les périodes futures.

Comme déjà évoqué en section 2.1.2, différentes approches de la littérature ont d'ores et déjà proposé des solutions pour exploiter les relations entre actions dans le cadre des problèmes de bandit [Buccapatnam et al., 2014, Cesa-Bianchi et al., 2013, Gentile et al., 2014]. Notre cadre diffère néanmoins de ces approches, car nous visons à exploiter des dépendances à la fois structurelles et temporelles, là où les approches existantes se limitent à un cadre structurel stationnaire. Notre objectif est d'appréhender la non-stationnarité des récompenses, en s'appuyant sur des relations de dépendances découvertes. Les approches de bandit non-stationnaire considèrent également que les distributions de récompense peuvent évoluer au cours du processus. *Discounted UCB*, qui utilise un facteur d'ancienneté des observations, et *Sliding Window UCB*, qui ne considère que les observations situées dans une fenêtre d'historique limitée, sont deux exemples d'approches de bandit non-stationnaires populaires [Garivier and Moulines, 2011]. Plutôt que supposer des changements abrupts de distribution comme dans [Garivier and Moulines, 2011], les *restless bandits* [Whittle, 1988] considèrent que l'état de chaque action évolue selon un processus aléatoire donné (ou *rested bandit* si seul l'état de l'action sélectionnée change à chaque itération). Suivant ce principe, [Slivkins and Upfal, 2008] suppose des mouvements browniens indépendants entre itérations successives, alors que [Ortner et al., 2014], [Audiffren and Ralaivola, 2015] ou [Tekin and Liu, 2012] considèrent qu'un processus de Markov caché dirige les évolutions des espérances de récompenses. Ces travaux diffèrent néanmoins du notre par le fait que les modèles d'évolution considèrent les actions indépendamment. L'approche présentée dans [Carpentier and Valko, 2016] propose également un cadre où des relations d'influence inter-utilisateurs sont graduellement découvertes. Cependant, notre cadre est bien différent de celui

de [Carpentier and Valko, 2016], qui s’emploie uniquement à découvrir les nœuds les plus influents du réseau, sans modélisation explicite des dynamiques des récompenses successives.

Nous introduisons une nouvelle instance de bandits, les bandits récurrents, où des relations temporelles doivent être extraites en ligne à partir d’observations largement incomplètes, afin d’appréhender l’aspect non-stationnaire des espérances de récompense. Dans la suite, deux types d’approches sont proposées pour répondre à ce problème : un premier modèle considérant des dépendances linéaires entre les récompenses de périodes d’écoute successives (section 2.2.4.1), et un second faisant intervenir des hypothèses de transition entre des états latents du système (section 2.2.4.2).

### 2.2.4.1 Modèle relationnel

Une première possibilité est donc de supposer un modèle de dépendances relationnelles entre récompenses au temps  $t - 1$  et espérances de récompense au temps  $t$  :

$$\forall i \in \{1, \dots, K\}, \exists \theta_i \in \mathbb{R}^{K+1} \text{ tel que } : \forall t \in \{2, \dots, T\} : \mathbb{E}[r_{i,t} | R_{t-1}] = \theta_i^\top R_{t-1}^+ \quad (2.38)$$

où  $R_t = (r_{1,t}, \dots, r_{K,t})^\top \in \mathbb{R}^K$  est le vecteur de récompenses au temps  $t$  et  $R_t^+ = (R_t, r_{K+1,t})$  y concatène un terme de biais constant  $r_{K+1,t} = 1$ . Les récompenses de chaque utilisateur  $i$  sont donc expliquées par une application linéaire du vecteur de récompenses passées selon des paramètres  $\theta_i$ . Nous considérons dans la suite les hypothèses suivantes :

- **Vraisemblance des données** :  $\forall i \in \{1, \dots, K\}, \exists \theta_i \in \mathbb{R}^{K+1}$  tel que  $\forall t \in \{2, \dots, T\} : r_{i,t} = \theta_i^\top R_{t-1}^+ + \epsilon_{i,t}$ , où  $\epsilon_{i,t} \sim \mathcal{N}(0, \sigma^2)$  (bruit gaussien de moyenne 0 et variance  $\sigma^2$ ).
- **Prior sur les paramètres** :  $\forall i \in \{1, \dots, K\} : \theta_i \sim \mathcal{N}(0, \alpha^2 \mathbf{I})$  (vecteur gaussien de taille  $(K + 1)$ , de moyenne 0 de matrice de covariance  $\alpha^2 \mathbf{I}$ ).
- **Prior au temps 1** :  $\forall i \in \{1, \dots, K\} : r_{i,1} \sim \mathcal{N}(0, \sigma^2)$ .

Le modèle présenté ici s’apparente à un processus autorégressif vectoriel bayésien de dimension  $K$  et d’ordre 1 [Karlsson et al., 2013]. Une difficulté supplémentaire dans notre cas est qu’un certain nombre de valeurs au temps  $t - 1$  sont manquantes, en raison du processus décisionnel de bandit associé qui ne permet d’observer que  $k$  récompenses à chaque itération (toutes les récompenses des utilisateurs non sélectionnés à  $t - 1$  sont inconnues).

Contrairement aux sections précédentes, nous employons ici un algorithme de Thompson Sampling (TS), bien qu’une version type *LinUCB* aurait pu être envisageable également. Afin de mettre en œuvre cet algorithme, nous devons pouvoir, pour tout  $t \geq 2$ , échantillonner la récompense espérée  $\tilde{r}_{i,t}$  à partir de la distribution postérieure de  $\theta_i^\top R_{t-1}^+$  pour chaque utilisateur  $i$ . Si les récompenses de tous les utilisateurs étaient disponibles à chaque étape, cela reviendrait à un problème de bandit contextuel classique. Cependant bien sûr, pour chaque  $s \in \{1, \dots, t - 1\}$ ,  $i \notin \mathcal{K}_s$  implique que la récompense  $r_{i,s}$  n’est pas disponible et doit être traitée comme une variable aléatoire. TS doit donc être mis en œuvre selon la distribution jointe suivante :

$$p((r_{i,s})_{s=1..t-1, i \notin \mathcal{K}_s}, (\theta_i)_{i=1..K} | (r_{i,s})_{s=1..t-1, i \in \mathcal{K}_s}) \quad (2.39)$$

Du fait de l’aspect récurrent du problème, cette distribution ne peut être obtenue directement. Si l’utilisation de méthodes MCMC, type Gibbs Sampling, auraient pu permettre des garanties théoriques de sous-linéarité du pseudo-regret cumulé du processus TS, car convergeant vers des échantillons issus de la vraie distribution jointe, en pratique ce genre de technique induirait des coûts computationnels

et une variance d'exploration prohibitifs. Une fois encore nous avons recours à un processus d'inférence variationnelle, dans lequel nous considérons la factorisation suivante :

$$q((r_{i,s})_{s=1..t-1, i \notin \mathcal{K}_s}, (\theta_i)_{i=1..K}) = \prod_{i=1}^K q_{\theta_i}(\theta_i) \prod_{s=1}^{t-1} \prod_{i \notin \mathcal{K}_s} q_{r_{i,s}}(r_{i,s}) \quad (2.40)$$

avec  $q_{\theta_i}$  et  $q_{r_{i,s}}$  correspondant aux distributions variationnelles des facteurs supposés indépendants dans  $q$ . La minimisation de la divergence de Kullback-Leibler de  $q$  par rapport à la distribution  $p$  cible permet d'établir les deux propositions ci-dessous (preuves données dans [Lamprier et al., 2017]).

**Proposition 5** Soit  $D = (R_s^+{}^\top)_{s=1..t-1}$  la matrice  $(t-1) \times (K+1)$  des récompenses passées pour un temps  $t > 1$  donné. Pour tout  $t > 2$  et tout  $i \in \{1, \dots, K\}$ , la distribution variationnelle optimale  $q_{\theta_i}^*$  correspond à une gaussienne multivariée  $\mathcal{N}(A_{i,t-1}^{-1} b_{i,t-1}, A_{i,t-1}^{-1})$ , avec :

$$A_{i,t-1} = \frac{\mathbb{E}[D_{1:t-2}^\top D_{1:t-2}]}{\sigma^2} + \frac{I}{\alpha^2}, \quad b_{i,t-1} = \frac{\mathbb{E}[D_{1:t-2}^\top] \mathbb{E}[D_{2:t-1}^i]}{\sigma^2}$$

où  $D_{u:v}$  est la sous-matrice des lignes  $u$  à  $v$  de  $D$  et  $D_{u:v}^i$  la colonne  $i$  de cette matrice.  $\mathbb{E}[D_{1:t-2}^\top D_{1:t-2}] = \sum_{s=1}^{t-2} \mathbb{E}[R_s^+] \mathbb{E}[R_s^+]^\top + \text{Var}(R_s^+)$ , avec  $\mathbb{E}[R_s^+]$  et  $\text{Var}(R_s^+)$  donnés en proposition 6.

**Proposition 6** Soient  $\Theta$  la matrice  $K \times (K+1)$  où la ligne  $i$  vaut  $\theta_i^\top$  et  $\beta_j$  la  $j$ -ième colonne de  $\Theta$ . Pour  $t \geq 2$  et  $1 \leq s \leq t-1$ , la distribution optimale  $q_{r_{i,s}}^*$  est une gaussienne multivariée  $\mathcal{N}(\mu_{i,s}, \sigma_{i,s}^2)$ , avec :

$$\begin{aligned} \text{— Si } s = 1 : \mu_{i,s} &= \frac{\mathbb{E}[\beta_i]^\top \mathbb{E}[R_{s+1}] - \sum_{j=1, j \neq i}^{K+1} \mathbb{E}[\beta_i^\top \beta_j] \mathbb{E}[r_{j,s}]}{1 + \mathbb{E}[\beta_i^\top \beta_i]}, \quad \sigma_{i,s}^2 = \frac{\sigma^2}{1 + \mathbb{E}[\beta_i^\top \beta_i]} \\ \text{— Si } s = t-1 : \mu_{i,s} &= \mathbb{E}[\theta_i]^\top \mathbb{E}[R_{s-1}^+], \quad \sigma_{i,s}^2 = \sigma^2 \\ \text{— Sinon : } \mu_{i,s} &= \frac{\mathbb{E}[\beta_i]^\top \mathbb{E}[R_{s+1}] - \sum_{j=1, j \neq i}^{K+1} \mathbb{E}[\beta_i^\top \beta_j] \mathbb{E}[r_{j,s}] + \mathbb{E}[\theta_i]^\top \mathbb{E}[R_{s-1}^+]}{1 + \mathbb{E}[\beta_i^\top \beta_i]}, \quad \sigma_{i,s}^2 = \frac{\sigma^2}{1 + \mathbb{E}[\beta_i^\top \beta_i]} \end{aligned}$$

où  $\mathbb{E}[\beta_i^\top \beta_j] = \sum_{l=1}^K \text{Var}(\theta_l)_{i,j} + \mathbb{E}[\theta_l]_i \mathbb{E}[\theta_l]_j$ .

Les distributions données en propositions 5 et 6 sont inter-dépendantes. Leur estimation doit donc être effectuée par une procédure itérative, que l'on détaille par l'algorithme 4. Notons que, pour les récompenses observées, nous utilisons bien sûr leur valeur plutôt que leur espérance. Concrètement, la composante  $i$  de  $\mathbb{E}[R_s^+]$  vaut  $r_{i,s}$  dans les propositions 5 et 6 si  $i \in \mathcal{K}_s$ . Par ailleurs,  $\text{Var}(R_s)$  correspond à la matrice diagonale où l'élément  $(i, i)$  vaut 0 si  $i \in \mathcal{K}_s$ ,  $\sigma_{i,s}^2$  sinon. Notons enfin que  $r_{K+1,s}$  est toujours connu :  $\mathbb{E}[r_{K+1,s}] = 1$  et  $\text{Var}(r_{K+1,s}) = 0$  pour tout  $s$ .

---

**Algorithme 4** : Variational Inference
 

---

**Input** :  $nblt$   
 1 **for**  $l = 1..nblt$  **do**  
 2     **for**  $i = 1..K$  **do**  
 3         Calcul de  $A_{i,t-1}$  et  $b_{i,t-1}$   
 4             selon la proposition 5;  
 5         **for**  $s = 1..t-1$  **do**  
 6             **if**  $i \notin \mathcal{K}_s$  **then** Calcul de  $\mu_{i,s}$ ,  $\sigma_{i,s}^2$   
 7                 selon la proposition 6;  
 8         **end**  
 9     **end**  
 10 **end**

---



---

**Algorithme 5** : Recurrent TS
 

---

1 **for**  $t = 1..T$  **do**  
 2     Perform Variational Inference;  
 3     **for**  $i = 1..K$  **do**  
 4         Sample  $\tilde{\theta}_i \sim q_{\theta_i}^*$ ;  
 5         **if**  $i \notin \mathcal{K}_{t-1}$  **then**  
 6             Sample  $\tilde{r}_{i,t-1} \sim q_{r_{i,t-1}}^*$ ;  
 7              $\tilde{r}_{i,t} = \tilde{\theta}_i^\top \tilde{R}_{t-1}^+$ ;  
 8         **end**  
 9          $\mathcal{K}_t \leftarrow \operatorname{argmax}_{\mathcal{K} \subseteq \mathcal{K}, |\mathcal{K}|=k} \sum_{i \in \mathcal{K}} \tilde{r}_{i,t}$ ;  
 10         **for**  $i \in \mathcal{K}_t$  **do** Collect  $r_{i,t}$ ;  
 11 **end**

---

L'algorithme 5 décrit notre procédure de Thompson Sampling relationnel récurrent, qui utilise l'algorithme 4 pour estimer les distributions postérieures par inférence variationnelle. À chaque itération  $t$ , l'algorithme échantillonne chaque récompense cachée au temps  $t-1$  selon  $q_{r_{i,t-1}}^*$  et des paramètres  $\theta$  selon  $q_{\theta_i}^*$  pour chaque utilisateur  $i$ . Cela permet de calculer une espérance de récompense pour chaque utilisateur selon  $\tilde{r}_{i,t} = \tilde{\theta}_i^\top \tilde{R}_{t-1}^+$  pour chaque utilisateur  $i$ . Les  $k$  utilisateurs avec la meilleure espérance  $\tilde{r}_{i,t}$  sont suivis pour la période d'écoute  $[t; t+1[$ . Notons que comme dans la section précédente, la complexité de l'algorithme augmente avec les itérations car le nombre de récompenses inconnues dans l'historique croît linéairement. Comme dans la section précédente, on propose de contourner ce problème en ne ré-évaluant les distributions variationnelles de récompenses  $r_{i,s}$  que pour  $s$  dans une fenêtre des  $H$  derniers pas de temps, les autres restant figées à leurs paramètres calculés à l'étape  $s+H$ .

La complexité reste un facteur limitant de l'algorithme car on a des matrices  $K \times K$  à inverser, ce qui est totalement prohibitif dans des applications de collecte avec un grand nombre d'utilisateurs comme sur Twitter. Par ailleurs, les dépendances temporelles ne sont prises ici qu'entre pas de temps successifs (i.e., de  $t$  vers  $t+1$ ). Or, étant donné le taux de données manquantes à chaque étape, cela peut s'avérer particulièrement instable. Par ailleurs, la prise en compte de dépendances plus long terme pourrait être souhaitable. Le modèle de la section vise à dépasser ces limites, par l'utilisation d'un espace d'états latents pour la modélisation des dynamiques du système.

### 2.2.4.2 Modèle à espace latent

Plutôt que de modéliser des dépendances temporelles directes entre récompenses des utilisateurs sur les itérations successives du processus de collecte, nous considérons ici une version où nous supposons l'existence d'un état latent  $h_t \in \mathbb{R}^d$ , responsable des valeurs de récompense à chaque étape  $t$ . La taille  $d$  de cet espace est supposé très inférieur à  $K$ . Les variations d'espérance de récompense sur les différents utilisateurs sont expliquées par les évolutions de l'état latent dans son espace de représentation. La dynamique de l'état latent est donné par une transformation linéaire à chaque étape :

$$\exists \Theta \in \mathbb{R}^{d \times d}, \forall t \in \{2, \dots, T\} : \mathbb{E}[h_t | h_{t-1}] = \Theta h_{t-1} \quad (2.41)$$

avec  $\Theta$  une matrice de transition  $d \times d$ . Notons que bien d'autres transformations auraient pu être envisagées, comme des déplacements résiduels dans l'espace d'états par exemple. Selon un état  $h_t$

donné, les espérances de récompense sont alors définies par :

$$\forall i \in \{1, \dots, K\}, \exists W_i \in \mathbb{R}^d, \exists b_i \in \mathbb{R}, \forall t \in \{1, \dots, T\} : \mathbb{E}[r_{i,t} | h_t] = W_i^\top h_t + b_i \quad (2.42)$$

Le modèle est illustré sur 3 itérations par la figure 2.8, où on observe une dépendance temporelle entre états successifs.

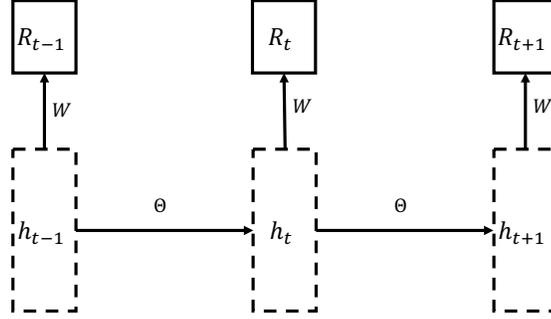


FIGURE 2.8 – Modèle récurrent génératif à états latents.

Pour ce modèle, nous considérons les hypothèses génératives suivantes :

- **Vraisemblance 1** :  $\exists \Theta \in \mathbb{R}^{d \times d}, \forall t \in \{2, \dots, T\} : h_t = \Theta h_{t-1} + \epsilon_t$ , où  $\epsilon_t \sim \mathcal{N}(0, \delta^2 I)$ .
- **Vraisemblance 2** :  $\forall i \in \{1, \dots, K\}, \exists W_i \in \mathbb{R}^d, \exists b_i \in \mathbb{R}$  tel que :  $\forall t \in \{1, \dots, T\} : r_{i,t} = W_i^\top h_t + b_i + \epsilon_{i,t}$ , où  $\epsilon_{i,t} \sim \mathcal{N}(0, \sigma^2)$ .
- **Prior sur  $h_1$**  :  $h_1 \sim \mathcal{N}(0, \delta^2 I)$ , où  $I$  est la matrice identité de taille  $d$ .
- **Prior sur  $\Theta$**  :  $\forall i \in \{1, \dots, d\} : \theta_i \sim \mathcal{N}(0, \alpha^2 I)$ , où  $\theta_i$  est la  $i$ -ième ligne de  $\Theta$ .
- **Prior sur  $W$**  :  $\forall i \in \{1, \dots, K\} : W_i \sim \mathcal{N}(0, \gamma^2 I)$  et  $b_i \sim \mathcal{N}(0, \gamma^2)$ .

Cela définit un système linéaire dynamique pour lequel les paramètres doivent être estimés. Notons qu'avec une matrice de transition  $\Theta$  et une matrice d'émission  $(W_i)_{i=1..K}$  connues, ce modèle correspondrait à un filtre de Kalman [Kalman, 1960], habituellement utilisé pour inférer les états de systèmes physiques avec observations bruitées mais dynamiques connues. Notre problème tombe dans le cadre des systèmes linéaires dynamiques bayésiens, pour lesquels une approche variationnelle a été proposée dans [Beal, 2003]. Cependant, l'application directe de cette méthode générique est complexe. Suivant une approche similaire à celle de la section précédente, nous considérons l'approximation de champ moyen suivante :

$$q(h_1, \dots, h_{t-1}, \Theta, W, b) = \prod_{s=1}^{t-1} q_{h_s}(h_s) \prod_{i=1}^K q_{W_i, b_i}(W_i, b_i) \prod_{j=1}^d q_{\theta_j}(\theta_j) \quad (2.43)$$

Des distributions postérieures gaussiennes peuvent alors être établies pour les différents facteurs par minimisation de la divergence de Kullback-Leibler de  $q$  avec la distribution cible  $p$ . Nous ne présentons pas ici les détails de ces distributions, qui sont donnés avec preuves associées dans [Lamprier et al., 2017]. Un algorithme de Thompson Sampling, similaire à l'algorithme 5 peut alors être défini sur la base de ces distributions postérieures. À chaque itération  $t$ , la procédure d'inférence variationnelle est mise en œuvre pour estimer les distributions postérieures des différents facteurs. Des échantillons

de  $h_{t-1}$ ,  $\Theta$  et de chaque  $(W_i, b_i)$  sont obtenus de ces distributions pour calculer  $\tilde{r}_{i,t} = \tilde{W}_i^\top \tilde{\Theta} h_{t-1} + \tilde{b}_i$  pour chaque utilisateur. Les  $k$  utilisateurs ayant la meilleure récompense espérée  $\tilde{r}_{i,t}$  sont sélectionnés. Un bénéfice majeur de cette approche comparée à celle de la section précédente est que la complexité n'augmente plus de manière quadratique avec le nombre d'utilisateurs. Au temps  $t$ , le nombre de variables aléatoires à estimer est de  $d^2 + d(t-1) + K(d+1)$ , ce qui reporte l'aspect quadratique au nombre de dimensions de l'espace latent  $d$ , qui est bien inférieur au nombre d'utilisateurs. Par ailleurs, la même technique qu'aux sections précédentes, par fenêtre glissante, peut être employée pour rendre la complexité constante sur l'ensemble du processus de collecte.

La figure 2.9 donne un aperçu des performances du modèle, nommé *StateTS\_<d>*, avec  $d$  la dimension de l'espace latent utilisé, sur la tâche de collecte sur un corpus et pour une fonction de récompense similaires à ceux des sections précédentes. Outre les approches déjà considérées, la figure donne des résultats pour les approches non-stationnaires *Discounted UCB (D-UCB)* et *Sliding-Window UCB (SW-UCB)* [Garivier and Moulines, 2011]. Les résultats montrent le bon comportement de la méthode, qui permet une collecte plus efficace que ses différents concurrents. Notons que nous n'avons pas reporté ici de méthodes utilisant des informations auxiliaires comme *HiddenLinUCB*, qui obtient souvent de meilleurs résultats sur les tâches de collecte. Néanmoins, une hybridation des deux approches pourrait être envisagée, afin d'exploiter les contenus observés à chaque  $t$ , en plus des récompenses associés, afin d'inférer les états latents  $h_t$ . Des résultats complémentaires sont présentés dans [Gisselbrecht, 2017], avec notamment une analyse des évolutions de l'état latent en présence de cycles dans les données.

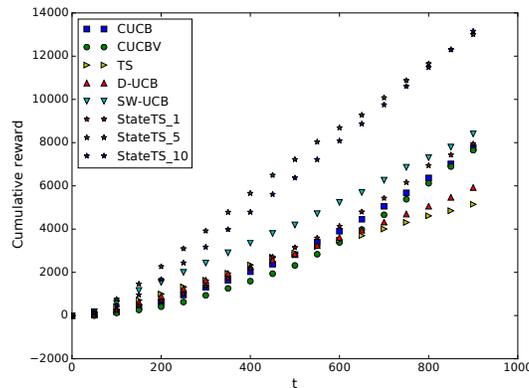


FIGURE 2.9 – Performances du modèle de Thompson Sampling récurrent StateTS.

### Publication associée :

- Lamprier, S., Gisselbrecht, T., and Gallinari, P. (2017). Variational thompson sampling for relational recurrent bandits. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part II*, pages 405–421

## 2.3 Conclusions et Perspectives

Dans ce chapitre, nous avons passé en revue diverses approches pour le traitement et la capture de données issues des réseaux sociaux, sous les contraintes d'accès imposées par les médias considérés. Plus généralement, la tâche s'apparente à une tâche de sélection de flux dynamique, pour laquelle nous avons proposé des approches de bandit, avec prise en compte de diverses dépendances dans les données. Bien souvent, nous avons à prendre des décisions sur la base de données fortement incomplètes, que ce soit dans un cadre stationnaire basé sur des profils inconnus, dans un cadre évolutif basé sur des contextes de décision partiellement cachés ou dans un cadre relationnel basé sur des utilités non observées. Pour tous ces contextes, à l'incertitude classique sur les paramètres du processus de bandit considéré, s'ajoute des incertitudes sur les estimations réalisées. À plusieurs reprises nous avons eu recours à un processus d'inférence variationnelle, ayant pour but d'approximer les distributions postérieures des variables aléatoires du modèle, afin d'en appréhender l'incertitude au travers de méthodes de type UCB ou Thompson Sampling.

Au delà de l'hybridation des différentes méthodes proposées, en vue de combiner divers indicateurs structurels et temporels pour l'estimation des espérances de récompense, différentes pistes de recherche nous paraissent mériter un approfondissement particulier, que nous détaillons ci-après. La section 2.3.1 discute ainsi des alternatives à l'inférence variationnelle envisageables dans le cadre des approches de bandits que l'on considère. La section 2.3.2 propose une étude des possibilités pour dépasser le cadre linéaire des méthodes de bandit contextuel proposées dans ce chapitre. Enfin, la section 2.3.3, plus prospective, traite des possibles perspectives de recherche permettant d'intégrer un aspect évolutif à la collecte, pour prendre en compte les données collectées dans des fonctions de récompense adaptatives (visant par exemple à former des jeux de données respectant diverses propriétés globales).

### 2.3.1 Inférence Bayésienne

Comme discuté en section 2.2.3.1, l'inférence variationnelle (VI) correspond à une utilisation "exclusive" de la KL (i.e.,  $KL(q||p)$  avec  $p$  la distribution cible), à la manière des approches Espérance-Maximisation (EM). La minimisation de cette KL conduit à l'obtention d'une distribution  $q$  incluse dans  $p$ , qui tend à ignorer des modes de  $p$  du fait des hypothèses simplificatrices émises pour  $q$  (typiquement des hypothèses de champ moyen). De ce fait, l'inférence variationnelle est connue pour sous-estimer l'incertitude des modèles [Riquelme et al., 2018]. Un autre type d'approches, les méthodes d'Espérance-Propagation (EP) [Opper and Winther, 2000, Minka, 2001], permet la minimisation d'une version "inclusive" (i.e.,  $KL(p||q)$ ) de la divergence de Kullback-Leibler.

Ces approches visent à approximer une distribution de probabilité jointe, de la forme  $p(z|x) \propto \prod_i \varphi_i(z, x)$ , avec  $\varphi_i$  une fonction potentiel de l'instanciation  $z$  connaissant les observations  $x$ . Un certain nombre de fonctions  $\varphi_i$  peuvent par exemple chacune s'intéresser dans notre cas à une observation de contexte manquante particulière, d'autres à la vraisemblance des récompenses selon les contextes et les paramètres dans  $z$ , etc. Généralement, par souci computationnel, les fonctions  $\varphi_i$  considérées sont prises dans la famille des distributions exponentielles. Comme pour l'inférence variationnelle, l'objectif est donc d'approximer  $p(z|x)$  selon une distribution  $q(z; \phi)$ , avec  $\phi$  les paramètres de cette distribution, mais cette fois l'idée est de définir des paramètres  $\phi$  qui minimisent la divergence  $KL(p(z|x)|q(z; \phi))$ , ce qui peut se faire de manière analytique par mise en correspondance de moments pour  $p$  et  $q$  de la famille exponentielle, si tant est que les moments de  $p$  sont calculables.

L'idée des approches Espérance-Propagation prend son inspiration dans les méthodes *Assu-*

*med Density Filtering (ADF)* [Opper and Winther, 1998], dont le principe est d'intégrer les fonctions  $\varphi_i(z, x)$  les unes après les autres dans la distribution  $q$  courante, en considérant à chaque étape  $i$  la minimisation de  $\text{KL}(\varphi_i(z, x)q(z; \phi^{(i-1)}) || q(z; \phi))$  selon  $\phi$  pour définir  $\phi^{(i)}$ . Les méthodes *ADF* sont néanmoins très dépendantes de l'ordre dans lequel les fonctions potentiel sont intégrées dans  $q$ . Pour dépasser cette limite, les approches *EP* proposent de considérer une factorisation de  $q$  selon un produit d'approximations des différentes fonctions potentiel :  $q(z; \phi) \propto \prod_i m_i(z)$ . L'idée à chaque étape de l'optimisation est de retirer la contribution d'un facteur  $m_i$  donné de  $q$  pour obtenir  $q(z; \phi^{\setminus i}) \propto q(z; \phi) / m_i(z)$ . On peut alors raffiner l'approximation en calculant  $\phi^* = \text{argmin}_{\phi} \text{KL}(\varphi_i(z, x)q(z; \phi^{\setminus i}) || q(z; \phi))$  puis en considérant  $m_i(z) \propto q(z; \phi^*) / q(z; \phi^{\setminus i})$ . Dans le cas de distributions gaussiennes, des contraintes additionnelles peuvent être ajoutées pour améliorer la convergence [Minka, 2001]. Notons que l'algorithme *Loopy Belief Propagation* [Pearl, 1986], qui est un algorithme par passage de messages de croyance entre facteurs du graphe bayésien, est un cas spécial des méthodes *EP*.

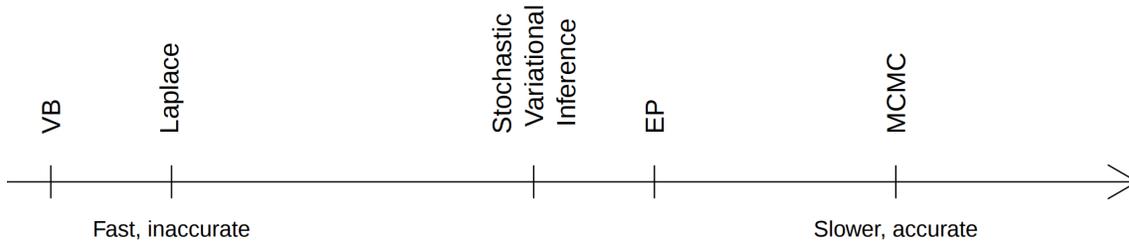


FIGURE 2.10 – Méthodes d'inférence bayésienne [Barthelmé, 2015].

Bien qu'il n'existe pas de garantie générale de convergence pour les méthodes *EP*, [Dehaene and Barthelmé, 2015] prouve des bornes d'erreur d'approximation pour des distributions cibles formées par des facteurs log-concaves, ce qui est le cas de nos modèles gaussiens. [Dehaene and Barthelmé, 2018] montre que, à moins d'une initialisation malencontreuse, les méthodes *EP* avec facteurs log-concaves convergent asymptotiquement vers la distribution jointe cible. Il serait utile d'expérimenter ces méthodes, souvent plus précises bien que plus lentes que l'inférence variationnelle (voir figure 2.10), dans le cadre de nos approches de collecte avec données manquantes. Notons aussi l'approche stochastique *SEP* de [Li et al., 2015], qui permet l'application en ligne de méthodes *EP* à large échelle, où une approximation globale de la distribution postérieure est maintenue comme en *VI*, mais où les mises à jour sont réalisées localement (*EP*). L'idée serait également de s'appuyer sur les résultats de [Dehaene and Barthelmé, 2015, Dehaene and Barthelmé, 2018] pour donner des garanties théoriques sur l'évolution du regret dans le cadre de nos approches de capture dynamique, que l'utilisation de techniques d'inférence variationnelle ne permettait pas. Bien que beaucoup plus lentes, des techniques *MCMC* telles que l'échantillonnage hamiltonien ou l'échantillonnage de Gibbs, tous deux des cas spéciaux de l'algorithme de Metropolis-Hasting, pourront également être envisagées pour garantir un pseudo-regret sous-linéaire pour l'approche récurrente à états, sous des hypothèses de régularité de l'état latent.

### 2.3.2 Réseaux de neurones probabilistes

Dans les travaux présentés dans ce chapitre, nous nous sommes toujours cantonnés à des modèles de dépendance linéaire (i.e., espérance de récompense linéaire en fonction des contextes ou de l'état du système, fonction de dynamique linéaire dans l'espace latent, relations linéaires entre utilisateurs), comme c'est le cas dans la plupart des travaux de bandit contextuel, car cela permet des résolutions analytiques des calculs de distribution postérieure des paramètres à chaque étape du processus. Cela permet d'éviter par exemple d'employer des méthodes itératives de montée de gradient pour la recherche de paramètres. Mais cela limite fortement l'expressivité des modèles. Si les dépendances sous-jacentes ne sont pas linéaires (e.g., l'espérance de récompense ne peut s'exprimer selon  $\mathbb{E}[r_{i,t}|x_{i,t}] = \theta^\top x_{i,t}$  pour tout  $i$  et temps  $t$ ), l'apprentissage d'un modèle de régression peut s'en trouver fortement biaisé. Comme énoncé précédemment, des approches comme [Wang et al., 2016] ont par exemple proposé de limiter ce biais d'apprentissage par inférence de dimensions additionnelles sur les contextes, visant à accueillir des variables explicatives permettant de stabiliser les prédictions selon la partie observée des contextes. Néanmoins, ce genre de technique ne répond que partiellement au problème, en supposant la linéarité du modèle par rapport à des vecteurs de contexte complétés, ce qui n'est pas toujours vérifié (sauf pour des vecteurs très grands desquels aucune régularité ne peut être exploitée). Des travaux tels que [Filippi et al., 2010] ont d'autre part considéré les bandits pour le cas du modèle linéaire généralisé. Pour le cas des problèmes de comptage suivant une loi de Poisson, nous avons notamment contribué dans [Gisselbrecht et al., 2015d], en proposant une approche de bandit basée sur une approximation log-gamma [El-Sayyad, 1973]. D'autres travaux relaxent le cadre linéaire en s'intéressant aux fonctions de récompense respectant une propriété de Lipschitz dans l'espace de contextes [Bubeck et al., 2008] ou à des fonctions appartenant à un espace de Hilbert reproductible (*RKHS*) [Valko et al., 2013]. Ces approches requièrent néanmoins toujours des contraintes relativement restrictives sur la forme de la fonction de récompense.

Une possibilité pour dépasser ce cadre est d'employer des réseaux de neurones probabilistes, permettant de définir toutes sortes de relations entre variables. Dans un cadre gaussien tel que celui que nous avons largement considéré dans ce chapitre, cela revient par exemple à définir des réseaux qui émettent une moyenne  $\mu_\theta(x)$  et une variance  $\sigma_\theta(x)$  pour une entrée  $x$ , avec  $\theta$  les poids du réseau. Dans le cadre des bandits, il nous faut considérer les poids du réseau  $\theta$  comme une variable aléatoire, dont on cherche à estimer la distribution postérieure afin d'établir des intervalles de confiance (pour les approches type *UCB*) ou d'être à même d'échantillonner de cette distribution (pour les approches type *TS*). Si dans le cas linéaire avec distribution de récompense  $p(r_{i,t}|x_{i,t};\theta)$  sous-gaussienne et prior gaussien  $p(\theta)$ , cette postérieure peut être calculée analytiquement (voir (2.12)), pour  $p(r_{i,t}|x_{i,t};\theta)$  un réseau de neurones avec activations non linéaires cette résolution analytique de  $p(\theta|(r_{i,s}, x_{i,s})_{i \in \mathcal{X}, s < t})$  est impossible. Une première possibilité, explorée dans [Snoek et al., 2015], est de ne considérer comme variable aléatoire que les paramètres de la dernière couche, linéaire, du réseau de neurones. Avec l'ensemble du réseau appris pour maximiser la vraisemblance des observations (par estimation ponctuelle selon par exemple un critère de maximum a posteriori), la postérieure n'est ensuite calculée que pour les paramètres  $\beta$  de la dernière couche du réseau, de manière classique et en considérant donc, pour tout utilisateur  $i$  et tout temps  $t$ ,  $r_{i,t} = z_{i,t}^\top \beta$  avec  $z_{i,t}$  l'entrée de la dernière couche du réseau. Cela mène à capturer l'incertitude selon une représentation  $z_{i,t}$  donnée fixe, bien que reconsidérée régulièrement, mais ne garantit pas une exploration efficace de l'espace des observations. Par exemple dans le cadre d'un algorithme de bandit, deux contextes  $x_1$  et  $x_2$  peuvent être projetés sur un même  $z$ , car associés aux mêmes récompenses sur les observations passées, bien que  $x_1$  corresponde à un contenu bien mieux connu que  $x_2$ . L'action correspondant à  $x_2$  ne peut alors pas bénéficier

d'une prime d'incertitude dans son score de sélection, tel que cela aurait été le cas dans un algorithme comme LinUCB, afin de reconsidérer éventuellement la représentation  $z$  qui en est faite.

Une alternative mieux fondée, selon par exemple [Blundell et al., 2015], consiste à faire ressembler une distribution  $q_\phi(\theta)$  à  $p(\theta|\mathcal{H}_{t-1})$ , avec  $\mathcal{H}_{t-1}$  l'historique d'observations de contextes et récompenses au temps  $t$  du processus :

$$\phi^* = \arg \min_{\phi} \text{KL}[q_\phi(\theta) \| p(\theta|\mathcal{H}_{t-1})] \quad (2.44)$$

$$= \arg \min_{\phi} \int q_\phi(\theta) \log \frac{q_\phi(\theta)}{p(\theta)p(\mathcal{H}_{t-1}|\theta)} d\theta \quad (2.45)$$

$$= \arg \min_{\phi} \text{KL}[q_\phi(\theta) \| p(\theta)] - \mathbb{E}_{q_\phi(\theta)} [\log p(\mathcal{H}_{t-1}|\theta)] \quad (2.46)$$

Avec  $p(\theta)$  un prior gaussien centré pour les paramètres du réseau et  $q_\phi$  définissant une gaussienne pour chaque paramètre, la divergence de Kullback-Leibler de (2.46) possède une forme close, qui peut être minimisée simplement par descente de gradient. Pour l'optimisation du deuxième terme de (2.46), il s'agit de procéder à une estimation de Monte Carlo du gradient par échantillonnage. Le gradient à considérer étant défini selon la distribution d'échantillonnage  $q_\phi$ , il nous faut procéder à une re-paramétrisation (*reparameterization trick* [Kingma and Welling, 2013, Rezende et al., 2014]), qui consiste en la reformulation de  $\theta$  selon une fonction déterministe  $\theta_\phi(\epsilon)$  d'une variable standard gaussienne  $\epsilon$ <sup>7</sup>. Le travail dans [Blundell et al., 2015] donne un schéma de retro-propagation efficace pour  $\nabla_\phi \log p(\mathcal{H}_{t-1}|\theta_\phi(\epsilon^{(i)}))$ , avec  $p$  un réseau de neurones de paramètres  $\theta_\phi(\epsilon^{(i)})$ , nommé *BayesByBackprop (BBB)*. Ce principe est appliqué dans [Blundell et al., 2015] pour l'établissement d'un algorithme de Thompson Sampling dans le cadre d'un problème de bandit contextuel.

Dans le cadre des bandits contextuels classiques, les approches variationnelles pour réseaux de neurones telles que *BBB* [Blundell et al., 2015] n'apparaissent cependant pas comme les plus performantes, notamment comparées aux méthodes de sous-gradients [Salas et al., 2018, Riquelme et al., 2018] ou aux processus gaussiens<sup>8</sup> (*GP*) [Rasmussen, 2003, Riquelme et al., 2018]. Il convient néanmoins de les expérimenter dans des cadres très incertains comme ceux que nous considérons dans nos approches avec contextes cachés, pour la capture de données à partir de flux. Pour des approches récurrentes comme le modèle à états cachés donné en section 2.2.4.2, des méthodes variationnelles pour réseaux de neurones récurrents, telles que les approches de [Krishnan et al., 2017] ou [Fraccaro et al., 2016] dont nous discutons dans les chapitres suivants, pourraient être envisagées.

Une application des méthodes *EP* décrites ci-dessus est aussi envisageable dans le cadre de fonctions de prédiction neuronales [Li et al., 2015]. Un cas intéressant est le modèle de [Hernández-Lobato et al., 2016], qui considère une  $\alpha$ -divergence plutôt que la classique divergence de Kullback-Leibler, qui tend vers une méthode VI pour  $\alpha \rightarrow 0$  et une méthode EP pour  $\alpha = 1$ . Une autre possibilité est d'utiliser des méthodes de sous-gradient, tel que par exemple le populaire optimiseur *ADAM* [Kingma and Ba, 2014], dont une interprétation probabiliste peut être extraite pour établir l'incertitude des paramètres [Salas et al., 2018].

7. Le *reparameterization trick* [Kingma and Welling, 2013, Rezende et al., 2014] consiste à considérer la fonction :  $\theta_\phi(\epsilon) = \epsilon \times \sigma_\phi + \mu_\phi$ , avec  $\epsilon \sim \mathcal{N}(0,1)$ .  $\mu_\phi$  et  $\sigma_\phi$  sont respectivement les paramètres de moyenne et d'écart-type de  $q_\phi$ . On a alors  $\nabla_\phi \mathbb{E}_{q_\phi(\theta)} [\log p(\mathcal{H}_{t-1}|\theta)] = \nabla_\phi \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1)} [\log p(\mathcal{H}_{t-1}|\theta_\phi(\epsilon))] \approx \frac{1}{N} \sum_{i=0}^N \nabla_\phi \log p(\mathcal{H}_{t-1}|\theta_\phi(\epsilon^{(i)}))$ , avec  $\epsilon^{(i)} \sim \mathcal{N}(0,1)$ .

8. Il peut être montré que tout réseau de neurone bayésien, sous priors gaussiens et en la limite d'un nombre infini d'unités sur les couches cachées, converge vers un processus gaussien [Neal, 2012]. Les GPs constituent donc une baseline importante mais leur complexité augmente de manière cubique avec les observations, ce qui les rend peu adaptés pour des tâches de décision en ligne à large échelle.

### 2.3.3 Récompenses dynamiques

Comme mentionné en section 2.1.1, les expérimentations des modèles présentés dans ce chapitre se sont toujours cantonnées à des fonctions de récompense stationnaires en fonction des contenus collectés, où un même message obtient toujours le même score quel que soit l'avancement de la collecte. Si des variations d'utilité sont observées au cours de la collecte sur les différents utilisateurs, elles ne sont que du fait des contenus postés par les utilisateurs qui sont amenés à évoluer, pas de notre manière de les appréhender qui serait modifiée. Or, pour des tâches de collecte de données sur les réseaux, un aspect important, que nous avons abordé en introduction, peut être d'amasser des données dont l'ensemble satisfait un certain nombre de propriétés macroscopiques désirables (e.g., forte densité d'un graphe relationnel des données, coefficient de clustering élevé, clusters thématiques homogènes, graphe *free-scale*, représentation équilibrée de divers groupes d'intérêt, etc.). Dans ce cas, il faut pouvoir mettre en regard chaque nouveau contenu avec les données précédemment collectées, au travers d'une fonction de récompenses dépendant de l'historique de la collecte. Soit  $\omega_{i,t} \in \Omega$  le contenu reçu de l'utilisateur  $i$  au moment  $t$  de la collecte, et  $\psi : \Omega \rightarrow \Psi$  une fonction de représentation de ce contenu, on peut considérer alors la récompense  $r_{i,t} = g(\psi(\omega_{i,t}), \gamma_{t-1})$ , avec  $\gamma_{t-1} \in \Gamma$  un ensemble de données dépendant de l'historique  $\mathcal{H}_{t-1}$  au temps  $t$ . L'objectif est d'assurer une complexité constante pour la fonction de récompense  $g : \Psi \times \Gamma \rightarrow \mathbb{R}$ , avec donc une construction incrémentale du résumé des données historiques sur lesquelles elle s'appuie :  $\gamma_t = \gamma((\omega_{i,t})_{i \in \mathcal{X}_t}, \gamma_{t-1})$ , où  $\gamma : \Omega^k \times \Gamma \rightarrow \Gamma$  est une fonction d'intégration des données nouvellement collectées à  $t$  dans le résumé des données historiques précédent  $\gamma_{t-1}$ . Selon les cas, en s'appuyant sur des méthodes de *Stream Data Mining* présentées en section 2.1.1 (i.e., *Sketching*, agrégation, échantillonnage, etc.),  $\gamma$  peut correspondre à un filtrage des données collectées pour se restreindre à un ensemble particulier de taille fixe, et dans ce cas la fonction  $g$  s'intéresse à la bonne intégration du nouveau contenu dans cet ensemble, ou bien à une fonction de construction incrémentale de statistiques suffisantes sur les données successivement collectées.

Pour le cas de ce genre de fonctions de récompense évolutives, des méthodes de bandit adaptées doivent être considérées. Les modèles stochastiques classiques, type CUCB ou CUCBV, reposent sur des hypothèses de stationnarité des distributions de récompenses. Leur terme d'exploration peut leur permettre de s'adapter à des changements ponctuels de distribution, mais seulement sur le long terme et si le nombre de changements est limité. Si le modèle contextuel dynamique présenté en section 2.2.3 est capable d'anticiper des variations d'utilité espérée en fonction des contextes décisionnels, la distribution des récompenses en fonction des contextes doit rester stationnaire. Une adaptation pourrait être d'intégrer les données historiques  $\gamma_t$  en entrée de la fonction de prédiction (e.g. dans le cas linéaire,  $\mathbb{E}[r_{i,t}|x_{i,t}, \gamma_t, \theta] = (x_{i,t}, \gamma_t)^\top \theta$ ), mais deux problèmes majeurs se posent alors.

**Apprentissage prédictif** D'une part l'apprentissage d'un prédicteur bayésien de  $\mathbb{E}[r_{i,t}|x_{i,t}, \gamma_t]$  paraît difficile car cela suppose la découverte de motifs de corrélation réguliers (linéaires ou non) entre les séquences  $((x_{i,t}, \gamma_t)_{i \in \mathcal{X}})_{t=1}^\infty$  et  $((r_{i,t})_{i \in \mathcal{X}})_{t=1}^\infty$ , malgré les évolutions continues de  $\gamma_t$  et les dépendances éventuellement complexes des récompenses par rapport aux contenus et à l'état de la collecte. Un début de solution pour faciliter l'apprentissage pourrait être de se restreindre à des fonctions  $\gamma$  menant à des séquences  $(\gamma_t)_{t=1}^\infty$  de vecteurs contenus dans une boule d'un espace  $\Gamma$  continu. Une autre piste serait d'exploiter nos connaissances de la fonction de récompense  $g$ , que l'on pourrait appliquer à des contenus (ou représentations simplifiées de contenus) prédits selon les contextes observés. De cette manière, on se ramène à un problème d'apprentissage stationnaire d'une fonction de prédiction de contenu  $f_\theta : \mathcal{X} \rightarrow \mathcal{Z}$ , avec  $\mathcal{Z}$  l'espace de représentation des contenus prédits, puisque les dyna-

miques de publication sur le réseau ne dépendent pas de l'état de la collecte (seules les récompenses en dépendent). Cela peut se faire par exemple selon un coût de moindres carrés  $\|f_\theta(x_{i,t}) - \varphi(\omega_{i,t})\|^2$  pour chaque contexte  $x_{i,t}$ , observé avant la période d'écoute démarrant à l'instant  $t$ , et menant à un contenu  $\omega_{i,t}$ , observé sur la période  $[t; t + 1[$ , avec  $\varphi : \Omega \rightarrow \mathcal{Z}$  une fonction de représentation du contenu. Dans le cas où  $\varphi$  est la fonction identité (i.e., la représentation prédite est définie dans le même espace que les représentations de contenu utilisées par la fonction de récompense :  $\mathcal{Z} = \Psi$ ), il s'agit simplement d'appliquer la fonction de récompense  $g$  aux sorties de  $f_\theta$  pour prédire les espérances de récompense selon l'état de la collecte  $\gamma_t$  (et définir des intervalles de confiance selon l'incertitude de  $f_\theta$ ). Cependant, selon la complexité de la fonction  $g$ , les représentations de contenu dans  $\Psi$  peuvent être difficiles à prédire. On peut alors bénéficier de la définition de fonctions de représentation  $\varphi$  simplificatrices, qui conservent un maximum d'information permettant l'induction des récompenses associées. Cette fonction de représentation peut être apprise en ligne, en parallèle d'une fonction  $\tilde{g}_\phi : \mathcal{Z} \times \Gamma \rightarrow \mathbb{R}$  qui cherche à mimer les sorties de la fonction  $g$ , en minimisant l'écart  $\|\tilde{g}_\phi(\varphi(\omega), \gamma) - g(\psi(\omega), \gamma)\|^2$  pour tout contenu  $\omega$  et résumé d'historique  $\gamma$ , selon les paramètres de  $\tilde{g}_\phi$  et de  $\varphi$ . Notons que l'apprentissage de  $\tilde{g}_\phi$  peut être largement facilité par le fait que l'on peut augmenter les données d'entraînement issues du processus de collecte par des données simulées, car ayant accès à la fonction cible  $g$  à imiter (il s'agit uniquement d'être à même de simuler des contenus et des variables d'historique  $\gamma$  réalistes).

**Politique de sélection** D'autre part, cela nous sort du cadre du bandit stochastique classique, dans lequel on suppose que les actions passées ne modifient pas les distributions de récompense sur chaque bras (ou selon les contextes dans le cas du bandit contextuel). Comme détaillé en section 2.2.4, un certain nombre d'approches, telles que les *Restless Bandits* [Tekin and Liu, 2012], s'écartent du cadre stationnaire classique en considérant des distributions qui évoluent au fil du temps, selon par exemple des processus de Markov sur chaque bras. Comme nos approches récurrentes proposées en section 2.2.4, ces instances relaxent l'aspect iid des récompenses observées, mais restent dans un cadre où les distributions de récompense sont indépendantes des actions de la politique. Le cadre du *Rested Bandit* [Cortes et al., 2017] s'écarte un peu plus du bandit classique, en considérant une instance où seule la distribution de l'action choisie à chaque pas de temps évolue. Notons l'instance similaire de *Rotting Bandits* [Levine et al., 2017] qui fait la même hypothèse mais avec une espérance de récompense pour chaque bras  $i$  qui ne fait que diminuer avec le nombre de fois où ce bras a été sélectionné (e.g., impact décroissant d'une même publicité sur un utilisateur). Dans ces deux cas, les espérances de récompenses à un instant  $t$  dépendent des actions passées puisque les sélections successives font évoluer les distributions des bras correspondants. Pour ce cadre, [Cortes et al., 2017] donne une version modifiée du pseudo-regret, appelée *path-specific dynamic pseudo-regret* :

$$\hat{R}_T = \sum_{t=1}^T \max_{i \in \mathcal{K}} \mathbb{E}[r_{i,t} | \mathcal{H}_{t-1}] - \mathbb{E}[r_{i_t,t} | \mathcal{H}_{t-1}]$$

dont l'espérance  $\mathbb{E}_{\mathcal{H}_T}[\hat{R}_T]$  est une borne supérieure du pseudo-regret classique. Cette définition du pseudo-regret à minimiser suggère des stratégies gloutonnes de sélection des actions, car on cherche à chaque étape à identifier l'action menant à la meilleure espérance de récompense immédiate, selon l'historique des actions prises. Dans ce cadre, des stratégies type *UCB* ou *Thompson Sampling* peuvent être simplement appliquées.

Un autre type d'instance, qui nous écarte encore un peu plus du cadre du bandit classique, est celui du *Recovering Bandit* [Pike-Burke and Grunewalder, 2019], dans lequel l'espérance de chaque bras

dépend du nombre de pas de temps depuis lequel il n'a pas été sélectionné (et d'un processus gaussien défini sur chaque bras). Dans un esprit similaire, [Cella and Cesa-Bianchi, 2019] suppose que l'espérance de récompense des actions chutent à 0 juste après leur sélection, pour revenir à leur valeur initiale après un nombre de pas de temps donné (inconnu de l'agent). Contrairement aux *Rested Bandits*, la récompense cumulée finale n'est pas invariante à la permutation de la séquence d'actions choisies. Il paraît alors d'autant plus important d'anticiper l'impact futur des décisions prises. Pour ce cadre, [Pike-Burke and Grunewalder, 2019] considère un pseudo-regret, nommé *d-step lookahead pseudo-regret* car défini sur des séquences d'un nombre  $d$  de pas de temps, similaire à :

$$\hat{R}_T = \sum_{h=0}^{\lfloor T/d \rfloor} \max_{s \in \mathcal{K}^d} \mathbb{E} \left[ \sum_{t=1}^d r_{s_t, hd+t} | \mathcal{H}_{hd} \right] - \mathbb{E} \left[ \sum_{t=hd+1}^{hd+d} r_{i_t, t} | \mathcal{H}_{hd} \right]$$

où le processus est divisé en époques de  $d$  pas de temps et le pseudo-regret est défini sur chaque époque en fonction de la meilleure séquence  $s = (s_1, \dots, s_d) \in \mathcal{K}^d$  en espérance sur cette époque, connaissant le passé du processus. Une notion de regret proche, basée également sur un découpage en époques mais sans possibilité de changement d'action à l'intérieur de chaque époque, est considérée dans [Ratliff et al., 2018], pour une instance de problème encore plus éloignée de l'instance classique mais plus proche de notre problème, où un état caché dépendant de l'historique définit les distributions de l'ensemble des actions. Ce genre de regret dépasse le cadre habituel de la simple récompense immédiate, en se focalisant sur les meilleures décisions à moyen terme (i.e., sur  $d$  pas de temps futurs). Un facteur de *discount*  $\gamma \in ]0; 1[$  peut être ajouté comme dans [Ratliff et al., 2018] ou [Krishnamurthy and Wahlberg, 2009] (*POMDP Rested bandit*), pour favoriser l'importance des récompenses à court terme par rapport aux plus lointaines. Cette formulation du pseudo-regret en époques nous rapproche fortement de l'apprentissage par renforcement épisodique, où chaque époque constituerait un épisode. Bien que la plupart des travaux en apprentissage par renforcement se focalise sur l'erreur en test d'une politique apprise hors-ligne (ce qui s'écarte fortement du contexte de la capture de données en temps réel sur les réseaux), divers travaux ont défini des bornes théoriques de regret cumulé en apprentissage [Jaksch et al., 2010, Osband et al., 2013, Azar et al., 2017]. Une différence notable est néanmoins qu'en apprentissage par renforcement, les distributions sont ré-initialisées en début de chaque nouvel épisode, indépendamment des actions prises dans les épisodes précédents. Pour notre cas où l'historique persiste d'une époque à l'autre, des adaptations d'approches type *UCB* ou *Thompson Sampling* ont été proposées dans [Pike-Burke and Grunewalder, 2019], avec prise en compte de récompenses cumulées sur chaque époque plutôt que des récompenses immédiates, sur lesquelles il sera possible de s'appuyer pour répondre au problème dans notre contexte de capture avec fonctions de récompense évolutives.

La considération de fonction de récompenses dynamiques dans des tâches de collecte en ligne ouvre de nombreuses voies de recherche, qui va bien au delà de la constitution de corpus de données respectant des propriétés de groupe tel qu'énoncé en introduction de la section pour justifier l'approche. Outre les les challenges d'apprentissage que cela pose, et dont nous avons largement débattu dans cette section, les définitions posées permettent d'envisager des fonctions de récompense conditionnées par un but, via des politiques universelles à la manière des tâches de renforcement guidées par des buts [Ren et al., 2019]. Cela permettrait par exemple de considérer des tâches d'extraction de fils de discussion thématique (i.e., si un thème semble émerger, on peut chercher à le suivre par une fonction de récompense conditionnée par ce thème). Des fonctions multi-buts simultanés peuvent également être considérées, notamment dans le cadre de modèles à états latents où l'on suppose que les dynamiques observées pour un but se transfèrent correctement sur des buts similaires. Des fonctions de récompense évolutives pourraient également concerner des définitions récursive de l'utilité,

à la manière de l'algorithme PageRank [Page et al., 1999], où l'on caractériserait l'importance de tout contenu collecté en fonction de l'importance de chaque autre contenu qui lui serait connecté, dans un cadre en ligne à partir de flux de données dynamiques. Dans ce cas les récompenses des contenus passés peuvent être également amenées à évoluer, ce qui ouvre sur des problématiques de recherche supplémentaires.

## Chapitre 3

# Diffusion d'information sur les réseaux

### Sommaire

---

<b>3.1 Diffusion Bayésienne : application et extension du modèle IC</b> . . . . .	<b>58</b>
3.1.1 Modèles de cascade . . . . .	60
3.1.2 Inférence de Graphes de Diffusion . . . . .	61
3.1.3 Application et Extension d'IC . . . . .	64
<b>3.2 Apprentissage de représentation pour la diffusion dans les réseaux</b> . . . . .	<b>69</b>
3.2.1 Apprentissage de représentation et diffusion . . . . .	69
3.2.2 Modèle de Cascade à représentations distribuées . . . . .	71
<b>3.3 Modèles neuronaux récurrents pour la diffusion dans les réseaux</b> . . . . .	<b>75</b>
3.3.1 Réseaux de neurones récurrents pour la diffusion . . . . .	76
3.3.2 Modèle de Cascade Récurrent . . . . .	78
3.3.2.1 Modèle Génératif Récurrent pour Diffusion Continue . . . . .	79
3.3.2.2 Apprentissage par Inférence de Trajectoires . . . . .	80
<b>3.4 Conclusions et Perspectives</b> . . . . .	<b>85</b>
3.4.1 Modèles graphiques bayésiens . . . . .	85
3.4.1.1 Corrélacion Temporelle versus Cause-Conséquence . . . . .	86
3.4.1.2 Données incomplètes . . . . .	88
3.4.1.3 Extraction d'épisodes de diffusion . . . . .	90
3.4.2 Modèles propagatifs (plus) profonds . . . . .	91
3.4.2.1 VAE Smoothing et Composition récurrente . . . . .	91
3.4.2.2 Deep Heat Diffusion . . . . .	92
3.4.2.3 RL / GANs / Inverse RL . . . . .	94
3.4.3 Modèles en ligne . . . . .	96
3.4.3.1 Modèles producteurs : Apprentissage par injection de contenu . . . . .	97
3.4.3.2 Modèles consommateurs : Apprentissage par suivi de flux . . . . .	99

---

L'extraction de dynamiques de l'information passe par l'étude des jeux d'influence sur la population étudiée. Dans le chapitre précédent nous nous attachions à exploiter des connaissances découvertes en ligne pour la capture et le suivi d'informations dans des flux dynamiques sous contraintes. Dans ce chapitre, nous nous plaçons dans un contexte plus classique d'apprentissage hors-ligne, où l'on suppose un jeu de données disponible, construit éventuellement en utilisant les méthodes de capture du chapitre précédent. On suppose également que le jeu de données est organisé sous la forme d'épisodes de diffusion, où chaque épisode concerne la propagation d'un item particulier - ou même -, qui peut correspondre à un contenu spécifique, idée ou comportement qui se diffuse dans la population étudiée. Chaque épisode considéré correspond à une séquence d'évènements de participation de nœuds du réseau (e.g., des utilisateurs du réseau) à la diffusion correspondante (e.g. séquence de partages successifs d'un même contenu). L'objectif est d'en extraire des connaissances sur les dynamiques de l'information dans la communauté étudiée. Outre l'apprentissage avec multiples données manquantes qu'implique généralement l'extraction de dynamiques de l'information à partir d'épisodes de diffusion (voir notamment sections 3.1.2 et 3.4.1.2), ce problème est confronté à divers autres défis. Parmi ceux-ci, on peut citer le fait que la participation d'un nœud à une diffusion est généralement un évènement stochastique rare, qui résulte de dépendances arborescentes complexes. De ce fait, il faut privilégier des modèles génératifs probabilistes à des modèles prédictifs déterministes, en s'appuyant sur des hypothèses simplificatrices fortes. Parmi elles, une hypothèse de monde clos est généralement émise, selon laquelle la diffusion s'explique entièrement par les dynamiques du réseau, le monde extérieur pouvant éventuellement être modélisé selon un nœud fictif unique [Gruhl et al., 2004]. Un autre défi important provient du fait que le réseau étudié n'est pas nécessairement statique, et les dynamiques sur ce réseau pas nécessairement stationnaires, les distributions d'influence pouvant dépendre de divers facteurs, tel que la nature de l'item se diffusant (e.g., un utilisateur peut avoir de l'influence sur un autre sur des sujets de sport sans en avoir en politique), des dispositions courantes des utilisateurs du réseau, ou encore du contexte de la diffusion (e.g., concurrence possible entre diffusions simultanées [Myers and Leskovec, 2012]).

Dans ce chapitre, nous présentons diverses de nos contributions pour répondre aux problématiques d'extraction de dynamiques de diffusion dans les réseaux, pour partie proposées au cours de la thèse de Simon Bourigault, en commençant par une extension d'un modèle populaire de diffusion bayésienne en section 3.1.3, que l'on décline ensuite dans une version neuronale avec apprentissage de représentation en section 3.2, puis dans une version neuronale récurrente en section 3.3. L'ensemble de ces travaux a permis la définition de modèles probabilistes de diffusion efficaces, qui ouvrent sur diverses perspectives de recherche que l'on envisage en section 3.4.

### 3.1 Diffusion Bayésienne : application et extension du modèle IC

L'étude des phénomènes de propagation sur des populations a d'abord émergé en épidémiologie, pour étudier la transmission des virus. Dans ce contexte, divers modèles macroscopiques ont été proposés pour modéliser l'évolution du nombre d'infections au cours du temps [Daley et al., 2001]. Le modèle SIR (*Susceptible, Infected, Recovered*) proposé dans [Kermack and McKendrick, 1927] est encore largement considéré aujourd'hui, notamment pour la modélisation de la propagation du COVID-19 qui sévit au moment où j'écris ces lignes [Chen et al., 2020]. Ce genre de modèle à compartiments prend la forme d'un système d'équations différentielles, régissant les transitions d'états des individus à l'échelle globale (i.e., sans modélisation des individus). Dans un même esprit, le modèle de Bass [Bass, 1969] a visé l'explication de la façon dont les consommateurs adoptent une innovation. 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. Ce genre de modèle a été appliqué aux réseaux sociaux notamment dans [Luu et al., 2012], où les auteurs observent qu'en pratique la diffusion dépend de la distribution des degrés dans le graphe social considéré, ce qui les conduit à proposer des extensions selon des équations différentielles du ratio d'infectés<sup>1</sup> intégrant une modélisation en loi exponentielle ou loi de puissance des degrés du graphe. Néanmoins, toutes les approches de cette famille ne s'intéressent aux dynamiques qu'à un niveau global, ne permettant pas d'analyse fine des dynamiques en jeu dans la communauté d'intérêt considérée.

De nombreuses autres approches ont donc concerné la modélisation à l'échelle des utilisateurs, ou groupes d'utilisateurs (voir par exemples [Mehmood et al., 2013] ou [Chikhaoui et al., 2015] pour des travaux sur l'identification d'influences intra et inter-communautés). Cette modélisation plus fine permet l'émergence de diverses tâches, parmi lesquelles :

- Prédiction de diffusion [Kempe et al., 2003, Ma et al., 2008, Najjar et al., 2012] : quels utilisateurs seront infectés par tel contenu initié par tel(s) nœud(s) du réseau?
- Détection de sources [Shah and Zaman, 2010] : qui sont les patients zero qui ont initié tel phénomène de propagation?
- Inférence de graphe d'influence [Gomez Rodriguez et al., 2010, Gomez-Rodriguez et al., 2011] : quels sont les principaux canaux de diffusion dans le réseau?
- Identification de *leaders* d'opinion [Kempe et al., 2003, Ma et al., 2008] : quels sont les influenceurs principaux du réseau?
- Maximisation de propagation [Kempe et al., 2003] : à qui donner tel contenu pour en maximiser l'impact?
- Coupure de feu [Anshelevich et al., 2009] : comment faire en sorte de stopper une propagation en cours?

La plupart des travaux dans ce cadre reposent sur des variations autour des modèles fondateurs *Linear Threshold* (LT) [Granovetter, 1978] et *Independent Cascade* (IC) [Goldenberg et al., 2001], que nous détaillons dans la suite.

À noter que les tâches de prédiction de diffusion (à l'échelle des utilisateurs) et les problématiques de recommandation temporelle de produits peuvent être vues comme deux facettes d'un même problème de découverte de corrélations entre nœuds d'un réseau, des utilisateurs dans le cadre de la diffusion et des items dans le cadre de la recommandation, selon des entités avec lesquelles ils interagissent (les items se diffusant pour la diffusion, les utilisateurs clients dans le cadre de la recommandation). Alors que les approches en diffusion visent à extraire des relations d'influence entre utilisateurs des réseaux, la recommandation s'intéresse plutôt à des corrélations (e.g., un client qui achète un marteau, achète aussi généralement des pointes). Cependant, les régularités d'infection observées en diffusion peuvent également correspondre à de simples relations de corrélations temporelles entre utilisateurs connectés, résultant par exemple de phénomènes d'homophilie (i.e., les nœuds connectés du réseau tendent à avoir des comportements similaires en réaction aux mêmes événements extérieurs, sans qu'il n'y ait nécessairement de relations d'influence entre eux). Ces deux aspects peuvent être délicats à distinguer à partir des données [Aral et al., 2009], les deux problématiques diffusion/recommandation se recouvrent largement dans la littérature [Zhang et al., 2007, Cheng et al.,

---

1. Suivant le vocabulaire en épidémiologie classiquement employé dans le domaine de la modélisation de diffusion, le terme d'infection utilisé pour une entité du réseau désigne le fait que cette entité a participé à la diffusion considérée.

2010, Ullah and Lee, 2016]. La plupart des approches en diffusion ignorent la distinction entre influence et corrélation temporelle.

Dans la suite de cette section introductive, nous présentons les modèles de cascade sur lesquels nous nous basons largement dans ce chapitre (section 3.1.1), ainsi que leur utilisation pour l'inférence de graphes de diffusion (section 3.1.2). Enfin, nous présentons en section 3.1.3 une extension du modèle IC, que nous avons proposée dans [Lamprier et al., 2016], afin de permettre l'application efficace du modèle sur des données sociales réelles.

### 3.1.1 Modèles de cascade

Pour simuler des processus de contamination à l'échelle des utilisateurs, les modèles de diffusion LT et IC sont tous deux basés sur un graphe de diffusion connu  $G = (\mathcal{U}, \Theta)$ , avec  $\mathcal{U}$  l'ensemble des nœuds du réseau et  $\Theta$  les liens pondérés du graphe, où  $\theta_{u,v}$  est un poids représentant l'influence de  $u \in \mathcal{U}$  sur  $v \in \mathcal{U}$ . LT est un modèle centré récepteur, qui modélise la pression sociale de chaque nœud du réseau par la somme des influences issues de ses prédécesseurs infectés par le contenu diffusé. Pour chaque diffusion, lorsque la pression sociale d'un nœud  $u \in \mathcal{U}$  dépasse un certain seuil  $s_u \in [0; 1]$ , déterminé aléatoirement pour tout  $u$  à chaque diffusion, le nœud en question passe alors à l'état infecté (modèle de percolation). Le modèle IC est lui un modèle centré émetteur de contenu où, à chaque diffusion, chaque nouvel infecté  $u \in \mathcal{U}$  possède une chance unique d'infecter chacun de ses successeurs  $v \in Succs_u \subseteq \mathcal{U}$  dans le graphe défini par les relations  $\Theta$ , selon une épreuve de Bernoulli indépendante de probabilité  $\theta_{u,v}$ . Le processus de diffusion s'arrête lorsque toutes les possibilités de nouvelle infection sont épuisées. Les infections successives ainsi définies forment des structures nommées cascades de diffusion, d'où le nom *Independent Cascade* du modèle IC. Notons que pour ces deux modèles, on se trouve dans un cadre SI (*Susceptible-Infected*), où chaque nœud ne peut être infecté qu'une seule fois. Ces modèles ont été très largement étudiés dans la littérature, notamment dans [Kempe et al., 2003] qui propose une version généralisée des deux modèles, relaxant l'hypothèse d'indépendance du modèle IC et celle d'additivité des influences du modèle LT. [Kempe et al., 2003] montre alors que toute instance du modèle IC généralisé est équivalente à une certaine instance du modèle LT généralisé, et vice-versa. D'autres extensions concernent notamment la prise en compte du contenu diffusé [Barbieri et al., 2013b] et/ou d'informations disponibles sur les nœuds (profils, types de relations, etc.) [Saito et al., 2011, Wang et al., 2012c, Guille and Hacid, 2012, Lagnier et al., 2013, Lagnier et al., 2018]. Des versions en temps continu ont également été largement envisagées, telles que notamment *CTLT* [Saito et al., 2010], *CTIC* [Saito et al., 2009] ou encore *NetRate* [Gomez-Rodriguez et al., 2011]. Dans ce chapitre, nous nous focalisons principalement sur les modèles de cascade de type IC, qui permettent une modélisation plus fine des liens d'influence entre les utilisateurs sur les réseaux sociaux [Guille et al., 2013], où les répliquions de contenu se font principalement par simple contact à l'information, LT concernant plutôt des phénomènes de diffusion liés à des adoptions d'innovation impliquant des décisions plus coûteuses ou risquées, et où donc les infections se font plutôt par pression sociale [Centola and Macy, 2007].

Selon les modèles de cascade, le calcul exact des probabilités postérieures d'infection de chaque nœud connaissant les sources et le graphe de diffusion est malheureusement un problème #P-complet pour le cas général de graphes de diffusion contenant des cycles [Chen et al., 2010a]. C'est par exemple particulièrement problématique pour des tâches de maximisation d'influence, car il faut considérer toutes les combinaisons de sources et les probabilités postérieures associées pour tous les nœuds, ce qui conduit à un problème #NP-difficile. Diverses approches ont proposé des heuristiques pour simplifier le problème de calcul de ces probabilités. Par exemple, [Kimura and Saito, 2006]

propose une approximation qui se focalise sur les plus courts chemins du graphe de diffusion, alors que [Chen et al., 2010b] simplifie le graphe de diffusion en graphes dirigés acycliques (DAG) locaux en se concentrant sur les relations de poids maximal. [Chen et al., 2010a] définit des arborescences de diffusion en se concentrant sur des chemins d'influence maximale. Une autre manière de travailler est de réaliser des estimations de type Monte-Carlo, en réalisant de multiples simulations et en considérant les ratios d'infection de chaque noeud sur ces simulations comme estimateurs de leur probabilité d'infection postérieure. [Kempe et al., 2003] montre que la simulation en cascade du modèle IC est équivalente à une percolation de liens où l'on conserve à chaque simulation chaque lien  $(u, v)$  selon sa probabilité  $\theta_{u,v}$ . Tous les noeuds atteignables à partir des sources dans le sous-graphe obtenu sont considérés infectés. Bien que bien plus coûteux qu'une simulation en cascade pour une simulation unique, cette procédure présente l'avantage de rendre une grosse partie de l'effort indépendante des sources de diffusion (particulièrement profitable pour les tâches de maximisation d'influence). En outre, elle permet de réduire considérablement la variance des estimateurs de Monte-Carlo [Bóta et al., 2013]. Notons finalement que des approches orthogonales définissent des noyaux de diffusion sur le graphe, tels que [Rosenfeld et al., 2016] pour l'estimation du nombre d'utilisateurs infectés selon IC, ou [Ma et al., 2008] qui définissent un noyau de chaleur sur le graphe en s'appuyant sur les travaux de [Kondor and Lafferty, 2002].

### 3.1.2 Inférence de Graphes de Diffusion

Divers travaux ont concerné l'apprentissage du graphe d'influence utilisé par les modèles de diffusion, basés sur des jeux d'épisode d'entraînement  $\mathcal{D}$ . Typiquement, chaque épisode  $D = (U^D, T^D) \in \mathcal{D}$ , qui correspond à la diffusion d'un item particulier sur le réseau, est défini par un ensemble d'utilisateurs infectés  $U^D \subseteq \mathcal{U}$  et  $T^D = \{t_u^D \in \mathbb{N} + \infty\}_{u \in \mathcal{U}}$  l'ensemble des temps d'infection associés, avec  $t_u^D$  correspondant au temps d'infection de  $u$  pour tout noeud  $u \in U^D$ , et valant  $\infty$  pour les noeuds non infectés dans l'épisode  $D$ . Les temps d'infection dans  $T^D$  sont des temps relatifs par rapport au temps du premier noeud infecté (i.e., la ou les sources de la diffusion, pour qui  $t_u^D$  vaut 1). Dans ce qui suit, on note  $U_{<v}^D$  l'ensemble des noeuds ayant été infecté avant le noeud  $v$  dans la diffusion  $D$  :  $U_{<v}^D = \{u \in \mathcal{U} \mid t_u^D < t_v^D\}$ . Notons que les épisodes correspondent à ce que l'on peut généralement observer d'une diffusion : ils décrivent quels noeuds participent à une diffusion et quand, mais pas les chemins pris par la diffusion. Notons également que nous nous restreignons aux seuls événements observables sur le réseau. C'est à dire par exemple que pour un cas typique où l'on n'est capable que d'observer les publications des utilisateurs d'un réseau social, les noeuds ayant été atteints par un contenu d'intérêt mais ne l'ayant pas re-partagé sur le réseau ne sont pas considérés comme infectés. Enfin, les épisodes considérés dans ce qui suit ne reportent que la première participation de chaque noeud à la diffusion (e.g., si un noeud re-partage plusieurs fois le même contenu, son temps d'infection correspond à l'instant du premier re-partage pour l'épisode correspondant).

Les cascades correspondent à des structures plus riches que les épisodes de diffusion, car elles expliquent la manière dont une diffusion se déroule. Une cascade  $C^D = (U^D, T^D, I^D)$  correspond à un graphe dirigé acyclique partant des sources de la diffusion  $D$  et atteignant l'ensemble des noeuds infectés  $U^D \subseteq \mathcal{U}$ , selon des transmissions depuis des noeuds émetteurs contenus dans  $I^D$ , aux temps d'infection donnés dans  $T^D$ .  $I^D = \{I_u^D \in 2^{U^D}\}_{u \in U^D}$  est donc l'ensemble des infecteurs dans  $D$ , où  $I_u^D$  contient l'ensemble des noeuds ayant réussi à infecter  $u$  (vaut  $\emptyset$  pour les sources de la diffusion). De nombreuses structures de cascade peuvent alors correspondre à un épisode de diffusion observé. Les modèles de cascade émettent généralement des hypothèses sur ces structures de cascade latentes pour l'extraction du graphe de diffusion.

L'objectif des modèles de cascade est donc de définir le graphe de diffusion  $G = (\mathcal{U}, \Theta)$ , avec  $\Theta$  contenant les relations d'influence du réseau. Notons que dans ce qui suit nous ne distinguons pas les relations d'influence (i.e., des relations  $(u \rightarrow v)$  de type " $u$  cause l'infection de  $v$  avec une probabilité  $\theta_{u,v}$ ") de la corrélation temporelle (i.e., des relations  $(u \rightarrow v)$  de type "à l'infection de  $u$  succède celle de  $v$  avec une probabilité  $\theta_{u,v}$ "). Selon les cas,  $\Theta$  peut être restreint à des liens explicites donnés par le réseau social étudié. Cependant, divers travaux ont montré que ces relations explicites sont rarement représentatives des vrais canaux de diffusion sur le réseau [Ver Steeg and Galstyan, 2013, Yang and Leskovec, 2010, Najar et al., 2012]. Par ailleurs, dans de nombreux contextes, ces relations ne sont que très partiellement disponibles. À l'instar de [Gomez-Rodriguez et al., 2011], nous considérons l'inférence sur des graphes complets. Une heuristique consistant à retirer tous les liens  $(u \rightarrow v)$  n'ayant aucun exemple d'épisode avec  $v$  infecté après  $u$  dans l'ensemble d'entraînement peut néanmoins permettre de réduire drastiquement la complexité de l'apprentissage des poids  $\Theta$  selon les cas, sans altérer la solution obtenue pour l'ensemble des algorithmes considérés dans la suite (i.e.,  $\theta_{u,v}$ , si conservé dans  $\Theta$ , convergerait rapidement vers une valeur nulle pour tout lien  $(u \rightarrow v)$  sans exemple de possible diffusion en apprentissage).

Dans ce chapitre, nous nous appuyons très largement sur la méthode d'apprentissage du modèle IC donnée dans [Saito et al., 2008], qui constitue une amélioration de l'approche de [Gruhl et al., 2004], en remplaçant l'hypothèse simplificatrice "exactement un influenceur par infection" par une hypothèse "au moins un influenceur par infection", qui correspond plus fidèlement aux dynamiques du modèle IC, plusieurs transmissions pouvant réussir simultanément. Selon IC, la probabilité pour tout  $v \in \mathcal{U}$  d'être infecté, connaissant un ensemble d'infecteurs potentiels  $I \subseteq \mathcal{U}$ , correspond à la probabilité qu'au moins un nœud  $u \in I$  réussisse à infecter  $v$  :

$$P_{\Theta}(v|I) = 1 - \prod_{u \in I} (1 - \theta_{u,v}) \quad (3.1)$$

Dans le modèle IC classique, lorsqu'une nouvelle infection d'un nœud  $v \in \mathcal{U}$  est provoquée par un nœud  $u \in U^D$  pour une diffusion  $D$  donnée, elle survient au temps  $t_v^D = t_u^D + 1$ . La vraisemblance d'un épisode de diffusion  $D$  est alors donnée par :

$$P_{\Theta}(D) = \prod_{v \in U^D} P_{\Theta}(v|\{u \in U^D | t_u^D = t_v^D - 1\}) \prod_{v \in \mathcal{U}} (1 - P_{\Theta}(v|\{u \in U^D | t_u^D < t_v^D - 1\})) \quad (3.2)$$

où le terme de gauche correspond à la probabilité que tous les infectés de l'épisode le soient à leur temps d'infection connaissant les infectés du pas de temps précédent dans  $D$ , et le terme de droite correspond à la probabilité qu'aucun nœud ne soit infecté avant son temps d'infection observé. Notons que l'on prend arbitrairement  $P_{\Theta}(v|\emptyset) = 1$  pour tout  $v \in U^D$  pour ignorer les infections non expliquées dans les données. Comme énoncé précédemment, ceci aurait pu être contourné de manière alternative à la manière de [Gruhl et al., 2004], en considérant un nœud fictif  $u_0$  représentant le monde extérieur, ajouté dans tout ensemble  $I$  considéré dans l'équation (3.1), permettant alors l'explication de toutes les infections observées, y compris des sources et des nœuds infectés sans prédécesseurs infectés au pas de temps précédant leur infection.

La log-vraisemblance  $\mathcal{L}(\Theta; \mathcal{D})$  d'un ensemble d'épisodes d'apprentissage  $\mathcal{D}$  est alors donnée par :

$$\mathcal{L}(\Theta; \mathcal{D}) = \sum_{D \in \mathcal{D}} \log(P_{\Theta}(D)) = \sum_{D \in \mathcal{D}} \sum_{v \in U^D} \log(P_v^D) + \sum_{v \in \mathcal{U}} \sum_{\substack{u \in U^D, \\ t_u^D < t_v^D - 1}} \log(1 - \theta_{u,v}) \quad (3.3)$$

où  $P_v^D$  est un raccourci pour  $P_{\Theta}(v|\{u \in U^D | t_u^D = t_v^D - 1\})$ . Cette log-vraisemblance est cependant très difficile à optimiser directement, du fait de la formulation de  $P_v^D$  donnée en équation 3.1. Néanmoins,

si nous connaissons quelles tentatives d'infection ont réussi et lesquelles ont échoué dans les processus de diffusion observés, l'optimisation du problème deviendrait alors beaucoup plus simple. Cela motive l'emploi d'un algorithme type *Espérance-Maximisation* (EM), à la manière de [Dempster et al., 1977], en considérant les succès ou échecs des tentatives d'infection comme facteurs latents. L'espérance de log-vraisemblance des épisodes  $\mathcal{D}$ , selon les paramètres courants  $\hat{\Theta}$  pour l'estimation des probabilités de succès des tentatives d'infection, est donnée par :

$$\mathcal{Q}(\Theta|\hat{\Theta}) = \sum_{D \in \mathcal{D}} \Phi^D(\Theta|\hat{\Theta}) + \sum_{v \in \mathcal{U}} \sum_{\substack{u \in U^D, \\ t_u^D < t_v^D - 1}} \log(1 - \theta_{u,v}) \quad (3.4)$$

avec  $\Phi^D(\Theta|\hat{\Theta})$  correspondant à la valeur espérée selon  $\hat{\Theta}$ , pour un épisode de diffusion  $D$ , du premier terme de la log-vraisemblance de  $D$  (i.e.,  $\sum_{v \in U^D} \log(P_v^D)$ ), qui correspond à la log-vraisemblance sur les infections de  $D$  (la vraisemblance sur les non-infections reste inchangée car ne dépendant d'aucun facteur caché). Selon les estimations courantes  $\hat{\Theta}$ , un nœud  $v$  est infecté dans  $D$  au temps  $t_v^D$  avec une probabilité  $\hat{P}_v^D = P_{\hat{\Theta}}(v|\{u \in U^D | t_u^D = t_v^D - 1\})$ . Pour tout  $D \in \mathcal{D}$  et tout  $v \in U^D$ , sachant que  $v$  est infecté au temps  $t_v^D$ , pour tout  $u \in U^D$  tel que  $t_u^D = t_v^D - 1$ , la probabilité conditionnelle  $\hat{P}_{u \rightarrow v}^D$  que l'infection de  $v$  à partir de  $u$  ait réussi est donnée par  $\hat{P}_{u \rightarrow v}^D = \hat{\theta}_{u,v} / \hat{P}_v^D$ . On a alors :

$$\Phi^D(\Theta|\hat{\Theta}) = \sum_{v \in U^D} \sum_{\substack{u \in U^D, \\ t_u^D = t_v^D - 1}} \frac{\hat{\theta}_{u,v}}{\hat{P}_v^D} \log(\theta_{u,v}) + (1 - \frac{\hat{\theta}_{u,v}}{\hat{P}_v^D}) \log(1 - \theta_{u,v}) \quad (3.5)$$

Notons alors que notre problème de maximisation de  $\mathcal{Q}(\Theta|\hat{\Theta})$  se décompose en un ensemble de problèmes de maximisation indépendants, avec un problème individuel par paramètre  $\theta_{u,v} \in \Theta$ . Or, pour chaque  $\theta_{u,v} \in \Theta$ , le problème de maximisation est concave. En annulant la dérivée de  $\mathcal{Q}(\Theta|\hat{\Theta})$  selon chaque paramètre  $\theta_{u,v} \in \Theta$ , on obtient alors la règle de mise à jour suivante :

$$\theta_{u,v} = \frac{\sum_{D \in \mathcal{D}_{u,v}^+} \frac{\hat{\theta}_{u,v}}{\hat{P}_v^D}}{|\mathcal{D}_{u,v}^+| + |\mathcal{D}_{u,v}^-|} \quad (3.6)$$

avec :

$$\mathcal{D}_{u,v}^+ = \{D \in \mathcal{D} | t^D(v) = t^D(u) + 1\} \quad (3.7)$$

$$\mathcal{D}_{u,v}^- = \{D \in \mathcal{D} | t^D(v) > t^D(u) + 1\} \quad (3.8)$$

Alors que  $\mathcal{D}_{u,v}^+$  correspond à l'ensemble des épisodes de  $\mathcal{D}$  tel que  $v$  est possiblement infecté par  $u$  (temps d'infection consécutifs),  $\mathcal{D}_{u,v}^-$  représente l'ensemble des épisodes de  $\mathcal{D}$  tels que l'infection de  $v$  par  $u$  a échoué ( $u$  appartient à  $D$  et  $v$  n'est ni infecté avant  $u$ , ni infecté au temps succédant celui de  $u$ ). Suivant les principes de l'algorithme EM, il est possible de montrer que la répétition de ces étapes d'Espérance (calcul de  $\hat{P}_v^D$  pour tout  $D$  et tout  $v$ ) et de Maximisation (définie en (3.6)), converge vers un optimum local de la log-vraisemblance donnée en (3.3) (en utilisant notamment l'inégalité de Gibbs).

Cet algorithme, proposé par [Saito et al., 2008], est à la base de nombreuses extensions, notamment une version en temps continu *CTIC* donnée dans [Saito et al., 2009], et d'une bonne partie de nos contributions données dans la suite. *CTIC* considère deux paramètres distincts pour chaque couple  $(u, v) \in \mathcal{U}^2$  : une probabilité  $k_{u,v} \in [0; 1]$  d'infection de  $v$  par  $u$  et un paramètre de délai d'infection  $r_{u,v}$ . Un algorithme similaire de type EM est proposé pour ce cas (voir les détails en section

3.3). Notons l'approche *NetInf* [Gomez Rodriguez et al., 2010], qui vise à filtrer un sous-ensemble de liens d'influence dans le graphe obtenu par un modèle type IC ou CTIC, selon des recherches d'arbres couvrants (cascades de diffusion) de poids maximaux pour chaque épisode, puis en exploitant la propriété de sous-modularité du problème d'optimisation associé. D'autres approches d'inférence de graphe de diffusion existent, telles que *NetRate* [Gomez-Rodriguez et al., 2011], qui correspond à une version en temps continu similaire à *CTIC*, mais où un seul paramètre par lien est utilisé pour déterminer les temps d'infection dans chaque épisode. Cette approche présente l'avantage considérable de permettre la formulation du problème d'inférence de graphe sous la forme d'une minimisation convexe, ce qui lui a valu une forte popularité et de nombreuses extensions [Du et al., 2012, Gomez-Rodriguez et al., 2013a, Daneshmand et al., 2014]. Cependant, cette formulation du problème m'est toujours apparue beaucoup plus limitée que le modèle *CTIC*, un noeud  $v$  pouvant être peu fréquemment infecté par un noeud  $u$  mais dans un délai très court (et inversement), ce qui ne peut pas être modélisé dans des modèles synchrones comme *NetRate* et ses extensions. Nos expérimentations sur des problèmes avec données réelles ont confirmé cette intuition, les performances de *NetRate* s'avérant souvent très mauvaises sur ces données. Notons enfin l'approche *InfoPath* [Gomez-Rodriguez et al., 2013b], qui applique *NetRate* de manière séquentielle pour la découverte de relations dans des contextes de distributions d'influence non stationnaires.

### 3.1.3 Application et Extension d'IC

Une limitation importante du modèle IC tel que défini à la section précédente, est qu'il travaille par pas de temps discrets et que toute infection d'un noeud  $v$  dans un épisode ne peut être expliquée que par des infectés au pas de temps précédent  $t_v^D - 1$ . Si cela fonctionne très bien pour l'apprentissage des probabilités d'influence à partir d'épisodes artificiels générés suivant exactement le processus IC sur un graphe de relations donné (comme c'est le cas dans les expérimentations de [Saito et al., 2008]), cela s'avère très problématique pour des données réelles. Tout d'abord, limiter les infections possibles sur des pas de temps contigus correspond à une hypothèse très forte qui peut ne pas s'appliquer car certaines influences peuvent impliquer des délais plus longs que d'autres. D'autre part, cela implique de discrétiser les temps d'infection qui sont souvent continus (ou quasi) dans les données réelles. Le modèle obtenu est alors très dépendant du pas de discrétisation utilisé. Alors que des pas de discrétisation trop grands tendent à regrouper de trop nombreuses infections sur un même pas de temps, ce qui conduit à ignorer de nombreuses possibles transmissions entre infections proches, une discrétisation trop fine peut induire de nombreux pas de temps sans infection, ce qui conduit à de nombreuses infections inexplicables (des infections sans aucun infecté dans le pas précédent). Une possibilité envisageable serait de retirer les pas de temps vides pour l'apprentissage, mais il reste qu'un pas trop petit réduit les possibilités d'explications de chaque infection (jusqu'à une simple chaîne d'infections pour les pas les plus petits). La figure 3.1 illustre cette dépendance au pas de discrétisation choisi (les pas de temps vides ont été retirés de la figure). Avec des pas de temps moyens, de nombreuses structures de cascades sont possibles pour un même épisode observé. Avec des valeurs extrêmes (1 sec ou 10 min) la variété des possibles structures de diffusion latentes est réduite à une simple cascade (une chaîne pour un pas de 1 seconde et un seul groupe pour un pas de 10 minutes).

Les modèles à temps continu énoncés à la section précédente n'ont pas ce genre de problèmes liés à la discrétisation. Cependant, observer des régularités sur les délais d'infection dans les données des réseaux sociaux peut s'avérer difficile, ce qui peut gêner la stabilité de l'apprentissage. Dans [Lamprier et al., 2016], nous avons plutôt proposé une relaxation du modèle IC classique, pour le rendre agnostique aux délais d'infection, généralement sujets à de fortes variations dans les données.

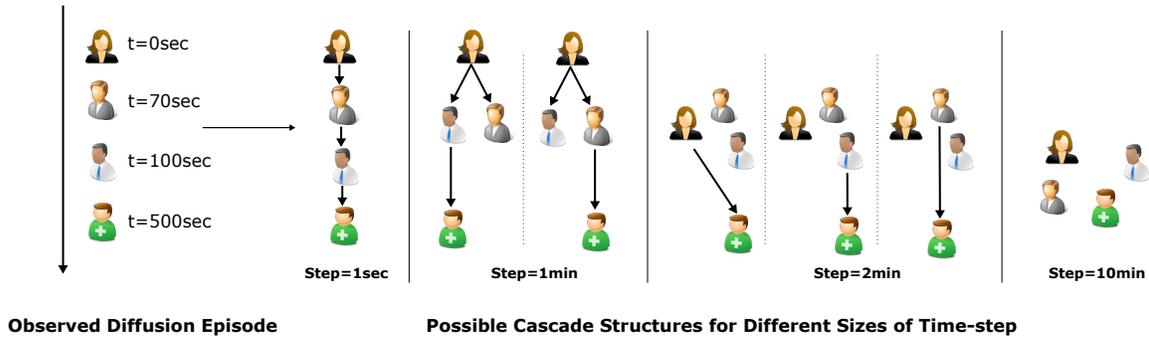


FIGURE 3.1 – Possibles structures de cascade pour un même épisode de diffusion avec différents pas de discrétisation selon  $IC$ . Toutes les structures possibles ne sont pas représentées aux temps 1min et 2min, plusieurs transmissions simultanées pouvant expliquer une même infection.

Ce modèle relâché, que l'on nomme *DAIC* (*Delay-Agnostic Independent Cascade*), ne permet donc pas de modéliser les temps d'infection des utilisateurs, mais il peut s'avérer très efficace pour l'extraction de relations d'influence/corrélation temporelle à partir des données. L'idée est que pour de nombreuses applications, la prédiction des temps d'infection n'est pas essentielle (e.g., prédiction de buzz, identification de leaders d'opinion, prédiction des nœuds finalement impactés par un contenu, etc.). Par ailleurs, la modélisation du délai d'infection peut éventuellement être apprise a posteriori, sur le graphe d'influence extrait.

La relaxation proposée revient simplement à considérer que lorsque l'infection d'un nœud  $v$  survient à partir d'un infecté  $u$  dans un épisode  $D$ , son temps d'infection est choisi uniformément dans  $[t_u^D + 1; t_u^D + T]$ , avec  $T$  un délai maximal, fixé à la durée maximale d'un épisode d'entraînement. Pour l'apprentissage à partir d'épisodes de diffusion, les temps d'infection ne sont plus utilisés que pour ordonner la séquence de diffusion et déterminer l'ensemble d'infecteurs potentiels de chaque infecté (selon (3.1)). La figure 3.2 reprend l'exemple de la figure 3.1 pour souligner la bien plus grande liberté du modèle en apprentissage, l'inférence du graphe de diffusion se faisant en s'appuyant sur un bien plus grand ensemble de structures de diffusion sous-jacentes possibles, tous les prédécesseurs infectés d'un nœud  $v \in U^D$  ayant effectué une tentative d'infection de  $v$ . Concrètement pour l'apprentissage, la relaxation d' $IC$  revient à définir  $\hat{P}_v^D = P_{\hat{\Theta}}(v|U_{<v}^D)$ , avec  $U_{<v}^D$  l'ensemble des nœuds infectés avant  $v$  dans  $D$ , et à modifier les ensembles suivants, le reste de l'algorithme d'apprentissage restant inchangé (y compris la règle de mise à jour (3.6)) :

$$\mathcal{D}_{u,v}^+ = \{D \in \mathcal{D} | t^D(u) < t^D(v) \wedge t^D(v) < \infty\} \quad (3.9)$$

$$\mathcal{D}_{u,v}^- = \{D \in \mathcal{D} | t^D(u) < \infty \wedge t^D(v) = \infty\} \quad (3.10)$$

La relaxation *DAIC* proposée permet donc d'être plus réaliste en supposant des influences possibles entre toutes les paires ordonnées des épisodes de diffusion, tout en évitant les difficultés d'apprentissage liées à la variance des délais d'infection. Néanmoins, cette plus grande liberté met en lumière un biais possible de l'algorithme de [Saito et al., 2008], résultant de représentations déséquilibrées des nœuds (ou utilisateurs) dans l'ensemble des épisodes d'entraînement. Selon la règle de mise à jour 3.6, la présence d'utilisateurs rares dans le corpus d'entraînement peut tendre à cacher certains indices de possible diffusion entre d'autres utilisateurs, pourtant mieux représentés et donc d'estimation d'influence supposée plus fiable. Pour illustrer, nous considérons dans la suite deux propositions qui montrent des exemples de comportement indésirable du processus d'apprentissage de *DAIC*.

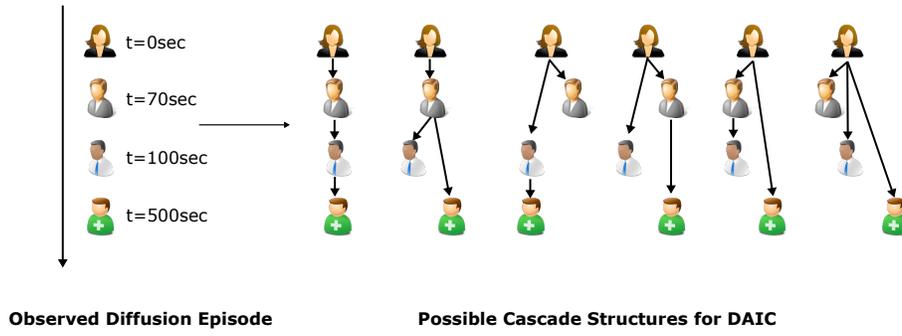


FIGURE 3.2 – Possibles structures de cascade pour un même épisode de diffusion selon DAIC. Là encore on s’est limité à représenter des cascades arborescentes mais d’autres structures seraient possibles, les infections pouvant résulter de plusieurs tentatives de transmission réussies dans DAIC.

**Proposition 7** Pour toute diffusion  $D \in \mathcal{D}$  et tout nœud  $v \in U^D$ , si il existe au moins un nœud  $u \in U_{<v}^D$  tel que  $|\mathcal{D}_{u,v}^-| = 0$ , alors on a :

$$\lim_{n \rightarrow +\infty} P_v^{D^{(n)}} = 1$$

avec  $P_v^{D^{(n)}}$  l’estimation de la probabilité d’infection de  $v$  dans l’épisode  $D$  selon les paramètres courants à la  $n$ -ième itération du processus d’apprentissage.

La démonstration de cette proposition, utilisant notamment un théorème du point fixe sur des suites récurrentes de valeurs de paramètres  $\Theta$ , est donnée en annexe de [Lamprier et al., 2016]. La proposition représente une situation où des infections en “cachent” clairement d’autres dans l’ensemble d’apprentissage  $\mathcal{D}$  : il suffit qu’au moins une relation  $\theta_{u,v}$  vers un nœud  $v$  n’ait pas de contre-exemple de diffusion dans l’ensemble d’entraînement (i.e., aucun épisode  $D$  avec  $u \in U^D$  sans que  $v$  ne soit infecté dans  $D$ , soit  $|\mathcal{D}_{u,v}^-| = 0$ ), pour que la probabilité d’infection de  $v$  converge sûrement vers 1 pour tout épisode où  $u$  est infecté avant  $v$ . Dans ce cas, l’infection de  $u$  est suffisante pour expliquer entièrement l’infection de  $v$ . Selon la règle de mise à jour (3.6), il s’en suit qu’aucune autre relation vers  $v$  ne peut bénéficier d’avoir sa source infectée avant  $v$  dans les épisodes où  $u$  est infecté avant  $v$ .

**Proposition 8** Pour toute relation  $\theta_{u,v} \in \Theta$  telle que  $|\mathcal{D}_{u,v}^-| > 0$ , si il existe dans chaque épisode  $D \in \mathcal{D}_{u,v}^+$  au moins un nœud  $u' \in U_v^D \cap \text{Preds}_v$  tel que  $|\mathcal{D}_{u',v}^-| = 0$ , alors on a :

$$\lim_{n \rightarrow +\infty} \theta_{u,v}^{(n)} = 0$$

avec  $\theta_{u,v}^{(n)}$  la probabilité d’infection de  $v$  par  $u$  selon les paramètres courants à la  $n$ -ième itération du processus d’apprentissage.

Cette proposition, dont la démonstration est plus évidente selon la règle de mise à jour des paramètres et en s’appuyant sur la proposition 7, est plus anecdotique mais permet d’aller plus loin dans l’analyse des cas problématiques de l’apprentissage dans DAIC. Elle indique que, si une paire de nœuds  $(u, v)$  possède au moins un contre-exemple de diffusion (i.e.,  $\mathcal{D}_{u,v}^-$  n’est pas vide), alors tout autre nœud avec aucun contre-exemple de diffusion vers  $v$  peut faire tendre la probabilité de transmission  $\theta_{u,v}$  vers 0. Ainsi, les utilisateurs des réseaux ne participant par exemple qu’une seule fois aux diffusions considérées peuvent considérablement perturber le processus d’apprentissage : toutes les

infections survenant après la leur dans la diffusion où ils apparaissent peuvent être entièrement expliquées par eux si la relation existe dans  $\Theta$ . La figure 3.3 illustre cette situation avec quatre épisodes démarrant de la même source. Alors que l'utilisateur gris est présent dans 3 des 4 épisodes après la source, la probabilité de transmission de la source vers lui converge vers 0, car tous les exemples de diffusion possible sont captés par des utilisateurs isolés.

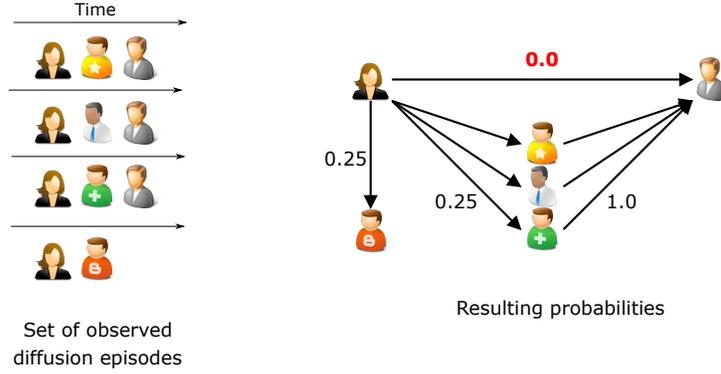


FIGURE 3.3 – Biais de l'apprentissage de *DAIC* en cas de représentations d'utilisateurs déséquilibrés.

Bien que la proposition 8 présentée ci-dessus décrive un cas extrême, qui ne couvre pas l'ensemble des situations problématiques liées aux déséquilibres de représentation des noeuds dans le corpus d'apprentissage (si c'était le cas une solution simple serait de se restreindre aux infections provenant d'utilisateurs avec un nombre minimal de participations sur le réseau), elle est représentative des problèmes de sur-apprentissage pouvant résulter de la formulation d'une probabilité d'infection "au moins un infecteur" telle que définie en (3.1). Ce problème peut également être observé avec le modèle *IC* classique mais est accru dans *DAIC* car les participations des utilisateurs à une diffusion impactent la totalité des événements subséquents dans cette diffusion, plutôt que de se cantonner au simple pas de temps leur succédant.

Pour répondre à ce problème identifié, nous avons proposé dans [Lamprier et al., 2016] de considérer un prior exponentiel sur les probabilités de transmission que l'on manipule, conduisant le modèle à se focaliser sur des canaux de diffusion bien représentés et donc plus fiables. Le problème d'optimisation devient alors l'estimation de maximum a posteriori (MAP) suivante :

$$\Theta^* = \arg \max_{\Theta} \prod_{D \in \mathcal{D}} P_{\Theta}(D) \prod_{\theta_{u,v} \in \Theta} \lambda \exp(-\lambda \theta_{u,v}) = \arg \max_{\Theta} \mathcal{L}(\Theta; \mathcal{D}) - \lambda \sum_{\theta_{u,v} \in \Theta} \theta_{u,v} \quad (3.11)$$

Cette nouvelle formulation du problème permet de prévenir les problèmes énoncés plus haut, car le prior exponentiel ajouté favorise les ensembles de paramètres parcimonieux, à la manière d'une régularisation L1. Cela conduit à des problèmes de résolution de polynômes de degré 2 à chaque étape de maximisation de l'algorithme EM, pour chaque paramètre  $\theta_{u,v} \in \Theta$  :  $\lambda \theta_{u,v}^2 - \beta \theta_{u,v} + \gamma = 0$ , avec  $\beta = (|\mathcal{D}_{u,v}^-| + |\mathcal{D}_{u,v}^+| + \lambda)$ ,  $\gamma = \sum_{D \in \mathcal{D}_{u,v}^+} \hat{\theta}_{u,v} / \hat{P}_v^D$ . Puisque  $|\mathcal{D}_{u,v}^+| \geq \gamma$ , on peut montrer que le discriminant  $\Delta = \beta^2 - 4\lambda\gamma$  est non négatif et donc que le problème possède toujours au moins une solution. On peut également montrer que des mises à jour selon la plus petite racine (i.e.,  $\theta_{u,v} = (\beta - \sqrt{\Delta})/2\lambda$ ) permet l'obtention de probabilités consistantes maximisant l'espérance du MAP selon les paramètres courants à chaque étape de l'algorithme EM (preuves disponibles dans [Lamprier et al., 2016]). Cependant, bien que le prior considéré permette efficacement de contourner le problème de représentations déséquilibrées de noeuds énoncé ci-dessus, il conduit à réduire l'espérance de diffusion dans

le réseau selon le modèle appris. Une possibilité est alors de conclure l'apprentissage effectué jusqu'à convergence selon le MAP donné en (3.11) par une itération de mise à jour classique (i.e., selon (3.6)). Cela permet de revenir à des niveaux de diffusion espérée comparables à ceux des épisodes observés, tout en évitant les problèmes de sur-apprentissage évoqués.

	Digg	ICWSM	Enron	Twitter	Memetracker
IC	0.036	0.097	0.033	0.013	0.012
<i>NetRate</i>	0.102	0.358	0.105	0.027	0.048
CTIC	0.119	0.482	0.132	0.032	0.061
DAIC <sub>0</sub>	<b>0.127</b>	<b>0.665</b>	<b>0.162</b>	0.026	<b>0.073</b>
DAIC <sub>5</sub>	<b>0.128</b>	<b>0.665</b>	<b>0.164</b>	0.035	<b>0.087</b>
DAIC <sub>10</sub>	<b>0.127</b>	<b>0.665</b>	<b>0.164</b>	<b>0.044</b>	<b>0.082</b>

TABLEAU 3.1 – Résultats F1. Les scores en gras sont significativement meilleurs que ceux de *CTIC* (Student-t test à 99%).

**Résultats** La table 3.1 reporte des résultats obtenus sur des corpus réels classiques en diffusion pour différents modèles évoqués ci-dessus. On donne plusieurs versions de  $DAIC_\lambda$ , avec  $\lambda$  le paramètre de la distribution de prior ( $DAIC_0$  fait référence à une version sans prior sur les paramètres). Le “vrai” graphe de diffusion sous-jacent étant inconnu, l'évaluation des modèles se fait sur une tâche de prédiction de diffusion, par simulations de Monte-Carlo en utilisant uniquement les sources des épisodes de test pour initialiser les simulations selon chaque modèle. Les résultats sont donnés en terme de F1 moyen, avec  $F1 = \frac{2 * p * r}{p + r}$  pour chaque simulation, qui est une mesure classique en recherche d'information, où  $p$  correspond à la précision, ratio de vrais positifs dans les infections prédites, et  $r$  au rappel, ratio d'infections à prédire effectivement prédites dans la simulation. Pour chaque modèle, les résultats correspondent à une moyenne sur 1000 simulations par épisode de test. Ils montrent clairement que le modèle IC classique est mal adapté pour des données réelles (malgré un réglage optimal du pas de discrétisation considéré). Le fait d'ignorer les temps d'infection (excepté pour l'établissement des ordres partiels d'infection de  $DAIC$ ) permet d'obtenir de très bons résultats en prédiction d'infection. La régularisation proposée par l'utilisation d'un prior exponentiel paraît permettre de gagner en performances, particulièrement pour les grands jeux de données avec utilisateurs volatiles *Twitter* et *Memetracker*. Plus de détails expérimentaux sont donnés dans [Lamprier et al., 2016].

**Publications associées :**

- Lamprier, S., Bourigault, S., and Gallinari, P. (2016). Influence learning for cascade diffusion models: focus on partial orders of infections. *Social Netw. Analys. Mining*, 6(1):93:1–93:16
- Lamprier, S., Bourigault, S., and Gallinari, P. (2015b). 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, ASONAM 2015, Paris, France, August 25 - 28, 2015*, pages 178–185

## 3.2 Apprentissage de représentation pour la diffusion dans les réseaux

Afin d'éviter les biais d'apprentissage dues aux trop grandes libertés données aux paramètres des liens de diffusion, une autre possibilité est de plonger le réseau d'utilisateurs dans un espace métrique continu, et d'exploiter les contraintes liées à la géométrie de l'espace pour gagner en capacité de généralisation. Cette idée est très exploitée en recommandation, notamment autour des approches de filtrage collaboratif [Su and Khoshgoftaar, 2009], afin d'exploiter les similarités de comportement des utilisateurs pour en extraire des régularités. L'hypothèse de ces approches est de se dire que des utilisateurs ayant aimé des produits similaires par le passé auront tendance à aimer les mêmes autres produits. Cette idée a été déclinée de multiples façons dans la littérature en recommandation, y compris avec des approches de factorisation matricielle intégrant des prises en compte de relations entre utilisateurs [Ma et al., 2018], mais étonnamment très peu en diffusion avant notre première contribution dans [Bourigault et al., 2014b].

Cette section présente le contexte de l'apprentissage de représentation et une première proposition d'application à la modélisation de diffusion en section 3.2.1. Une version du modèle de cascade IC (ou plus précisément de son extension DAIC décrite à la section précédente), dont les probabilités de diffusion sont issues de représentations distribuées des nœuds du réseau considéré, est ensuite proposée en section 3.2.2.

### 3.2.1 Apprentissage de représentation et diffusion

**Apprentissage de représentations** Au delà des approches de factorisation matricielle, l'apprentissage de représentations (*Representation Learning*), basé notamment sur des architectures neuronales, a connu un très large essor au cours de la dernière décennie, pour permettre la prise en compte efficace de dépendances relationnelles complexes entre les données [Bengio et al., 2013]. Le principe général est de projeter des entités d'intérêt dans un espace de représentation (généralement  $\mathbb{R}^d$ ), de manière à ce que les proximités dans l'espace modélisent une ou plusieurs relations existant entre les entités en dehors de cet espace. Ces représentations distribuées présentent plusieurs avantages, à la fois pour la compacité des modèles (relations encodées dans l'espace en  $N \times d$  paramètres, plutôt que  $N \times N$  poids par relation dans un graphe complet classique) et la régularisation des relations apprises induite par les contraintes géométriques de l'espace considéré. Faisant suite au positionnement multidimensionnel (*MDS*) introduit dans [Kruskal, 1964], qui vise à définir des représentations dont les distances correspondent à des distances connues entre entités, l'apprentissage de représentations (et de métriques associées) a été appliqué à de très nombreux problèmes, pour représenter des données complexes dans des espaces plus simples à manipuler. De très nombreuses approches ont depuis été proposées pour divers types de données (e.g., graphes hétérogènes [Dos Santos et al., 2018], données textuelles [Bengio et al., 2003, Mikolov et al., 2013], images [Chen et al., 2016b], etc.) et dans divers contextes (e.g., apprentissage supervisé, semi-supervisé [Clark et al., 2018] ou non supervisé avec objectifs d'interprétabilité [Chen et al., 2016b], apprentissage par renforcement [Nachum et al., 2018], etc.). Un exemple très populaire d'apprentissage de représentation est donné dans [Mikolov et al., 2013], avec les algorithmes *CBOW* et *Skip-Gram*, donnant lieu à la très célèbre représentation *Word2Vec* utilisée dans de nombreux travaux de modélisation de la langue. Depuis, une large série d'approches "Truc2Vec" ont été proposées dans la littérature, avec notamment *Doc2Vec* [Le and Mikolov, 2014] pour la représentation de documents textuels, *Item2Vec* [Barkan and Koenigstein, 2016] pour la recommandation de produits, *Node2Vec* [Grover and Leskovec, 2016] pour la représentation de nœuds de graphes et même *Cas2Vec* [Kefato et al., 2018] pour la diffusion d'information d'un point

de vue macroscopique (basée sur des évolutions de volumes d'infections).

**Modèle de diffusion de chaleur** Outre une approche en recommandation de playlist dans [Chen et al., 2012], qui reste dans un contexte de recommandation temporelle proche mais quelque peu déconnecté de la diffusion d'information, à notre connaissance notre contribution dans [Bourigault et al., 2014b] était le premier travail à employer l'apprentissage de représentation pour la prédiction de diffusion dans les réseaux. L'idée y était d'apprendre, pour chaque nœud  $i$  du graphe de diffusion à définir, une représentation récepteur  $\omega_i \in \mathbb{R}^d$  et une représentation émetteur  $z_i \in \mathbb{R}^d$ , de manière à ce que l'information émise à partir d'une source  $j$ , partant de  $z_j \in \mathbb{R}^d$ , atteigne les représentations récepteur des nœuds selon leur ordre chronologique d'infection, en considérant que l'information se propage dans l'espace de représentation continu suivant l'équation de la chaleur. Pour une chaleur initiale nulle en tout point de l'espace, sauf en la source où toute la chaleur est concentrée (sous la forme d'un Dirac en ce point), le noyau  $K_{\mathcal{Z},\Omega}(t, s_D, u) = (4\pi t)^{-\frac{d}{2}} \exp(-\frac{\|z_{s_D} - \omega_u\|^2}{4t})$  retourne la chaleur au point  $\omega_u$  au temps  $t$  (i.e., représentant la propension de  $u$  à être infecté) selon les ensembles de représentations émetteur  $\mathcal{Z}$  et récepteur  $\Omega$ , sachant que l'information est partie du nœud  $s_D$  au temps 0. Ce noyau indique simplement qu'il s'agit de placer plus proches de la source les nœuds les plus rapidement atteints par l'information (selon une distance euclidienne). Basé sur ce principe, l'apprentissage du modèle (i.e., apprentissage des représentations des nœuds) est réalisé par descente de gradient stochastique selon un coût d'ordonnancement *pairwise*. Notons que l'apprentissage de deux représentations par nœud du graphe, une en tant qu'émetteur et une en tant que récepteur, permet de définir des tendances de diffusion asymétriques, sachant que le graphe de diffusion sous-jacent est dirigé (l'influence de A sur B n'est pas nécessairement équivalente à l'influence de B sur A). Des extensions avec prise en compte de la nature de la diffusion ont également été considérées dans [Bourigault et al., 2014a], avec translation de la source de chaleur ou déformation de l'espace de diffusion (via une distance de Mahalanobis apprise) en fonction du contenu diffusé.

Cette approche nous avait permis d'obtenir de bons résultats en terme de *Mean Average Precision*, qui est une mesure d'évaluation populaire en recherche d'information qui considère la propension des documents pertinents à être classés en tête de la liste de résultats fournie. Ici on ordonne les nœuds en fonction de leur distance à la source et on applique la mesure sur la liste ordonnée, en fonction des vrais infectés des épisodes de test. Un grand avantage de l'approche est le nombre de paramètres bien inférieur à celui d'un modèle comme *IC*, qui rencontre des problèmes computationnels importants dans le cas de grands graphes de diffusion denses. D'autre part, l'utilisation en prédiction est bien plus rapide car ne requiert pas des estimations de Monte-Carlo ou autre processus coûteux d'approximation de distributions d'infection des différents nœuds du réseau considéré : il suffit de regarder les distances relatives à la source. Cependant ce premier modèle présente diverses limitations, au premier rang desquelles le fait qu'il ne soit pas possible de prédire quels nœuds seront finalement infectés, ni le volume de la diffusion, en fonction des sorties du modèle. La prédiction émise par le modèle correspond en effet à une simple liste ordonnée, qui indique quels nœuds seront plus probablement infectés avant quels autres, mais ne permet pas de connaître les niveaux de probabilité d'infection des nœuds. Par ailleurs, on pourrait émettre un reproche similaire à celui mentionné pour *NetRate* en section 3.1.2 : ce n'est pas parce qu'une infection survient dans un délai plus court qu'elle est plus probable. Il y a ici aussi une forme de confusion temps-probabilité d'infection, qui peut être problématique pour l'apprentissage de relations d'influence consistantes. Le modèle ne permet par ailleurs pas de prédire de temps d'infection. Enfin, dans la version de [Bourigault et al., 2014a], les prédictions ne pouvaient être effectuées que selon le conditionnement à une seule source

de diffusion. Or, il peut être intéressant d'être capable de prédire le futur d'une diffusion connaissant son début, quel que soit le nombre d'infectés connus. Une version avec multiples sources est discutée en perspectives (section 3.4.2.2).

**Objectifs pour la diffusion sociale** Malgré ces diverses limitations, l'utilisation de représentations apprises pour les noeuds du réseau reste pertinente pour des tâches liées à la diffusion sur les réseaux, au delà du plus faible nombre de paramètres des modèles correspondants. Le fait de considérer des graphes complets comme dans les modèles de cascade de la section précédente revient à ignorer de nombreuses propriétés spécifiques des relations entre utilisateurs dans les réseaux sociaux [Mislove et al., 2007] (e.g., distribution des degrés en loi de puissance, faible diamètre, etc.). Dans le cadre de la diffusion, [Barbieri et al., 2013a] distingue deux types de communautés. D'une part les communautés cohésives, au sein desquelles les noeuds interagissent beaucoup entre eux. D'autre part les communautés bimodales, qui sont formées par des noeuds interagissant avec les mêmes groupes d'autres noeuds. Ces deux types de communautés ont un impact sur la diffusion d'information [Barbieri et al., 2013a, Yang et al., 2014], et donc sur les régularités que l'on devrait extraire des données d'entraînement. Les communautés cohésives impliquent des régularités triangulaires du type de celle représentée à gauche de la figure 3.7 : Si  $a$  a forte influence d'un noeud  $a$  vers un noeud  $b$  d'une part, et de  $b$  vers un noeud  $c$  d'autre part, alors l'influence de  $a$  vers  $c$  est vraisemblable. Les communautés bimodales impliquent des régularités du type de celle à droite de la figure 3.7 : Si on observe une forte influence d'un noeud  $a$  vers deux noeuds  $b$  et  $c$ , et que l'on observe également une forte influence d'un autre noeud  $d$  sur  $b$ , alors l'influence de  $d$  sur  $c$  est vraisemblable. Ces propriétés, qui n'étaient pas du

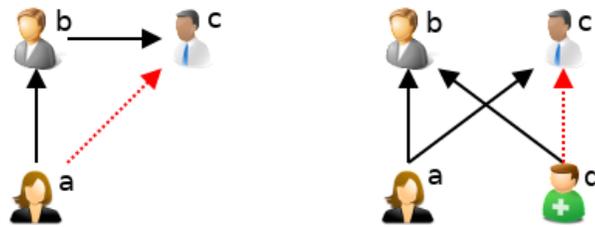


FIGURE 3.4 – Régularités sur les relations des utilisateurs. Les liens rouges sont des relations vraisemblables connaissant les relations d'influence en noir.

tout prises en compte dans l'apprentissage de modèles de cascades explicites avec poids individuels appris sur chaque lien comme des variables libres, sont réinstituées dans le cadre de modèles basés sur l'apprentissage de représentations distribuées, grâce aux contraintes géométriques de l'espace de représentation, telles que l'inégalité triangulaire de toute distance dans un espace euclidien. Dans la section suivante nous présentons une approche de cascade du type de *DAIC*, mais dans un cadre avec représentations distribuées des noeuds considérés, présentée dans [Bourigault et al., 2016d].

### 3.2.2 Modèle de Cascade à représentations distribuées

À la manière des modèles de cascade de type *IC*, notre objectif est de définir un modèle explicatif, qui représente les relations de diffusion entre noeuds d'un graphe de diffusion donné, à partir duquel il est possible de simuler des cascades de diffusion selon un processus génératif probabiliste. Néanmoins, contrairement aux modèles de cascades classiques, dans lesquels les relations considérées sont explicites, nous considérons ici des relations de diffusion définies implicitement, en fonction

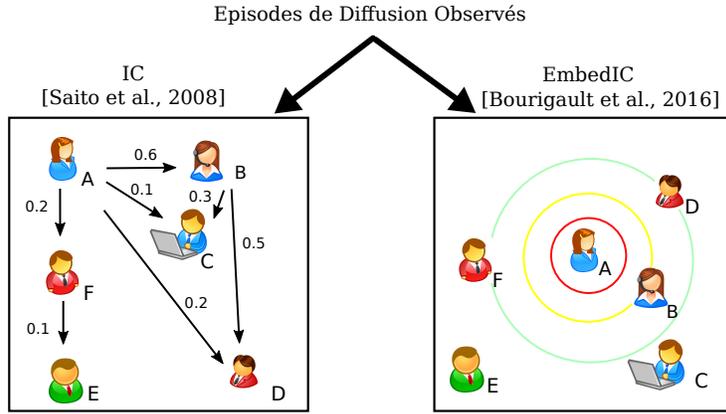


FIGURE 3.5 – Apprentissage d'un graphe de diffusion à partir d'épisodes d'entraînement. À gauche, la version classique sur graphe explicite. À droite, notre approche de graphe de diffusion intégré à un espace de représentation (une seule représentation émetteur-récepteur ici pour simplifier). Les cercles représentent des lignes de niveau d'équiprobabilité de diffusion depuis l'utilisatrice A.

de positions relatives des noeuds dans un espace de représentation. Comme dans le modèle à diffusion de chaleur discuté à la section précédente, nous considérons deux ensembles de représentations  $\mathcal{Z} = (z_i)_{u_i \in \mathcal{U}}$ , pour les émetteurs de contenu, et  $\Omega = (\omega_j)_{u_j \in \mathcal{U}}$ , pour les récepteurs de contenu, de manière à modéliser un graphe de diffusion dirigé. L'idée est alors de définir les probabilités de transmission  $\theta_{i,j}$  du modèle *IC* (ou *DAIC*) selon une fonction  $f : \mathbb{R}^d \times \mathbb{R}^d \rightarrow [0, 1]$  définie dans l'espace de représentation, avec  $\theta_{i,j} = f(z_i, \omega_j)$ . Dans [Bourigault et al., 2016d], nous proposons de considérer la fonction logistique suivante, permettant de définir des probabilités décroissantes en fonction des distances séparant les représentations émetteur  $\mathcal{Z}$  et récepteur  $\Omega$  dans l'espace de projection :

$$f(z_i, \omega_j) = \frac{1}{1 + \exp(z_i^{(0)} + \omega_j^{(0)} + \sum_{x=1}^{d-1} (z_i^{(x)} - \omega_j^{(x)})^2)} \quad (3.12)$$

où  $z^{(x)}$  est la  $x$ -ème composante du vecteur  $z$ . Notons que nous avons choisi de considérer la première composante de chaque représentation comme une valeur de biais, où  $z_i^{(0)}$  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. Cela permet également de définir des valeurs de probabilité supérieures à 0.5, malgré la non-négativité de la distance entre représentations. De par sa forme en S, la fonction logistique est plus fortement impactée par des variations survenant sur les distances modérées, tombant dans la partie de plus forte pente de la fonction, qu'à ses deux extrêmes. Cela permet de focaliser l'effort d'apprentissage sur les influences les moins évidentes du graphe de diffusion.

Comme illustré dans la figure 3.5, l'objectif est donc d'apprendre ce graphe de diffusion implicite à partir d'un ensemble d'apprentissage d'épisodes de diffusion observés, à la manière de l'algorithme d'apprentissage considéré pour *IC* dans [Saito et al., 2008], ou plus précisément notre extension *DAIC* présentée à la section précédente (version sans le prior sur les pondérations). Il s'agit alors de considérer l'espérance de vraisemblance selon les représentations courantes, à chaque étape de l'apprentissage, donnée comme pour *DAIC* par :

$$\mathcal{Q}(\Theta|\hat{\Theta}) = \sum_{D \in \mathcal{D}} \sum_{v \in U^D} \sum_{u \in U_v^D} \frac{\hat{\theta}_{u,v}}{\hat{p}_v^D} \log(\theta_{u,v}) + \left(1 - \frac{\hat{\theta}_{u,v}}{\hat{p}_v^D}\right) \log(1 - \theta_{u,v}) + \sum_{v \in \mathcal{U} \setminus U^D} \sum_{u \in U^D} \log(1 - \theta_{u,v}) \quad (3.13)$$

Corpus	Model	LogLikelihood	MAP	nbParams
MemeTracker	DAIC	-795,85	0,22	229073
	NetRate	-850,48	0,17	229073
	CTIC	-802,52	0,22	458146
	EmbedIC	<b>-791,3</b>	<b>0,23</b>	24900
Digg	DAIC	-69,5	0,411	689416
	NetRate	-64,01*	0,409	689416
	CTIC	-64,18*	0,413	1378832
	EmbedIC	<b>-51,75*</b>	<b>0,434*</b>	164750
Twitter	DAIC	-412,75	0,047	884832
	NetRate	-428,78	0,039	884832
	CTIC	-401,56	0,049	1769664
	EmbedIC	<b>-223,15*</b>	<b>0,056*</b>	142050
LastFM	DAIC	-409,5	0,132	708159
	NetRate	-413,02	0,112	708159
	CTIC	-409,3	0,128	1416318
	EmbedIC	<b>-405</b>	<b>0,151*</b>	49300

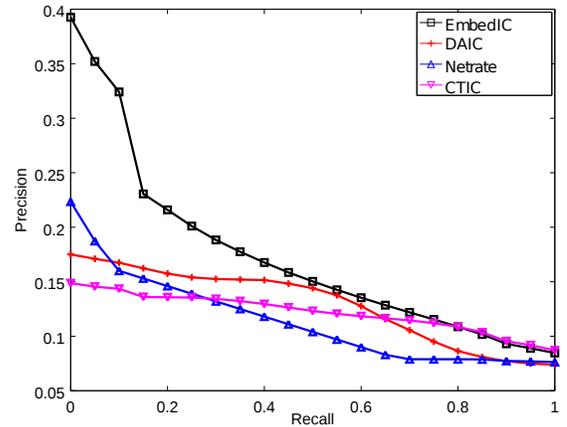


FIGURE 3.6 – Résultats de *EmbedIC*. À gauche résultats en prédiction sur les épisodes de tests. À droite, courbes précision-rappel de découverte de liens de diffusion sur le corpus MemeTracker.

avec  $\Theta = (\mathcal{Z}, \Omega)$  l'ensemble des paramètres du modèles,  $\hat{\Theta} = (\hat{\mathcal{Z}}, \hat{\Omega})$  les valeurs courantes des représentations  $\mathcal{Z}$  et  $\Omega$ , et  $\theta_{u,v} = f(z_u, \omega_v)$ .

Il est alors possible d'établir un algorithme d'apprentissage similaire à celui de *DAIC*. Néanmoins, 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  $\mathcal{Q}(\Theta|\hat{\Theta})$  ne peut alors plus se décomposer en un ensemble de sous-problèmes concaves comme cela était le cas avec *DAIC*. On n'a alors plus de solution analytique à chaque étape de maximisation. Il est cependant possible de mettre en oeuvre un algorithme de type *GEM* (*Generalized Expectation Maximization*), [Dempster et al., 1977] ayant montré que pour assurer la convergence vers un maximum local, il n'est pas indispensable de maximiser  $Q$  à chaque étape du *EM*, une simple amélioration étant suffisante. L'étape de maximisation est alors définie selon un pas de montée de gradient selon les représentations impliquées dans les différentes diffusions considérées. Nous définissons un algorithme d'apprentissage stochastique : À chaque itération, un épisode  $D$  et un nœud  $v \in \mathcal{U} \setminus \{u \in U^D | t_u^D = 0\}$  sont échantillonnés uniformément. On calcule alors le gradient de  $\mathcal{Q}(\Theta|\hat{\Theta})$  selon la représentation réceptrice de  $v$  et toute représentation émettrice d'un nœud  $u \in U_v^D$ , pour cet épisode et en ne considérant que les liens de diffusion vers  $v$ . Les gradients sont alors ajoutés aux représentations correspondantes. Un soin particulier est attaché au respect des volumes d'implication des différentes représentations de la formule 3.13, par application de facteurs de correction aux gradients considérés pour corriger des biais issus de l'échantillonnage uniforme. L'algorithme complet est donné dans [Bourigault et al., 2016d], que l'on pourrait étendre aisément pour travailler par mini-batches. Notons qu'une alternative à l'algorithme *GEM* considéré dans [Bourigault et al., 2016d] aurait pu être de se baser sur le changement de variable  $\beta_{u,v} = -\log(1 - \theta_{u,v})$  proposé dans [Netrapalli and Sanghavi, 2012], et la modélisation de  $\beta_{u,v}$  par  $f(z_u, \omega_v)$  avec  $f$  une fonction positive décroissante selon la distance entre  $z_u$  et  $\omega_v$ , pour considérer un algorithme de montée de gradient stochastique plus standard (mais avec risques d'instabilité accrus), basé sur la vraisemblance

$$\mathcal{L}(\beta; \mathcal{D}) = \sum_{D \in \mathcal{D}} \sum_{v \in U^D} \log(1 - \exp(-\sum_{u \in U_v^D} \beta_{u,v})) - \sum_{v \in \mathcal{U} \setminus U^D} \sum_{u \in U^D} \beta_{u,v}.$$

**Résultats** La figure 3.6 donne un échantillon de résultats sur des corpus classiques en diffusion. La table de gauche donne, pour les différents modèles, des résultats de log-vraisemblance en test

connaissant les sources, ainsi que des résultats en *Mean Average Precision (MAP)*, qui considère la moyenne des précisions prises après chaque vrai infecté des listes ordonnées selon des estimations Monte Carlo des probabilités d'infection. Le graphique de droite montre des courbes précision-rappel de découverte de liens explicites connus sur le corpus MemeTracker. Dans les deux cas, on observe les bonnes performances de l'approche *EmbedIC*, pour un nombre de paramètres largement inférieur aux approches classiques (colonne *nbParams*). Les régularités extraites grâce aux contraintes de l'espace de représentation ont permis de découvrir des relations de diffusion plus efficacement qu'avec les approches classiques, sujettes au sur-apprentissage. Notons par ailleurs que les biais d'apprentissage évoqués à la section 3.1.3 (dus aux déséquilibres de représentation des différents nœuds dans l'ensemble d'entraînement) se trouvent considérablement limités lorsque l'on travaille avec des représentations continues des nœuds, justement grâce aux contraintes géométriques de l'espace qui impliquent une considération globale des relations d'influence.

Notons qu'une autre contribution, pour la détection de sources dans les processus de diffusion, nous a également permis d'obtenir de bons résultats par apprentissage de représentation [Bourigault et al., 2016c], avec des performances bien plus stables que les approches graphiques classiques [Shah and Zaman, 2012, Pinto et al., 2012, Farajtabar et al., 2015], par ailleurs souvent plus complexes. Dans cette approche l'idée est de faire correspondre une représentation commune de la diffusion (typiquement un barycentre des représentations individuelles des infectés observés, pondéré par des importances apprises sur les utilisateurs infectés) avec la représentation émetteur de la source. En test, l'idée est de simplement retourner les candidats de représentation émetteur les plus proches de la représentation commune comme possibles sources de la diffusion considérée. Des résultats intéressants ont également été observés dans une version avec intégration du contenu de la diffusion (par translation de la représentation commune selon une fonction du contenu).

#### Publications associées :

- Bourigault, S., Lamprier, S., and Gallinari, P. (2018). Détection de sources de diffusion par apprentissage de représentations distribuées. *Document Numérique*, 21(3):11–31
- Bourigault, S., Lamprier, S., and Gallinari, P. (2017). Apprentissage de représentation pour la détection de source dans les réseaux sociaux. In *CONFérence en Recherche d'Informations et Applications - CORIA 2017, 14th French Information Retrieval Conference, Marseille, France, March 29-31, 2017. Proceedings*, pages 235–250
- Bourigault, S., Lamprier, S., and Gallinari, P. (2016c). Learning distributed representations of users for source detection in online social networks. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II*, pages 265–281
- Bourigault, S., Lamprier, S., and Gallinari, P. (2016d). 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, San Francisco, CA, USA, February 22-25, 2016*, pages 573–582
- Bourigault, S., Lamprier, S., and Gallinari, P. (2016b). Apprentissage de représentations probabilistes pour la prédiction de diffusions d'informations sur les réseaux sociaux. In *CORIA 2016 - CONFérence en Recherche d'Informations et Applications - 13th French Information Retrieval Conference. CIFED 2016 Colloque International Francophone sur l'Écrit et le Document, Toulouse, France, March 9-11, 2016*, pages 89–104

- Bourigault, S., Lamprier, S., and Gallinari, P. (2016a). Apprentissage de représentations pour la modélisation de processus de diffusion dans les réseaux sociaux. *Document Numérique*, 19(2-3):31–52
- Bourigault, S., Lagnier, C., Lamprier, S., Denoyer, L., and Gallinari, P. (2014b). Learning social network embeddings for predicting information diffusion. In *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, pages 393–402
- Bourigault, S., Lagnier, C., Lamprier, S., Denoyer, L., and Gallinari, P. (2014a). Apprentissage de représentation pour la diffusion d'information dans les réseaux sociaux. In *CORIA 2014 - Conférence en Recherche d'Informations et Applications- 11th French Information Retrieval Conference. CIFED 2014 Colloque International Francophone sur l'Écrit et le Document, Nancy, France, March 19-23, 2014*, pages 155–170
- Lagnier, C., Bourigault, S., Lamprier, S., Denoyer, L., and Gallinari, P. (2014). Learning information spread in content networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*

### 3.3 Modèles neuronaux récurrents pour la diffusion dans les réseaux

Les modèles de cascade classiques, ainsi que notre contribution *EmbedIC* basée sur de l'apprentissage de représentation, s'appuient sur une hypothèse de Markov sur les chemins empruntés par le contenu au cours de la diffusion : les prochaines infections de la diffusion en cours ne dépendent que des nœuds infectés à l'instant courant, et pas des trajectoires empruntées par le contenu se diffusant. Or, l'historique de la diffusion peut contenir des informations importantes pour la prédiction de son futur. Dans de nombreuses applications, l'objet de la diffusion est soit inconnu (du fait par exemple de contraintes de confidentialité ou de limitations techniques) ou ses caractéristiques disponibles ne sont pas aisément exploitables par les modèles (e.g., faible corrélation avec les tendances de diffusion, contenu fortement bruité, large variabilité, etc.). Aussi, le contenu transitant sur le réseau peut évoluer au cours de la diffusion, en fonction par exemple des nœuds par lesquels il a transité, ce qui complique sa prise en compte dans les modèles. Pour tous ces cas, la considération des trajectoires passées peut donner des indices utiles sur la nature de la diffusion en cours. La figure 3.7 illustre deux types de processus de diffusion survenant sur le même réseau de propagation, que les modèles de cascade de type *IC* ne peuvent pas distinguer. Le réseau représenté contient un nœud *hub* par lequel transite tous les contenus diffusés (le nœud C), ce qui est une configuration fréquente sur les réseaux sociaux (i.e., réseaux souvent *scale-free*, où les degrés des nœuds suivent une loi de puissance). Dans les deux types de processus, l'information transite donc par le nœud C, mais lorsque l'information est initiée par le nœud A, c'est plutôt le nœud D qui est infecté par C, alors que pour les informations émises par le nœud B la tendance d'infection est plutôt pour le nœud E (cela s'étend naturellement à des groupes de nœuds A,B,C,D,E plutôt que des nœuds individuels). Afin d'être à même de traiter ce genre de cas, il s'agit de dépasser l'hypothèse commune markovienne des modèles de cascade. Les réseaux de neurones récurrents sont de bons candidats pour répondre à cet objectif, mais leur application pour la prise en compte des dépendances complexes des processus de diffusion n'est pas évidente.

Dans la suite, on discute en section 3.3.1 des différentes possibilités pour dépasser le cadre markovien des modèles de cascade, en présentant notamment les approches de la littérature basées sur des réseaux de neurones récurrents pour la modélisation de processus de diffusion. La section 3.3.2

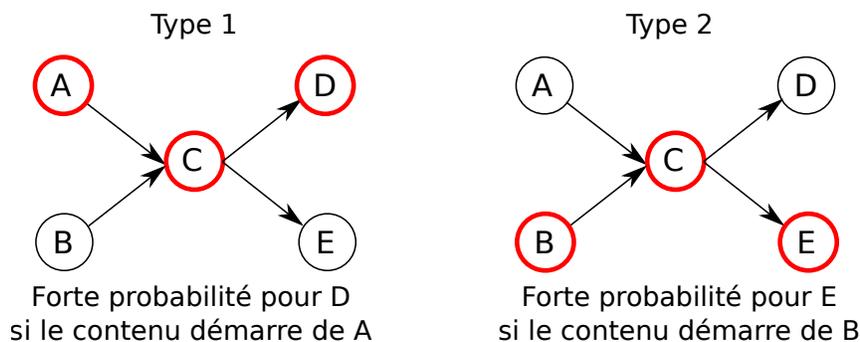


FIGURE 3.7 – Exemple de limitation pour les modèles markoviens. 2 types de diffusion sur le même graphe d'influence, que des modèles de type *IC* ne peuvent pas distinguer pour prédire les infections après le nœud hub C. Les nœuds entourés en rouge sont les nœuds infectés, les flèches sont les liens d'influence.

présente ensuite une contribution récente sur ce thème, basé sur le modèle CTIC et l'inférence bayésienne de trajectoires pour extraire les dynamiques de diffusion dans les réseaux étudiés.

### 3.3.1 Réseaux de neurones récurrents pour la diffusion

Diverses approches récentes sont basées sur des réseaux de neurones récurrents (RNN) pour prédire le futur des diffusions selon leur passé, et ainsi dépasser les hypothèses de Markov des modèles de cascade. Le principe des RNNs est d'appliquer séquentiellement le même bloc neuronal à chaque composante d'une séquence d'entrée, en considérant à chaque nouvelle application une mémoire produite au pas de temps précédent. Ce mécanisme les rend aisément applicables à des séquences de tailles variables, tout en leur permettant de prendre en compte des dépendances long-terme. Depuis leur introduction, les RNNs ont été très largement utilisés pour diverses applications impliquant des séquences, telles que la reconnaissance de la parole [Graves and Schmidhuber, 2005, Graves et al., 2013], la modélisation de la langue [Mikolov et al., 2010, Mikolov and Zweig, 2012, Sutskever et al., 2011, Cho et al., 2014], la prédiction de vidéos [Shi et al., 2015, Wang et al., 2017b], et bien d'autres. Ils sont aussi notamment très utilisés en apprentissage par renforcement, notamment dans les environnements partiellement observables (*POMDPs*), pour lesquels la prise en compte de la séquence d'observations passées peut être utile à l'inférence de l'état courant [Wierstra et al., 2010]. Diverses implémentations des RNNs ont été proposées dans la littérature, les plus célèbres étant les modèles LSTM (*Long-Short Term Memory*) [Hochreiter and Schmidhuber, 1997] et GRU (*Gated Recurrent Unit*) [Cho et al., 2014], proposés pour limiter les problèmes usuels d'effondrement ou d'explosion du gradient dans les RNNs [Bengio et al., 1994, Pascanu et al., 2013].

Dans le cadre de la diffusion, une possibilité d'application naïve serait de considérer les épisodes de diffusion comme des séquences. Dans cette veine, le processus temporel RMTTP de [Du et al., 2016] (*Recurrent Marked Temporal Point Process*) pourrait être considéré pour capturer les dynamiques de diffusion des réseaux, avec l'état caché du RNN contenant l'historique de la diffusion, afin d'être capable de prédire, pour chaque instant, qui sera prochainement infecté et quand. Cependant, les diffusions ne sont pas des séquences, les épisodes observés résultent de dépendances arborescentes comme illustré dans la figure 3.8, des méthodes récurrentes naïves ne sont alors que peu efficaces pour capturer les dynamiques sous-jacentes aux infections observées. Inclure la totalité de l'historique de diffusion dans un état latent pour chaque nouveau nœud infecté, plutôt que de se restreindre aux seuls événements dont son infection découle, implique la considération de nombreuses infec-

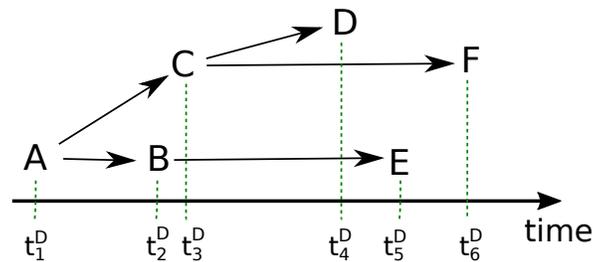


FIGURE 3.8 – Dépendances arborescentes dans les processus de diffusion. Par exemple, l’infection du nœud F ne dépend que de celle des nœuds C et A, pas de la totalité de l’historique, dont la prise en compte pourrait perturber le modèle. Le fait que D soit infecté avant ou après E, ou même simplement que B, D et E soient infectés, ne devrait pas impacter la probabilité d’infection de F à partir de C.

tions indépendantes, dont les occurrences ne sont pas directement corrélées avec la branche de diffusion du nœud en question. Un modèle qui se focaliserait sur les vrais chemins pris par la diffusion en cours paraîtrait plus pertinent. Si ces chemins de diffusion étaient connus, il conviendrait simplement d’adapter des modèles neuronaux récurrents pour des structures de dépendances arborescentes, comme proposé dans [Tai et al., 2015] pour des tâches de traitement du langage naturel (NLP). Ce n’est malheureusement pas le cas, la plupart du temps la topologie de la diffusion est inconnue, les épisodes d’entraînement ne contenant que des instants d’infection de nœuds du réseau considéré.

Pour répondre à cette difficulté, [Wang et al., 2017a] s’appuie sur un graphe de relations connues pour proposer un modèle récurrent, *Topo-LSTM*, dans lequel chaque nœud infecté est associé à un état latent, qui est construit selon les branches d’infection dont il dépend (moyenne des états de chaque branche d’ancêtres activée dans le graphe connu) et qui conditionne les influences futures du nœud. Cela permet de prendre en compte la topologie du réseau de diffusion dans le modèle, mais pas la trajectoire spécifique du contenu se diffusant (car moyennant tous les chemins possibles). Le modèle donné dans [Wang et al., 2017c] propose de dépasser cela en définissant un mécanisme d’attention sur le passé des épisodes de diffusion, pour assigner des poids aux infecteurs potentiels de chaque nouvel infecté observé, pris en compte pour la construction de son état latent. Le réseau d’attention est appris de manière à identifier de quel nœud provient toute nouvelle infection en fonction des états des nœuds infectés par le passé. Cependant, ce genre d’approche tend à converger vers des vecteurs d’attention au centre du simplexe, la diffusion correspondant à un processus hautement stochastique. Ce processus d’inférence déterministe limite la capacité du modèle de [Wang et al., 2017c] à prédire les infections futures, en produisant l’état de chaque nouvel infecté selon un mix des états de ses multiples infecteurs potentiels, plutôt qu’en échantillonnant les trajectoires passées selon leurs probabilités postérieures. Une approche similaire, donnée dans [Wang et al., 2018], n’utilise pas de RNN mais définit un module de composition d’états latents en fonction du passé via un modèle d’attention, à la manière des célèbres réseaux transformeurs [Vaswani et al., 2017]. À l’intersection de ces approches, [Islam et al., 2018] définit un processus temporel ponctuel conditionné, pour chaque nouvelle infection, par une représentation de la cascade à l’aide d’une architecture de type réseau *Transformer*. On note enfin l’approche intéressante de [Yang et al., 2019], basée sur de l’apprentissage par renforcement, qui contrôle les prédictions des processus temporels ponctuels à base de RNNs, via une supervision sur la taille des cascades prédites.

Récemment, divers travaux ont proposé d’employer des processus de marche aléatoire sur les

graphes, pour apprendre des représentations respectant les contraintes topologiques des réseaux à partir de trajectoires échantillonnées. Alors que DeepWalk [Perozzi et al., 2014] utilise uniquement des informations structurelles du réseau, les modèles proposés dans [Nguyen et al., 2018] et [Shi et al., 2018] incluent des contraintes temporelles dans les marches aléatoires, afin d'échantillonner des trajectoires réalistes au regard de séquences d'évènements observées. L'approche DeepCas [Li et al., 2016] applique cette idée à la prédiction de volume futur pour des cascades de diffusion, en échantillonnant, pour tout épisode observé, des séquences d'évènements de transmission qui ont pu mener à cet épisode, selon le graphe de diffusion considéré, afin d'extraire une représentation de la cascade associée via des méthodes de représentation de séquences (RNN) et un mécanisme d'attention pour agréger les différentes représentations. Cependant, ces approches requièrent de disposer d'un graphe de relations en entrée. De plus, le processus d'échantillonnage de DeepCas utilise de simples heuristiques pour parcourir le graphe, et ne correspond pas à un processus d'inférence probabiliste qui s'appuierait sur des estimations de distributions postérieures selon les épisodes de diffusion observés.

### 3.3.2 Modèle de Cascade Récurrent

Dans [Lamprier, 2019], nous avons proposé le premier modèle bayésien topologique à base de RNN pour séquences avec dépendances arborescentes, que nous appliquons à la modélisation de cascades de diffusion. Plutôt que de s'appuyer un échantillonnage préliminaire de trajectoires comme dans DeepCas, l'idée est d'établir un mécanisme d'inférence probabiliste utilisable au cours de l'apprentissage du modèle, selon les estimations courantes des paramètres, pour définir des représentations des nœuds infectés utiles pour prédire le futur de la diffusion. Dans ce modèle, comme illustré à la figure 3.9, nous considérons que les probabilités de diffusion à partir d'un nœud infecté  $v$  dépendent d'un état latent qui lui est associé, qui contient des informations utiles issues du passé de la diffusion. Cet état dépend de l'état du nœud  $u$  qui a le premier réussi à transmettre le contenu à

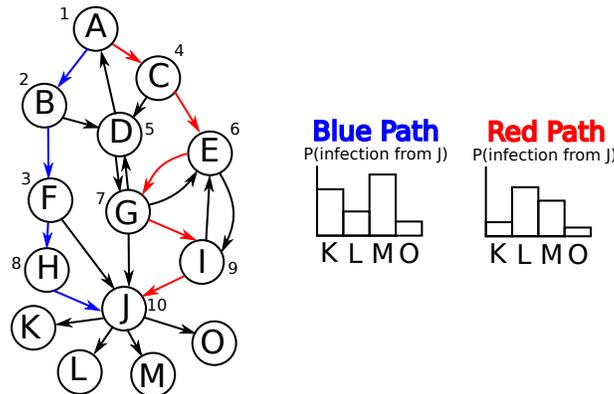


FIGURE 3.9 – Trajectoires de diffusion et impact sur les distributions d'infection. Les numéros à côtés des nœuds du graphe de diffusion correspondent à leur temps d'infection observé. Nous nous intéressons au cas du nœud J infecté au temps 10, dont l'infection peut découler de diverses trajectoires. On met en évidence deux de ces trajectoires. L'état de J dépend de la trajectoire considérée, et donc également les probabilités d'infection des successeurs de J qui ne sont pas les mêmes que le contenu ait emprunté le chemin bleu ou le chemin rouge.

$v$  (on pourrait envisager une version où toutes les tentatives réussies sont prises en compte dans la construction de l'état du nœud  $v$ , mais cela complexifierait considérablement le modèle). Cela implique de travailler avec un modèle en temps continu, qui permet d'éviter d'éventuelles transmissions

simultanées. Le modèle proposé s'appuie fortement sur le modèle CTIC [Saito et al., 2009] et son processus d'apprentissage, où l'on ajoute un processus d'inférence de trajectoire et l'utilisation de représentations continues de nœuds pour définir les probabilités d'infection. Notons qu'une version avec apprentissage de représentation a déjà été proposée avec succès pour le modèle CTIC dans [Zhang et al., 2018], cependant restant dans un cadre markovien classique, environ deux ans après notre extension d'IC aux espaces de représentation continue dans [Bourigault et al., 2016d]. Il s'agit alors d'y intégrer un mécanisme de prise en compte des trajectoires passées vraisemblables.

### 3.3.2.1 Modèle Génératif Récurrent pour Diffusion Continue

Dans CTIC, deux paramètres sont définis par relation ( $u \rightarrow v$ ) entre nœuds du réseau :  $k_{u,v} \in [0; 1]$ , qui correspond à la probabilité que le nœud  $u$  réussisse sa tentative d'infection de  $v$ , et  $r_{u,v} > 0$ , qui est un paramètre de délai utilisé dans une loi exponentielle. Si  $u$  réussit à infecter  $v$  dans un épisode  $D$ ,  $v$  est alors infecté au temps  $t_v^D = t_u^D + \delta$  (sauf si un autre nœud l'infecte plus tôt), avec  $\delta \sim \text{Exp}(r_{u,v})$ .

Notre modèle dans [Lamprier, 2019] s'appuie sur cette base, mais plutôt que de considérer une paire de paramètres par relation (ce qui implique  $2 \times |\mathcal{U}| \times (|\mathcal{U}| - 1)$  paramètres à stocker), on propose de considérer des fonctions neuronales produisant ces paramètres selon un état continu  $z_u^D \in \mathbb{R}^d$  de l'émetteur  $u$ , qui dépend donc du chemin pris par le contenu depuis la source de  $D$  pour atteindre  $u$ , ainsi que d'un état continu du récepteur  $v$ . Pour tout nœud  $v$  infecté dans  $D$ , connaissant l'état  $z_u^D$  du nœud  $u$  qui a infecté  $v$  en premier, l'état  $z_v^D$  est construit selon :

$$z_v^D = f_\phi(z_u^D, \omega_v^{(f)}) \quad (3.14)$$

avec  $f_\phi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$  une fonction, de paramètres  $\phi$ , qui transforme l'état de  $u$  selon une représentation continue statique  $\omega_v^{(f)} \in \mathbb{R}^d$  du nœud  $v$ . Cette fonction peut par exemple correspondre à une cellule RNN de Elman classique, un réseau de neurones multi-couches, une fonction résiduelle neuronale, ou encore une *Gated Recurrent Unit* (GRU). Dans nos expérimentations, on utilise un module GRU.

Pour un épisode  $D$ , la probabilité que  $u$  réussisse à transmettre le contenu se diffusant à un nœud  $v$  est donnée par :

$$k_{u,v}^D = \sigma \left( \langle z_u^D, \omega_v^{(k)} \rangle \right) \quad (3.15)$$

où  $\sigma(\cdot)$  correspond à la fonction sigmoïde et  $\omega_v^{(k)} \in \mathbb{R}^d$  est la représentation statique continue de  $v$  en tant que récepteur de contenu. De manière similaire à CTIC, si le nœud  $u$  réussit à infecter  $v$ , le délai d'infection dépend d'une loi exponentielle de paramètre  $r_{u,v}$ . Afin de simplifier l'apprentissage, on considère que le paramètre de délai ne dépend pas de l'historique de diffusion, seule la probabilité  $k_{u,v}$  en dépend. Ainsi, pour toute paire  $(u, v)$ , le paramètre de délai est le même pour tout épisode  $D$  :

$$r_{u,v} = \exp(-|\langle \omega_u^{(r,1)}, \omega_v^{(r,2)} \rangle|) \quad (3.16)$$

avec  $\omega_u^{(r,1)}$  et  $\omega_u^{(r,2)}$  correspondant à des représentations de délai de dimension  $d$  pour tout nœud  $u \in \mathcal{U}$ , respectivement pour  $u$  en tant qu'émetteur et en tant que récepteur de contenu.

Dans ce modèle, on considère que tous les épisodes sont initiés par un nœud additionnel fictif  $u_0$ , simulant le monde extérieur, infecté au temps  $t = 0$  de chaque épisode. Cela permet de faire démarrer tous les épisodes de la même manière, avec donc un état latent commun  $z_{u_0}^D = z_0$  pour tout  $D$ . Le processus génératif complet, similaire à celui de CTIC, est donné dans [Lamprier, 2019]. Pour chaque nouvel épisode, tant qu'il reste des nœuds dans un ensemble de nœuds infectieux (initialisé à

$\{u_0\}$ ), le processus sélectionne à chaque itération le nœud infectieux  $u$  de temps d'infection minimal, le retire de l'ensemble des nœuds infectieux, l'enregistre en tant qu'infecté et réalise des épreuves de Bernoulli pour savoir quels nœuds non-encore infectés  $v$  le nœud  $u$  réussit à infecter, selon les probabilités  $k_{u,v}^D$ . Pour les tentatives réussies, un temps  $t$  est échantillonné pour chaque nœud correspondant  $v$  selon donc une loi exponentielle de paramètre  $r_{u,v}$ . Si le nouveau temps  $t$  pour  $v$  est inférieur à son temps courant  $t_v^D$  (initialisé à  $t_v^D = \infty$  mais ayant pu avoir été échantillonné selon une infection précédente), ce nouveau temps est enregistré en tant que nouveau  $t_v^D$ ,  $v$  est ajouté à l'ensemble des nœuds infectieux (si pas déjà inclus) et son nouvel état  $z_v^D$  est alors calculé selon ce nouvel infecteur  $u$ .

### 3.3.2.2 Apprentissage par Inférence de Trajectoires

Soit  $\Theta = (\phi, z_0, (\omega_u^{(f)})_{u \in \mathcal{U}}, (\omega_u^{(r,1)})_{u \in \mathcal{U}}, (\omega_u^{(r,2)})_{u \in \mathcal{U}}, (\omega_u^{(k)})_{u \in \mathcal{U}})$  l'ensemble des paramètres du modèle. Ces paramètres sont appris en maximisant la vraisemblance d'un ensemble d'épisodes d'entraînement  $\mathcal{D}$  :

$$p(\mathcal{D}) = \prod_{D \in \mathcal{D}} p(D) = \prod_{D \in \mathcal{D}} \prod_{v \in U^D} h_v^D \prod_{v \notin U^D} g_v^D \quad (3.17)$$

où  $h_v^D$  correspond à la probabilité que  $v$  soit infecté au temps  $t_v^D$  selon les nœuds précédemment infectés dans  $D$  et  $g_v^D$  est la probabilité que  $v$  ne soit infecté par aucun infecté de  $D$ .

Comme dans CTIC [Saito et al., 2009], cette formulation requiert de considérer la probabilité qu'un nœud  $u \in U^D$  infecte un autre nœud  $v \in U^D$  au temps d'infection  $t_v^D > t_u^D$  dans  $D$ , donnée par :

$$a_{u,v}^D = k_{u,v}^D r_{u,v} \exp^{-r_{u,v}(t_v^D - t_u^D)} \quad (3.18)$$

Aussi, la probabilité que  $u$  n'infecte pas  $v$  avant  $t_v^D$  selon son état  $z_u^D$  est donnée par :

$$\begin{aligned} b_{u,v}^D &= 1 - k_{u,v}^D \int_{t_u^D}^{t_v^D} r_{u,v} \exp^{-r_{u,v}(t - t_u^D)} dt \\ &= k_{u,v}^D \exp^{-r_{u,v}(t_v^D - t_u^D)} + 1 - k_{u,v}^D \end{aligned} \quad (3.19)$$

La probabilité que le nœud  $v$  soit infecté au temps  $t_v^D$  connaissant les états  $z_u^D$  de tout nœud  $u$  infecté avant  $v$  dans  $D$  est alors donnée par :

$$h_v^D = \sum_{\substack{u \in \mathcal{U}, \\ t_u^D < t_v^D}} a_{u,v}^D \prod_{\substack{x \in \mathcal{U} \setminus \{u\}, \\ t_x^D < t_v^D}} b_{x,v}^D = \prod_{\substack{x \in \mathcal{U}, \\ t_x^D < t_v^D}} b_{x,v}^D \sum_{\substack{u \in \mathcal{U}, \\ t_u^D < t_v^D}} \frac{a_{u,v}^D}{b_{u,v}^D} \quad (3.20)$$

où, étant donné que l'on travaille avec des temps continus, on ignore la possibilité des infections simultanées. De manière similaire, la probabilité qu'un nœud  $v$  ne soit pas infecté dans  $D$  à la fin de la période d'observation de durée  $T$ , sachant que l'on connaît l'ensemble des états de tous les nœuds de  $D$ , est donnée par :

$$g_v^D = \prod_{u \in U^D} (k_{u,v}^D \exp^{-r_{u,v}(T - t_u^D)} + 1 - k_{u,v}^D) \approx \prod_{u \in U^D} (1 - k_{u,v}^D) \quad (3.21)$$

où l'approximation est obtenue en supposant une période d'observation suffisamment longue.

Le processus d'apprentissage de notre modèle est donc basé sur la maximisation de la vraisemblance donnée en Eq. 3.17 comme pour le modèle CTIC classique. Cependant, dans notre cas, les

probabilités d'infection  $k_{u,v}^D$  pour tous les couples  $(u, v)$  considérés dans les equations 3.20 et 3.21 requièrent la connaissance des états  $z_u^D$  pour tout  $D \in \mathcal{D}$ . Ces états dépendent chacun de la chaîne d'infecteurs par lesquels est passé le contenu se diffusant pour atteindre les nœuds correspondants. Malheureusement ces informations ne sont pas disponibles dans les données d'entraînement, il s'agit de marginaliser selon toutes les séquences d'infecteurs possibles  $I$  pour tout  $D \in \mathcal{D}$  :

$$\log p(\mathcal{D}) = \sum_{D \in \mathcal{D}} \log p(D) = \sum_{D \in \mathcal{D}} \log \sum_{I \in \mathcal{I}^D} p(D, I) \quad (3.22)$$

où  $\mathcal{I}^D$  est l'ensemble des possibles séquences d'infecteurs pour  $D$  et  $p(D, I)$  correspond à la probabilité jointe de l'épisode  $D$  et de la séquence d'infecteurs  $I \in \mathcal{I}^D$ . Puisqu'on écarte les possibilités d'infections simultanées, on peut travailler pour chaque épisode  $D$  avec des séquences d'infectés  $U^D = (U_0^D, \dots, U_{|D|-1}^D)$ , afin de simplifier l'écriture des différentes quantités manipulées. Dans la suite, on note alors  $U_i^D$ , pour tout  $i \in \{0, \dots, |D| - 1\}$ , le  $i$ -ième nœud infecté de l'épisode  $D$ . Cela permet notamment d'y faire correspondre des séquences d'infecteurs cachés  $I^D = (I_i^D)_{i \in \{0, \dots, |D|-1\}}$ , où  $I_i^D$  correspond à l'index dans  $U^D$  du nœud ayant en premier infecté le nœud  $U_i^D$  (fixé arbitrairement à 0 pour  $U_0^D$  qui correspond au nœud "monde extérieur"  $u_0$ , spontanément infecté en début de tout épisode). On a alors  $\mathcal{I}^D = \{(I_i^D \in \mathbb{N})_{i \in \{0, \dots, |D|-1\}} \mid I_0^D = 0 \wedge \forall i \in \{0, \dots, |D|-1\}, I_i^D < i\}$ . Dans la suite on note également  $D_i = (U_i^D, t_{U_i^D}^D)$  le  $i$ -ème infecté de  $D$  associé à son temps d'infection.

La décomposition  $p(D, I) = p(I)p(D|I)$  impliquerait le calcul difficile de  $p(D|I)$  selon notre modèle de cascade récurrent, qui requerrait l'estimation de la probabilité conditionnelle des infections (ou non-infections) de tout nœud  $u \in \mathcal{U}$  dans  $D$  selon la séquence complète d'infecteurs, antérieurs ou ultérieurs à  $t_u^D$ . Préféablement, en utilisant la *bayesian chain rule*, la probabilité jointe peut être factorisée selon (dérivation donnée dans [Lamprier, 2019]) :

$$p(D, I) = \prod_{i=1}^{|D|-1} p(D_i | D_{<i}, I_{<i}) p(I_i | D_{\leq i}, I_{<i}) \times \prod_{v \notin U^D} p(v \notin U^D | D_{\leq |D|-1}, I) \quad (3.23)$$

où  $D_{<i} = (D_j)_{j \in \{0, \dots, i-1\}}$  correspond à la séquence des  $i$  premières infections de  $D$  et  $I_{<i} = (I_j)_{j \in \{0, \dots, i-1\}}$  correspond au vecteur des  $i$  premières composantes de  $I$ . On a pour tout  $i \in \{1, \dots, |D| - 1\}$  :  $p(D_i | D_{<i}, I_{<i}) = h_{U_i^D}^D$ , tous les états utiles  $z_u^D$ , pour tout  $u \in U_{<i}^D$ , pouvant être directement déduits de  $D_{<i}$  et  $I_{<i}$  selon l'équation récurrente 3.14. De la même manière, on a :  $p(v \notin U^D | D_{\leq |D|-1}, I) = g_v^D$ . D'autre part  $p(I_i | D_{\leq i}, I_{<i})$  correspond à la probabilité conditionnelle que le nœud  $U_i^D$  ait été le premier nœud à infecter  $U_i^D$ , connaissant tous les événements d'infection antérieurs ainsi que le temps d'infection de  $U_i^D$ . Cela peut être obtenu, selon l'équation 3.20, via :

$$p(I_i | D_{\leq i}, I_{<i}) = \frac{a_{U_i^D, U_i^D}^D / b_{U_i^D, U_i^D}^D}{\sum_{u \in \mathcal{U}, t_u^D < t_{U_i^D}^D} a_{u, U_i^D}^D / b_{u, U_i^D}^D} \quad (3.24)$$

Malheureusement la log-vraisemblance donnée en (3.22) reste particulièrement difficile à optimiser directement car cela requiert de considérer tous les vecteurs possibles  $I \in \mathcal{I}^D$  pour chaque épisode  $D$  d'entraînement à chaque itération de l'optimisation. De plus, le produit de probabilités en (3.23) conduirait à des gradients nuls en raison des limites de représentation en virgule flottante. En conséquence, il est nécessaire de définir un mécanisme d'inférence de trajectoires, à partir duquel il sera possible d'échantillonner au cours de l'apprentissage. Différent choix peuvent être considérés. Tout

d'abord, des méthodes MCMC (e.g., échantillonneur de Gibbs) pourraient être envisagées à chaque itération d'un GEM, mais cela requerrait d'échantillonner à partir de postérieures de trajectoires complètes des cascades, ce qui s'avérerait particulièrement complexe et instable. Ces calculs coûteux pour l'échantillonnage des distributions postérieures pourraient être évités en utilisant des distributions propositionnelles plus simples, comme réalisé par exemple via échantillonnage préférentiel avec variables auxiliaires dans [Farajtabar et al., 2015] pour des problèmes de détection de sources de diffusion. Cependant, cela se heurterait très probablement à une variance très forte dans notre cas. Une autre possibilité est d'utiliser à nouveau l'inférence variationnelle [Kingma and Welling, 2013], où l'on cherche à faire tendre une distribution auxiliaire  $q$  vers la distribution postérieure  $P(I|D)$ , de laquelle il est plus facile d'échantillonner. Diverses stratégies d'inférence par lissage (*smoothing*), où l'on considère la totalité de l'épisode pour l'inférence, seraient envisageables pour la distribution  $q$  (voir perspectives, section 3.4.2.1). Dans [Lamprier, 2019], nous avons plutôt proposé de simplifier le problème en se basant sur une stratégie d'inférence par filtrage, qui s'appuie donc sur des distributions conditionnelles selon le passé uniquement. Selon notre modèle on prend alors  $q_i^D(I_i) = p(I_i|D_{\leq i}, I_{< i})$ , ce qui peut être calculé efficacement et permet de nous passer de tout paramètre variationnel additionnel.

De l'inégalité de Jensen sur les fonctions concaves, on obtient pour tout épisode  $D$  :

$$\begin{aligned} \log p(D) = \log \sum_{I \in \mathcal{I}^D} p(D, I) &\geq \mathbb{E}_{I \sim q^D} [\log p(D, I) - \log q^D(I)] \\ &= \mathbb{E}_{I \sim q^D} \left[ \sum_{i=1}^{|\mathcal{D}|-1} \log p(D_i | D_{< i}, I_{< i}) + \sum_{v \notin U^D} \log p(v \notin U^D | D_{\leq |\mathcal{D}|-1}, I) \right] \\ &\triangleq \mathcal{L}(D; \Theta) \end{aligned} \quad (3.25)$$

où  $q^D(I) = \prod_{i=1}^{|\mathcal{D}|-1} p(I_i | D_{\leq i}, I_{< i})$ .  $\mathcal{L}(D; \Theta)$  correspond alors à une borne inférieure de la log-vraisemblance (ELBO), définie comme une espérance à partir de laquelle il est possible d'échantillonner, infecteur après infecteur de manière chronologique, en ne considérant que le passé de la diffusion à chaque étape. Maximiser cette borne inférieure encourage le processus à choisir des trajectoires qui expliquent au mieux les épisodes observés. Pour la maximiser par optimisation stochastique, nous employons l'estimateur de fonction de score (*score function estimator*, *log-trick* ou encore *Reinforce-trick* [Ranganath et al., 2014]), qui exploite la dérivée de la fonction  $\log(\nabla_{\theta} \log p(\mathbf{x}; \theta) = \frac{\nabla_{\theta} p(\mathbf{x}; \theta)}{p(\mathbf{x}; \theta)})$  pour exprimer le gradient de l'espérance en une espérance de gradients, à partir de laquelle on peut échantillonner pour l'estimation non biaisée des gradients. Une autre possibilité aurait été de se baser sur un *Gumbel-Softmax* et la distribution *Concrete* associée, via l'astuce de re-paramétrisation (*reparameterization trick*) comme proposé dans [Maddison et al., 2016], mais nous avons observé de meilleurs résultats de convergence avec l'estimateur de fonction de score. Le gradient de notre ELBO pour tous les épisodes de  $\mathcal{D}$  est alors donné par :

$$\nabla_{\Theta} \mathcal{L}(\mathcal{D}; \Theta) = \sum_{D \in \mathcal{D}} \mathbb{E}_{I \sim q^D} [(\log p(D, I) - \log q^D(I) - 1 - b) \nabla_{\Theta} \log q^D(I) + \nabla_{\Theta} \log p(D, I)] \quad (3.26)$$

$$= \sum_{D \in \mathcal{D}} \mathbb{E}_{I \sim q^D} [(\log p^I(D) - b) \nabla_{\Theta} \log q^D(I) + \nabla_{\Theta} \log p^I(D)] \quad (3.27)$$

où  $p^I(D)$  est un raccourci pour  $\prod_{i=1}^{|\mathcal{D}|-1} p(D_i | D_{< i}, I_{< i}) \prod_{v \notin U^D} p(v \notin U^D | D_{\leq |\mathcal{D}|-1}, I)$  et  $b$  est une *baseline* moyenne mobile de l'ELBO par épisode, utilisée pour réduire la variance de l'estimateur. Cette formulation permet d'obtenir un estimateur de gradient non biaisé en remplaçant les espérances par des moyennes sur  $K$  échantillons pour chaque épisode à chaque itération ( $K = 1$  dans nos expérimentations). La simplification donnée en 3.27 est obtenue en faisant remarquer que  $p^I(D) = p(D, I) / q_i^D(I_i)$ .

NLL	$\tau_0$	$\tau_1$	$\tau_2$	$\tau_3$	CE	$\tau_0$	$\tau_1$	$\tau_2$	$\tau_3$	INF	$\tau_0$	$\tau_1$	$\tau_2$	$\tau_3$
RMTTPP	19,7	14,5	11,8	7,65	RMTTPP	79,0	21,0	14,2	9,3	RMTTPP	-	-	-	-
CYAN	18,7	13,6	11,0	7,09	CYAN	86,9	19,6	14,4	9,4	CYAN	0,30	0,15	0,11	0,11
CYAN-cov	19,5	14,3	11,6	7,41	CYAN-cov	91,7	27,8	19,7	12,4	CYAN-cov	0,29	0,14	0,10	0,10
DAN	18,7	13,5	10,9	6,84	DAN	97,9	98,6	75,7	43,8	DAN	0,42	0,31	0,23	0,20
CTIC	20,1	14,3	10,2	5,99	CTIC	<b>63,1</b>	23,2	24,7	21,8	ctic	0,73	0,67	0,61	0,58
embCTIC	19,9	14,1	10,1	5,97	embCTIC	65,8	22,6	17,2	12,1	embCTIC	0,73	0,67	0,61	0,58
recCTIC	<b>17,4</b>	<b>11,6</b>	<b>8,3</b>	<b>4,97</b>	recCTIC	68,7	<b>15,9</b>	<b>11,2</b>	<b>8,3</b>	recCTIC	<b>0,90</b>	<b>0,88</b>	<b>0,86</b>	<b>0,84</b>

TABEAU 3.2 – Résultats de recCTIC (comparés à divers modèles de cascade et modèles récurrents), en termes de *Negative Log-Likelihood* (NLL), *Cross-Entropy* des infectés (CE) et prediction des *Infecteurs* (INF).

Elle permet d'exhiber deux termes : alors que le premier terme encourage de fortes probabilités conditionnelles pour les ancêtres maximisant la vraisemblance des épisodes complets, le second terme tend à l'amélioration de la vraisemblance des infections observées en fonction du passé de la trajectoire échantillonnées. Néanmoins, il est plus efficace d'utiliser la formulation 3.26, sachant que tout calcul  $p(D_i|D_{<i}, I_{<i})$  requiert la marginalisation sur tous les infecteurs  $I_i$  possibles (voir Eq. 3.20), alors que le calcul de  $\log p(D, I)$  n'implique qu'une somme de log-probabilités où chaque composante  $\log p(D_i, I_i|D_{<i}, I_{<i}) = \log a_{U_i^D, U_i^D} + \sum_{u \in \mathcal{U} \setminus \{U_i^D\}, t_u^D < t_{U_i^D}^D} \log b_{u, U_i^D}^D$ . Le pseudo-algorithme d'apprentissage complet, travaillant par minibatches d'épisodes, ordonnés par longueur pour limiter le *padding*, et exploitant au maximum les capacités de parallélisation par programmation matricielle pour déploiement sur GPU (en *pytorch* dans nos expérimentations) est donné en appendice de [Lamprier, 2019]. L'optimisation est réalisée à chaque étape via le célèbre optimiseur ADAM [Kingma and Ba, 2014].

**Résultats** La table 3.2 reporte un échantillon de résultats expérimentaux obtenus pour notre méthode de diffusion récurrente. Ils concernent des données artificielles obtenues sur un réseau aléatoire *scale-free* de 100 noeuds généré selon l'algorithme *Barabási–Albert* [Barabási and Albert, 1999]. Le processus de génération des cascades sur ce graphe suit le modèle CTIC, mais où, pour chaque cascade simulée, les probabilités de transmission  $k_{u,v}^D = \sigma(\langle \omega^D, m_{u,v} \otimes x_{u,v} \rangle)$  dépendent d'une fonction du contenu transmis  $\omega^D \in [0; 1]^l$  dans cette diffusion D, d'un masque aléatoire binaire  $m_{u,v} \in \{0; 1\}^l$  pour le lien  $(u, v)$  et d'un vecteur de caractéristiques  $x_{u,v} \in [0; 1]^l$  défini pour le receveur  $v$  (avec  $l = 5$  dans nos expérimentations). Les caractéristiques des noeuds hub (noeuds de degré supérieur à 30) sont échantillonnées selon une distribution de Dirichlet de paramètre  $\alpha = 10$  (noeuds multi-contenus), alors que celles des autres noeuds sont échantillonnées d'une Dirichlet avec  $\alpha = 0.1$  (noeuds spécialisés sur un ou quelques contenus spécifiques). Avant chaque simulation D, le contenu  $\omega^D$  est échantillonné d'une Dirichlet de paramètre  $\alpha = 0.1$ . Ce contenu reste caché de l'agent apprenant, mais permet de définir des épisodes dans lesquels les distributions d'infection dépendent de l'historique de diffusion. Les paramètres de délai  $r_{u,v}$  sont tous tirés uniformément dans  $[0, 1]$ . 10000 épisodes sont générés de cette manière, dont 5000 pour l'apprentissage des modèles.

En plus des résultats pour notre modèle récurrent recCTIC, le tableau reporte des résultats pour le modèle CTIC classique ainsi qu'une version embCTIC de notre modèle où  $z_u^D$  est remplacé par une représentation continue statique apprise pour chaque  $u$ . Ces modèles de cascade sont comparés à divers modèles neuronaux récurrents discutés en section 3.3.1, à savoir le processus temporel ponctuel RMTTPP de [Du et al., 2016], qui correspond à l'application d'un RNN classique pour prédire le prochain évènement selon la séquence historique de l'épisode, le modèle CYAN [Wang et al., 2017c], similaire à RMTTPP mais où un mécanisme d'attention vise à sélectionner de quel état précédent repartir pour chaque nouvel évènement et son extension CYAN-cov qui utilise un mécanisme d'attention

plus évolué visant à donner plus de poids aux noeuds importants. Enfin, DAN correspond au modèle de [Wang et al., 2018], proche de CYAN mais se passant de RNN, basé sur des modules de composition par attention uniquement.

Le tableau de gauche donne des résultats en modélisation de la diffusion, où l'on considère des résultats en terme de *Negative Log-Likelihood* :  $NLL = -(1/|\mathcal{D}_{test}|) \sum_{D \in \mathcal{D}_{test}} \log p(D)$ , avec  $p(D)$  la probabilité de l'épisode de test  $D$  selon le modèle considéré (estimée via échantillonnage préférentiel dans le cas de notre modèle). Le tableau du milieu donne des résultats en prédiction de diffusion, où on considère l'entropie croisée des infectés des épisodes de test :  $CE = \frac{1}{|\mathcal{D}_{test}| \times |\mathcal{U}|} \sum_{D \in \mathcal{D}_{test}} \sum_{u \in \mathcal{U}} \log p(u \in U^D)^{\mathbb{1}(u \in U^D)} + \log p(u \notin U^D)^{\mathbb{1}(u \notin U^D)}$ , où  $\mathbb{1}(\cdot)$  correspond à la fonction indicatrice et  $p(u \in U^D)$  est estimée via des simulations de Monte-Carlo selon le processus génératif des différents modèles. Enfin, le tableau de droite reporte des résultats en prédiction de trajectoires, où  $INF = \frac{1}{\sum_{D \in \mathcal{D}_{test}} (|D|-1)} \sum_{D \in \mathcal{D}_{test}} \sum_{i \in \{1, \dots, |D|-1\}} p(I_i = I_i^D | D_{\leq i})$  correspond à un ratio de choix des vrais infecteurs  $I_i^D$  de la cascade (uniquement connus pour les cascades artificielles), selon les poids d'attention pour les modèles CYAN, CYAN-cov et DAN, et selon les probabilités de sélection des infecteurs (Eq. 3.24) pour les autres modèles (via échantillonnage préférentiel sur les infecteurs précédents pour recCTIC). RMTTP est exclu de ce tableau car il ne définit aucun mécanisme d'inférence des infecteurs. Pour ces trois types de mesure, on reporte des résultats avec différents volumes d'observations initiales des épisodes de test : les infections survenues avant un certain délai  $\tau$  à partir du début de l'épisode sont données en entrée des modèles, à partir desquelles ils peuvent inférer des représentations internes, les mesures d'évaluation sont calculées sur la suite des épisodes. Dans les tableaux,  $\tau_0$  indique que rien n'est observé (prédiction de la cascade moyenne),  $\tau_1$  indique que seules les infections au premier pas de temps  $t = 1$  sont connues des modèles (conditionnement sur les sources seulement),  $\tau_2$  et  $\tau_3$  correspondent à des conditionnements sur les infections survenues avant un délai de  $\tau = \max T / 20$  et  $\tau = \max T / 10$  respectivement. De nombreux détails sur le conditionnement des modèles, particulièrement recCTIC pour lequel ce conditionnement n'est pas trivial, sont donnés en annexe de [Lamprier, 2019].

Sur les trois tâches et quel que soit le niveau de conditionnement, la méthode recCTIC démontre des performances supérieures à celles des modèles comparés, excepté en terme de CE avec  $\tau_0$ , le modèle souffrant d'une plus grande variance en simulation lorsqu'aucun conditionnement ne lui est donné. On note par ailleurs que plus le conditionnement est important, plus le modèle est performant par rapport aux autres modèles, le modèle est capable de tirer efficacement partie des observations de début d'épisode pour inférer des trajectoires utiles sur la suite. Enfin, on note les très bonnes performances en inférence des infecteurs sur ce jeu de données, le modèle attribuant un bien meilleur crédit aux infecteurs réels, et laissant donc supposer des trajectoires représentatives des vrais chemins de diffusion dans le réseau considéré. Bien sûr il ne s'agit là que de données artificielles, bien adaptées au modèle car formées selon le modèle CTIC avec probabilités de transmission dépendantes du contenu diffusé caché et avec présence de noeuds hubs, qui était une configuration représentative des limites des modèles de cascade existants, discutée à l'occasion de la figure 3.7. Néanmoins, ces résultats montrent les bonnes capacités du modèle lorsque les données suivent relativement bien ses hypothèses, et en particulier montrent qu'il est capable de dépasser significativement les capacités des modèles récurrents existants grâce à son mécanisme d'inférence probabiliste. Les résultats donnés dans [Lamprier, 2019] sur des données réelles confirment cette tendance.

**Publication associée :**

- Lamprier, S. (2019). A recurrent neural cascade-based model for continuous-time diffusion. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 3632–3641

## 3.4 Conclusions et Perspectives

Dans ce chapitre nous avons abordé divers aspects de modélisation pour la diffusion dans les réseaux, particulièrement en l'absence de graphe de relations explicites sur lequel baser les approches et lorsque les observations se limitent à des événements de participation à une diffusion (e.g., partage de contenu, adoption de produit, de comportement, etc.). Nous nous sommes concentrés principalement sur les modèles de cascade qui paraissent les mieux adaptés pour la modélisation des dynamiques de l'information dans les réseaux sociaux. Les régularités temporelles dans ces réseaux sont difficiles à observer et l'apprentissage de ces modèles peut s'avérer délicat dans des contextes réels (modélisation à large échelle, données fortement bruitées, facteurs d'influence multiples, distributions non-stationnaires, etc.). De ce fait, un certain nombre d'approches pour des tâches de prédiction ont choisi de se tourner vers des méthodes plus directes de mise en correspondance d'entrées (e.g., un vecteur binaire des sources observées) avec des sorties attendues (e.g., un vecteur d'infections finales). C'est le cas par exemple d'approches comme celles de [Najar et al., 2012] ou [Du et al., 2014], qui apprennent des fonctions de prédiction des infectés finaux à partir des sources, de notre contribution pour la détection de sources discutée en fin de section 3.2.2 [Bourigault et al., 2016c], de l'approche de [Yang and Leskovec, 2010] qui considère des fonctions d'influence globale pour les infectés plutôt que de se concentrer sur des relations entre paires de nœuds, ou encore de celle de [Tsur and Rappoport, 2012] qui prédit la diffusion de hashtags par régression logistique selon des entrées variées sur les activités des utilisateurs ou le contenu se diffusant. Bien que des approches comme celle de [Narasimhan et al., 2015] appuient leurs fonctions de prédiction d'infections sur des modèles itératifs type LT ou IC, avec garanties d'apprentissage PAC (*probably approximately correct*) sur les infections finales, de nombreuses approches se passent complètement de ces modèles explicatifs, souvent sujets au sur-apprentissage et peu robustes aux bruits dans les données d'entraînement. Nos contributions dans ce chapitre avaient pour objet de reconsidérer les méthodes explicatives sur des graphes sous-jacents, en y intégrant des mécanismes de régularisation et de prise en compte de dépendances complexes cachées, leur permettant d'atteindre des performances compétitives sur des tâches de prédiction, tout en permettant l'extraction de dynamiques dans les réseaux considérés.

Ce travail ouvre à diverses pistes de recherche que nous envisageons de poursuivre, afin d'accroître les capacités de modélisation des dynamiques de diffusion et/ou d'étendre le champ d'application des méthodes proposées. La section 3.4.1 commence par formuler les modèles de cascade sous la forme de réseaux bayésiens dynamiques, avant de discuter de trois perspectives dans ce cadre, autour de problématiques de causalité dans les réseaux, d'apprentissage avec données incomplètes et d'extraction d'épisodes d'apprentissage à partir de données sociales brutes. La section 3.4.2 donne différentes extensions possibles par apprentissage profond et apprentissage par renforcement des modèles proposés dans ce chapitre. Enfin, la section 3.4.3 discute de l'application en ligne des modèles de diffusion, pour l'extraction de relations d'influence à partir de flux continus de données sociales.

### 3.4.1 Modèles graphiques bayésiens

Contrairement aux réseaux de Markov, qui sont des graphes non dirigés sur lesquels on peut mesurer des fonctions de potentiel d'instanciations jointes de variables, les réseaux bayésiens permettent

de représenter des relations de cause à effet. Le modèle IC largement étudié dans ce chapitre correspond en fait à un modèle de réseau bayésien dynamique [Murphy and Russell, 2002]. Ce type de modèle s'intéresse à la probabilité jointe d'une séquences de variables  $V_{0:T}$ , décrivant les états successifs d'un ensemble d'éléments  $V$ , où  $V_t = \{V_t^v\}_{v \in V}$  correspond à l'ensemble des variables décrivant l'état des éléments de  $V$  au temps  $t$ . Les réseaux bayésiens dynamiques sont composés de deux graphes bayésiens  $(B_{prior}, B_{dyn})$ , où  $B_{prior}$  est un DAG décrivant des relations probabilistes de cause à effet avant le temps 0, qui définit alors les distributions initiales sur les états des différentes variables  $V_0$  pour tous les éléments de  $V$ , et  $B_{dyn}$  décrit les relations entre variables de pas de temps successifs, de  $V_t$  vers  $V_{t+1}$  pour tout  $t \in \{0, \dots, T-1\}$ . La probabilité jointe dans ce type de réseau est donnée par :

$$P(V_{0:T}) = P(V_0) \times \prod_{t=1}^T P(V_t | V_{t-1}) = \prod_{v \in V} P(V_0^v | V_0^{pa_{prior}(v)}) \times \prod_{t=1}^T \prod_{v \in V} P(V_t^v | V_{t-1}^{pa_{dyn}(v)})$$

où  $pa_{prior}(v)$  et  $pa_{dyn}(v)$  correspondent respectivement aux parents directs de  $v \in V$  dans le réseau  $B_{prior}$  et le réseau  $B_{dyn}$ . Ce type de modèle est une généralisation de modèles probabilistes de séries temporelles comme les modèles de Markov ou les filtres de Kalman.

Dans le cas du modèle IC, on travaille avec des variables binaires, où  $V_t^v = 1$  si  $v$  est infecté au temps  $t$ , 0 sinon.  $B_{prior}$  ne contient aucune relation de cause à effet, il contient uniquement les probabilités pour chaque élément d'être source de diffusion. Dans le cas de l'ajout d'un noeud monde  $u_0$  comme dans [Gruhl et al., 2004] ou comme on l'a fait pour recCTIC, ces probabilités initiales valent 0 pour tous les nœuds sauf  $u_0$  pour lequel elle est de 1.  $B_{dyn}$  contient le graphe de diffusion  $G$ , où tout  $\theta_{u,v}$  correspond à une relation de  $V_t^u$  à  $V_{t+1}^v$  pour tout  $t \in \{0, \dots, T-1\}$ . Néanmoins, il faut modéliser le fait qu'un nœud n'est infectieux que pour le pas de temps suivant. Pour cela on a besoin d'un ensemble de variables supplémentaires  $W_{0:T}$ , où  $W_t^v = 1$  si  $v$  a été infecté avant  $t$ , 0 sinon. Chacune de ces variables "mémoire"  $W_t^v$ , avec  $W_0^v = 0$  pour tout  $v$ , sont déclenchées de manière déterministe lorsque  $V_{t-1}^v = 1$ . On a  $P(W_t^v = 1 | V_{t-1}, W_{t-1}) = I(V_{t-1}^v = 1 \vee W_{t-1}^v = 1)$ . On a aussi  $P(V_t^v = 1 | V_{t-1}, W_{t-1}) = (1 - V_{t-1}^v)(1 - W_{t-1}^v)(1 - \prod_{u \in \Theta, V_{t-1}^u = 1} (1 - \theta_{u,v}))$ . Pour le cas continu, les modèles tels que NetRate et CTIC entrent dans le cadre d'une version à temps continu de ces réseaux bayésiens dynamiques [Nodelman et al., 2002, Nodelman et al., 2012].

Cette formulation des modèles de cascade en réseaux bayésiens dynamiques permet d'envisager l'apprentissage de dynamiques de diffusion sous un nouvel angle, ouvrant notamment sur la possibilité d'application de diverses méthodes d'optimisation proposées pour ce cadre [Murphy and Russell, 2002, Nodelman et al., 2012, Benhamou et al., 2018]. En outre, cela pose un cadre formel pour la reconsidération de diverses hypothèses usuelles, notamment les hypothèses de monde clos et d'observabilité complète des phénomènes de diffusion, explorées ci-dessous. Trois types de perspectives dans ce cadre sont discutées dans la suite de cette section.

### 3.4.1.1 Corrélacion Temporelle versus Cause-Conséquence

La formulation des modèles de cascade en réseaux bayésiens met en évidence le type de relation cause-conséquence qu'ils modélisent. Néanmoins, l'extraction de ce genre de relations n'est pas évidente en général, car elles se confondent largement avec des relations de corrélation temporelle. La distinction entre relations de causalité et relations de corrélation est au centre de nombreuses recherches, considérée par des grands noms de l'apprentissage statistique comme la clé des avancées futures en IA, menant au raisonnement. Un exemple, quelque peu anecdotique mais parlant, de l'importance de la distinction de ces deux notions est donné dans [Schölkopf, 2019] pour le cadre de la

recommandation d'articles : après l'achat d'une housse pour ordinateur portable par un consommateur sur un site en ligne, un système naïf ne distinguant pas causalité de corrélation peut en venir à lui proposer des ordinateurs portables alors qu'il est probable que ce consommateur en ait déjà un (sinon pourquoi acheter une housse?). Ce type d'ambiguïté peut être en partie levé par la prise en compte d'informations temporelles lorsque disponibles, comme dans [Anagnostopoulos et al., 2008] qui propose un test de permutation temporelle dans le cadre du modèle de diffusion LT. Les modèles de cascade, basés sur la temporalité des événements, n'ont par ailleurs pas ce genre de problème, ils modélisent ces relations naturellement. Néanmoins, ils se heurtent comme les autres modèles aux phénomènes de corrélation temporelle, dans lesquels un stimuli externe (variable confondante) impacte de la même manière les différents utilisateurs, mais avec des distributions de délais différentes sur chacun d'eux. Dans un ouvrage fondateur, [Reichenbach, 1991] a établi la connexion entre causalité et dépendance statistique par le principe de cause commune, qui suppose l'existence d'une variable confondante  $Z$  pour toute paire de variables  $(X, Y)$  statistiquement dépendantes, les rendant conditionnellement indépendantes (i.e.,  $X \perp Y|Z$ ). La causalité de  $X$  vers  $Y$  est alors un cas particulier de ce modèle général, qui survient lorsque  $Z$  se confond avec  $X$ . Or, il est établi que sans hypothèses supplémentaires, nous ne pouvons pas distinguer ces cas en utilisant les seules données observées, les distributions d'observation sur  $X$  et  $Y$  étant les mêmes quel que soit  $Z$ . Un modèle causal contient clairement plus d'informations qu'un modèle statistique.

Les modèles de cascade émettent généralement une hypothèse de monde clos pour l'apprentissage, revenant à réduire toute dépendance statistique observée au cas particulier de la causalité par influence sur le réseau, ce qui tend à sur-estimer largement ces relations d'influence entre utilisateurs [Aral et al., 2009]. L'idéal pour séparer les influences des causes externes serait d'être à même d'intervenir sur le réseau, par exemple en bloquant les partages de certains utilisateurs vers certains autres pendant une période de temps, et d'étudier les variations de distributions selon ces interventions. Les médias sociaux tels que *Facebook* ou *Twitter* peuvent le faire par exemple en agissant sur leurs stratégies de diffusion des partages. Cependant dans la plupart des cas, nous n'avons pas ce genre de possibilité d'intervention. Diverses approches autour de l'inférence causale ont été développées, pour traiter notamment des biais de sélection dans des études où l'on n'a pas le contrôle sur les sujets étudiés ou lorsque les interventions sont irréalisables. Parmi elles, les méthodes d'appariement introduites dans [Rosenbaum and Rubin, 1983] comparent les observations associées à des paires d'individus traités-non traités, qui devraient réagir de la même manière selon leurs caractéristiques si le traitement n'a pas d'effet. [Aral et al., 2009] l'applique à la diffusion sur les réseaux pour se rendre compte du rapport influence-homophilie dans l'explication des corrélations d'infections observées. Néanmoins cela reste cantonné à une approche macroscopique, pas directement applicable pour l'extraction du graphe d'influence du réseau. Une approche plus orientée sur l'inférence causale a été proposée par l'introduction par Pearl de l'opérateur d'intervention *do* et des règles *do-calculus* associées [Pearl, 2012]. L'idée est d'estimer la probabilité conditionnelle  $P(Y|do(X))$ , qui correspond à la probabilité d'observation de  $Y$  selon une intervention sur une variable liée  $X$ , que l'on distingue de  $P(Y|X)$ , correspondant à la probabilité classique de  $Y$  sachant l'observation de  $X$ . La différence réside dans le fait que l'intervention sur  $X$  isole l'influence d'une éventuelle variable confondante  $Z$ . Dans divers cas, même lorsque l'on ne dispose que de données d'observation originales, il est possible d'exprimer  $P(Y|do(X))$  selon uniquement des termes ne dépendant pas de l'opérateur *do*, ce qui amène à l'estimation possible du lien de causalité non biaisé à partir des données d'observations uniquement. Cela suppose néanmoins l'estimation de distributions selon la variable confondante, qui n'est pas toujours observable. Dans ce cas, des hypothèses additionnelles sont nécessaires pour espérer pouvoir distinguer corrélation de causalité [Schölkopf, 2019].

Dans le cadre des cascades de diffusion, sous hypothèses de prior pour une variable confondante  $Z$  représentant l'état du monde extérieur (en lien éventuel avec la nature du contenu se diffusant), il est possible de définir un mécanisme d'inférence de cette variable cachée  $Z$  à partir d'épisodes complets, à la manière de recCTIC et de l'approche par lissage envisagée en section 3.4.2.1, et de s'y appuyer pour isoler les causes d'infection. Cela pourrait être par ailleurs guidé dans un cadre semi-supervisé où certains événements extérieurs seraient observables ponctuellement. Une source d'inspiration intéressante est le travail décrit dans [Blondel et al., 2017], qui applique les notions de *do-calculus* aux réseaux bayésiens dynamiques pour former des réseaux causaux dynamiques. Une possibilité de modélisation serait également de s'appuyer sur les modèles causaux structurels (SCM), introduits dans [Pearl, 2009], qui expriment les relations causales sous la forme de fonctions déterministes  $X^i := f_i(X^{pa(i)}, U_i)$ , avec  $U_1, \dots, U_n$  un ensemble de variables de bruit conjointement indépendantes. Le bruit  $U_i$  permet d'assurer que la fonction  $f_i$  représente bien la probabilité conditionnelle  $P(X^i | X^{pa(i)})$ , tout en conférant plus de liberté au modèle qu'en travaillant directement avec ces probabilités, notamment pour l'expression des interventions dans le réseau. Cela permet en outre de répondre à des questions contrefactuelles, du type "à quoi ressemblerait tel épisode de diffusion dans lequel tel utilisateur  $v$  ne serait pas infecté?", par l'application récursive des fonctions  $f$  sur l'ensemble du réseau causal, permettant alors de propager aisément l'intervention contre-factuelle issue d'une modification à un point de celui-ci, et ainsi individualiser la réponse pour l'épisode considéré. Notons que cela rejoint une de nos contributions récentes en apprentissage statistique équitable (*fairness*) par production d'exemples contrefactuels [Grari et al., 2020a], suivant les travaux fondateurs de [Kusner et al., 2017] pour cette tâche.

### 3.4.1.2 Données incomplètes

Au delà des difficultés d'observation de variables confondantes discutées à la section précédente, peuvent se poser également des limites d'observation des infections de certains nœuds du réseau [Kossinets, 2006]. Or, la plupart des modèles de diffusion de la littérature considèrent que la totalité des infectés sont connus pour chaque épisode de diffusion d'entraînement : si un nœud n'est pas observé comme appartenant à un épisode de diffusion, il est considéré comme n'ayant pas été impacté par le contenu correspondant. Dans de nombreux contextes il n'est pas possible d'observer la totalité de l'activité du réseau et il se peut qu'un nœud, bien qu'ayant réagi à un contenu donné, n'ait pas été enregistré comme tel. Il ne faudrait alors pas le considérer comme un exemple négatif de diffusion lors de l'apprentissage des modèles. Le travail dans [Najar et al., 2012] cité ci-dessus part justement du constat que les modèles de cascade sont très sensibles aux données d'infection manquantes dans les épisodes d'entraînement, pour justifier le développement de méthodes prédictives plus directes, ne nécessitant pas l'explication des canaux de diffusion. De la même manière que pour l'apprentissage de modèles de filtrage collaboratif en recommandation, où il faut dépasser les méthodes du type décomposition en valeurs singulières (SVD) classiques pour éviter de considérer l'absence de notation sur des couples utilisateur-produit comme des exemples de mauvaise notation [Koren et al., 2009], les modèles d'apprentissage pour la diffusion devraient inclure des mécanismes de prise en compte des absences de données possibles dans les épisodes d'entraînement.

Quelques travaux ont d'ores et déjà considéré la prise en compte de données manquantes dans des modèles de cascade. Dans [Wu et al., 2013], les auteurs s'intéressent, dans un contexte de modélisation des dynamiques de métapopulation, à l'absence d'observations dans le cadre du modèle IC classique. Chaque mesure sur les populations est coûteuse et il est fréquent que de nombreuses données de participation aux cascades manquent. Pour chaque épisode  $D$ , seuls certains nœuds

$u \in \mathcal{O}^D \subseteq \mathcal{U}$  sont observés. Pour tous les autres noeuds dans  $\bar{\mathcal{O}}^D = \mathcal{U} \setminus \mathcal{O}^D$ , on ne connaît pas leur statut d'infection au cours de l'épisode  $D$ . Une application naïve de IC mènerait à considérer ces noeuds comme non infectés dans  $D$ , ce qui mène à des biais dans l'estimation des paramètres (sous-estimation des probabilités impliquant ces noeuds et sur-estimations de l'influence des autres noeuds du fait du report des explications pour les épisodes concernés). L'idée dans [Wu et al., 2013] est de s'appuyer sur un processus de Monte Carlo EM, dans lequel des épisodes complétés sont échantillonnés selon un processus de Gibbs à chaque étape de l'algorithme. Soit  $T_u^D \in \{1, \dots, T\} + \infty$  la variable aléatoire prenant la valeur du temps d'infection de  $u$  dans la diffusion  $D$ . Pour chaque épisode dans  $\mathcal{D}$  et chaque noeud  $u \in \mathcal{O}^D$ , on a  $T_u^D = t_u^D$ . Pour tous les autres, il s'agit d'échantillonner itérativement  $T_u^D$  selon  $P_{\Theta}(T_u^D = t | D_{-u}) \propto P_{\Theta}(D_{-u} \cup \{(u, t)\})$ , avec  $D_{-u}$  l'ensemble des couples  $\{(v, T_v^D)\}_{v \in \mathcal{U} \setminus \{u\}}$  pour la diffusion  $D$ . Cela peut se faire efficacement par une simple mise à jour des vraisemblances selon l'ancienne valeur  $T_u^D$  : probabilité pour  $u$  d'être infecté au temps  $t$   $\times$  probabilité pour tous ses successeurs de rester dans leur état courant malgré la modification  $T_u^D \leftarrow t$ . Ré-échantillonner un suffisamment grand nombre de fois l'ensemble des variables manquantes en fonction de l'instanciation de toutes les autres tend à faire converger les échantillons d'épisodes vers la probabilité jointe de l'ensemble des infections selon les paramètres  $\Theta$  courants. Il s'agit ensuite de maximiser la vraisemblance des épisodes échantillonnés selon un optimiseur de type GEM, puis de ré-échantillonner de nouveaux épisodes et ainsi de suite jusqu'à convergence, avec pour objectif de maximiser la probabilité  $\prod_{D \in \mathcal{D}} P_{\Theta}(\{T_u^D\}_{u \in \mathcal{O}^D})$ . Une autre manière d'atteindre cet objectif est la méthode par passage de message donnée dans [Lokhov, 2016] dans le cadre du modèle NetRate. Cependant, dans ces deux travaux le processus d'apprentissage possède l'information de quels noeuds sont cachés dans chaque épisode. Si cette configuration du problème paraît assez naturelle en étude des dynamiques de métapopulations où chaque mesure est décidée par des experts en fonction de leur utilité ainsi que des coûts associés, la connaissance des noeuds cachés n'est pas toujours réaliste dans le contexte des réseaux sociaux. Outre le cadre de données collectées via des méthodes dynamiques telles que celles du chapitre 2, où l'on sait à chaque instant qui on écoute (et encore, des données annexes sur les autres utilisateurs pouvant être considérées à travers les comptes écoutés), dans le cadre de données collectées de manière passive via par exemple l'API *Sample Streaming* de *Twitter* (voir section 2.1.1) il peut être difficile de distinguer les non-infections des non-observations.

Notons également le travail de [He et al., 2016], qui se base sur les coûts d'apprentissage de [Narasimhan et al., 2015] pour proposer une méthode d'apprentissage PAC dans le cas de données d'infection manquantes. Cependant, ce travail ne permet pas l'extraction de dynamiques d'influence entre utilisateurs, car basé sur des fonctions de prédiction des infections finales sous la forme de sommes pondérées de fonctions de base binaires à la manière de [Du et al., 2012]. Ce travail se place aussi dans un cadre différent où les temps d'infection ne sont pas observés (on dispose seulement d'un ensemble de sources à faire correspondre à un ensemble d'infectés), à la manière de [Narasimhan et al., 2015] ou [Shaghaghian and Coates, 2016]. Néanmoins, dans [He et al., 2016] l'identité des noeuds cachés est inconnue, seul est donné (ou estimé) un paramètre de *retention*, correspondant au taux de noeuds observés dans chaque épisode, ce qui semble mieux cadrer avec le contexte des données issues des réseaux sociaux (avec par exemple seulement 1% de l'activité globale de *Twitter* délivrée à chaque instant par l'API *Sample Streaming*).

Un travail engagé pour poursuivre nos contributions en modélisation des processus de diffusion est à la jonction de ces approches, où l'on cherche à traiter ces deux types de configurations (infections cachées connues ou inconnues), dans le cadre de nos modèles DAIC, embedIC et recCTIC. Plus spécifiquement, pour le cas des infections cachées inconnues, il s'agit de considérer des distributions postérieures des temps d'infections de tous les non infectés supposés, selon les infections observées

et un prior dépendant du taux de rétention connu. Différentes possibilités sont envisagées, de l'utilisation d'échantillonnage MCMC comme dans [Wu et al., 2013] (cependant très complexe puisqu'il faut considérer la possible infection sur l'intervalle de temps  $[0; T]$  pour l'ensemble des non-infectés à chaque étape), à des approches plus globales type inférence variationnelle (s'appuyant par exemple sur des encodeurs des épisodes observés à base de réseaux transformeurs). Une autre approche serait d'exploiter le travail de [Zong et al., 2012], qui propose une méthode de reconstruction de cascades partiellement observées, par recherche d'arbres consistants selon un graphe de diffusion connu.

Au delà des non observations d'infections, pourrait également se poser la question des infections passives. Récemment, la distinction entre processus de diffusion passive (telle que la propagation de virus par exemple) et diffusion active, qui requiert des actions de la part des utilisateurs pour déclencher leur infection, a été soulignée dans [Milli et al., 2018]. La plupart des données étudiées dans le cadre de la modélisation de la diffusion sur les réseaux sociaux concernent des processus actifs, où l'on considère infectés seulement les utilisateurs ayant par exemple re-partagé une information. Le fait qu'un contenu ait atteint, et peut-être impacté, un utilisateur est ignoré si celui-ci n'a pas réalisé une action enregistrée en réaction. La question de la prise en compte des infections passives ne semble pas avoir été traitée dans la littérature, et paraît constituer une piste de travail intéressante, afin de ne pas considérer les multiples infections passives comme des tentatives de transmission échouées, sous des hypothèses de modélisation appropriées.

### 3.4.1.3 Extraction d'épisodes de diffusion

L'ensemble des travaux mentionnés jusqu'ici considère des épisodes de diffusion connus, supposant que l'on dispose de données regroupant un ensemble de séquences de participations d'utilisateurs à des diffusions particulières. Cependant, ces données ne sont pas nécessairement évidentes à obtenir selon le type de diffusion étudié, particulièrement lorsque les items propagés correspondent à des données complexes comme du contenu textuel par exemple. Une approche très populaire est celle de [Leskovec et al., 2009], qui a donné naissance au jeu de données MemeTracker. Diverses autres approches de suivi de contenu sur les noeuds des réseaux existent, comme par exemple celle de [Balali et al., 2013] pour la reconstruction de fils de discussion dans les blogs, mais celles-ci restent essentiellement heuristiques. Elles ne prennent pas en compte un modèle de diffusion sous-jacent. Une possibilité d'extension des modèles proposés, et notamment ceux basés sur de l'apprentissage de représentation, serait de les appliquer sur des données non structurées en épisodes, où les épisodes seraient formés selon les paramètres courants des modèles et selon diverses hypothèses de régularité, dans un processus de type Espérance-Maximisation. Il s'agit à chaque étape d'échantillonner des épisodes en regroupant des sous-ensembles d'évènements des utilisateurs du réseau selon leur probabilité de survenir en cascade en fonction des paramètres de diffusion courants (et éventuellement de similarité de contenu pour guider l'apprentissage), qui seront utilisés pour mettre à jour les paramètres de diffusion par maximisation de vraisemblance. Ce genre de travail paraît essentiel car il permet de s'attaquer à un spectre bien plus large de jeux de données, sans annotations ou regroupements d'évènements évidents, et conduit à l'obtention de fils d'évènements liés, aux contenus possiblement évolutifs, en plus de la découverte du graphe d'influence associé. L'idée se rapproche quelque peu de l'approche intéressante donnée dans [Barbieri et al., 2013a], pour la détection de communautés selon la vraisemblance de cascades associées. Plutôt que regrouper les noeuds du réseau selon les épisodes observés et les paramètres de diffusion courants, il s'agit de former les épisodes sur lesquels apprendre ces paramètres. L'idée de regrouper les noeuds pour gagner en régularité est également une piste prometteuse à poursuivre, notamment dans le cadre des approches par apprentissage de repré-

sentation, dans lesquels ces regroupements peuvent se faire plus efficacement selon des mixtures de distributions sur l'espace de représentation considéré.

### 3.4.2 Modèles propagatifs (plus) profonds

#### 3.4.2.1 VAE Smoothing et Composition récurrente

À la section 3.3.2.2, on liste diverses manières pour faire de l'inférence des séquences d'infecteurs dans le modèle recCTIC. Dans l'état actuel du modèle, l'inférence est réalisée par filtrage, c'est à dire que seules les informations passées sont utilisées pour déterminer les distributions (i.e., on considère  $q(I|D) = \prod_i^{|D|-1} q(I_i|D_{\leq i}, I_{< i})$ ). Or, cela peut mener à attribuer des probabilités importantes à des séquences d'infecteurs jusqu'à un indice  $i$  qui rendent impossibles (ou presque) les observations d'infection (ou de non-infection) succédant  $i$ . Bien que le gradient considéré en (3.27) tende à maximiser la probabilité des séquences d'infections correspondant à de fortes vraisemblances des épisodes observés, il est probable que la convergence soit bien plus lente qu'avec une approche par lissage, c'est à dire employant des distributions d'inférence conditionnées sur l'épisode global.

L'inférence par lissage peut être envisagée pour l'approche de [Lamprier, 2019], en considérant une distribution  $q(I|D)$  donnée par :

$$q(I|D) = \prod_i^{|D|-1} q_\phi(I_i|D, I_{< i}) \quad (3.28)$$

avec  $\phi$  les paramètres variationnels du modèle. Les facteurs de la distribution  $q$ , concernant l'infecteur de chaque infection d'index  $i \in \{0, \dots, |D| - 1\}$  de la séquence, sont alors déterminés par un réseau de neurones  $q_\phi$ , prenant en entrée la totalité de l'épisode considéré et l'ensemble des états récurrents (construits selon (3.14) et la liste des infecteurs  $I_{< i}$ ) pour les infectés de l'épisode avant la position  $i$ , et produisant alors les paramètres d'une distribution catégorielle.

Pour l'inférence variationnelle dans les séquences basées sur un modèle de transition à états cachés, [Krishnan et al., 2017] propose de s'appuyer sur un RNN bi-directionnel pour prendre en compte l'ensemble des observations. Basé sur un processus de dépendance linéaire, ce travail définit une ELBO factorisée, se décomposant en sous-problèmes indépendants, car l'état inféré au temps  $t$  ne dépend alors que de l'état inféré au temps  $t - 1$  et des observations de la séquence (hypothèse markovienne sur les états successifs inférés). Une telle factorisation n'est pas possible dans notre cas, puisque chaque nouvel état dépend de l'ensemble des infecteurs précédents, et qu'une approximation par champs moyens tendrait vraisemblablement à des distributions d'inférence moins performantes que celle nous avons utilisée dans [Lamprier, 2019]. Nous proposons donc de considérer des échantillonnages sur séquences complètes dans notre cas, et si une trop grande variance est observée dans les estimateurs de gradients, une attention particulière pourra être portée sur la réduction de cette variance, notamment par l'utilisation de baselines plus évoluées, fonctions d'avantage ou techniques de bootstrapping classiques en apprentissage par renforcement [Greensmith et al., 2004], ou encore par *Rao-Blackwellisation* [Casella and Robert, 1996]. Notons qu'il n'y a par ailleurs pas de raison évidente que la variance soit particulièrement plus élevée que pour les expérimentations dans [Lamprier, 2019], utilisant des stratégies d'inférence par filtrage.

Une plus grande difficulté concerne le choix du réseau d'encodage considéré, pour la prise en compte des événements utiles de l'épisode d'observations. Considérer des réseaux RNN comme dans [Krishnan et al., 2017], revient à se ramener à une prise en compte des épisodes sous la forme de séquences. Contrairement au modèle RMTTPP considéré en section 3.3.2.2, cela ne concerne que la procédure d'inférence, ce qui a un plus faible impact sur le modèle. Néanmoins d'autres architectures

paraissent mieux adaptées dans notre cas, notamment des réseaux *Transformer* qui seraient à même de focaliser l'attention sur les éléments de la branche d'infections du nœud considéré pour l'inférence, conditionnellement aux infecteurs inférés sur les évènements précédents. Diverses architectures de ce type pourront être envisagées.

Par ailleurs, une limite de l'approche recCTIC concerne le fait qu'elle ne considère que l'influence du premier infecteur de chaque nœud pour composer son état latent. Or, l'ensemble des branches d'ancêtres menant à une infection peuvent contenir des informations utiles à la prédiction de la suite de l'épisode. Ce travail sur l'inférence de recCTIC sera également l'occasion de revenir sur ce point, en passant d'une structure inférée arborescente à une structure de graphe dirigé acyclique (DAG) pour chaque diffusion considérée. Il s'agira alors d'inférer des ensembles d'infecteurs pour chaque nouvelle infection, plutôt qu'un unique individu. La fonction de construction d'état récurrente 3.14 devra également être revue pour prendre en compte des états multiples en entrée de chaque application. Une possibilité prometteuse pour cette fonction serait de se baser sur le réseau de composition défini dans Topo-LSTM [Wang et al., 2017a], initialement prévu pour travailler sur un graphe statique connu, mais que l'on peut alors appliquer sur notre DAG construit incrémentalement pour chaque épisode.

### 3.4.2.2 Deep Heat Diffusion

Une idée intéressante serait également de poursuivre le travail initié dans [Bourigault et al., 2014b], pour l'étendre à de multiples sources et de le replacer dans un cadre probabiliste. L'idée de notre contribution dans [Bourigault et al., 2014b], présentée en section 3.2.1, était de s'appuyer sur le noyau de la chaleur pour définir des représentations d'utilisateurs du réseau, de manière à ce que les utilisateurs soient atteints par la chaleur se diffusant depuis la source de diffusion dans l'ordre chronologique des observations d'infection. Comme mentionné précédemment cela présente un premier défaut de ne pas permettre la prise en compte conjointe de plusieurs sources de diffusion. Une possibilité pour dépasser cela serait de s'appuyer sur une fonction  $T(t, x) : \mathbb{R}^+ \times \mathcal{X} \rightarrow \mathbb{R}$  considérant plusieurs sources de chaleur disposées dans un espace euclidien  $\mathcal{X}$  et retournant la chaleur à l'instant  $t$  pour la position  $x \in \mathcal{X}$  selon :

$$T(t, x) = \sum_{u \in S(t)} \phi(t - t^u, z_u, x), \text{ with } \phi(t, y, x) = \begin{cases} t = 0 : \delta(\|x - y\|) \\ t > 0 : K(t, y, x) \end{cases} \quad (3.29)$$

où  $S(t) \subseteq \mathcal{U}$  correspond à l'ensemble des sources de diffusion avant l'instant  $t$ ,  $t^u$  l'instant d'injection d'un Dirac de chaleur en la position émetteur  $z_u \in \mathcal{X}$  de l'utilisateur  $u$  (correspondant à l'instant d'infection de l'utilisateur  $u$ ) et  $K(t, y, x)$  le noyau de la chaleur. On considère  $T(0, x) = 0$  pour tout  $x$  (sauf éventuellement au niveau de la représentation du nœud "monde"  $u_0$ ). On propose de faire correspondre des représentations récepteur des nœuds à des positions permettant de bien expliquer les infections selon cette fonction de chaleur.  $T^D(t, \omega_v)$  correspond alors à la chaleur au point de la représentation récepteur de  $v$  au temps  $t$  de l'épisode D, avec  $S(t) = \{u \in U^D \mid t_u^D < t\}$  l'ensemble des sources actives au temps  $t$ .

La considération de cette fonction, qui découle de la somme des noyaux de chaleur des différentes sources précédant chaque instant (démonstration possible par induction), correspond à émettre une hypothèse d'additivité des influences, comme dans les approches de type *Linear Threshold* (LT). Comparé aux approches LT classiques cependant, la pression sociale évolue au cours du temps, selon donc des processus continus dans l'espace. Le processus associé est illustré en figure 3.10, qui représente un processus de diffusion sous la forme d'une réaction en chaîne où les infections se déclenchent les

unes après les autres en fonction de la chaleur reçue, chaque nouvelle infection relâchant de la chaleur supplémentaire dans le système, à la manière de bombes se déclenchant les unes après les autres en fonction de leurs proximités respectives.

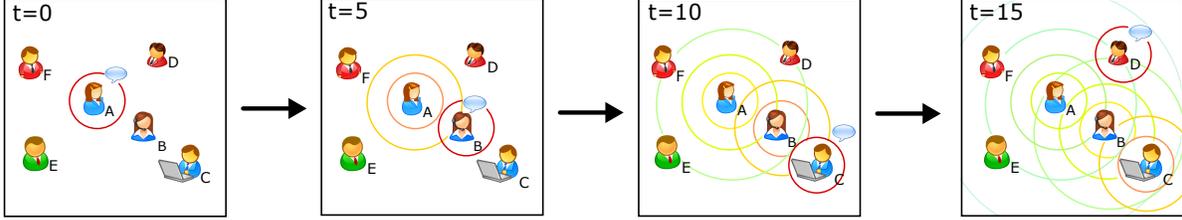


FIGURE 3.10 – Réaction en chaîne de diffusion sur quatre étapes. Les utilisateurs A,B,C et D sont infectés à tours de rôle. Chaque nouvelle infection réinjecte de la chaleur dans le système. Les cercles représentent la chaleur se diffusant dans l'espace.

Selon ce processus, on pourrait considérer qu'une infection de l'utilisateur  $v$  est déclenchée au temps  $t$  d'une diffusion  $D$ , sachant qu'elle n'est pas survenue avant  $t$ , selon la fonction de hasard (ou taux d'infection instantanée [Gomez-Rodriguez et al., 2011], ou encore intensité conditionnelle selon le langage des processus temporels ponctuels [Du et al., 2016]) :

$$\lambda^D(t, v) = \sigma(\alpha_v T^D(t, \omega_v) + \beta_v) \quad (3.30)$$

où  $\sigma$  correspond à la fonction sigmoïde.  $\alpha_v$  et  $\beta_v$  sont des paramètres de prior pour  $v$ , agissant sur sa propension générale à être infecté, indépendamment des utilisateurs infectés par le passé. Selon une fonction de hasard  $\lambda(t)$ , la fonction de survie  $S(t) = 1 - F(t)$ , qui représente la probabilité qu'un évènement n'arrive pas avant le temps  $t$ , est donnée par  $S(t) = \exp(-\int_0^t \lambda(s) ds)$ . On a alors  $P(\tilde{t}_v^D \geq t) = \exp(-\int_0^t \lambda^D(s, v) ds)$ . En exploitant que  $\lambda(t) = \frac{f(t)}{S(t)}$ , avec  $f$  la fonction de densité à  $t$ , on a enfin la densité d'un déclenchement à  $\tilde{t}_v^D$  pour  $v$  dans  $D$  donné par  $P^D(\tilde{t}_v^D, v) = \lambda^D(t, v)P(\tilde{t}_v^D \geq t)$ . Ce genre de modèle correspond à un processus ponctuel de Hawkes [Hawkes, 1971].

Afin d'éviter les limites énoncées pour le modèle de [Bourigault et al., 2014a], qui mélange comme NetRate temps d'infection et probabilité d'infection, on propose de considérer des délais  $\tau$  entre le moment où l'infection est décidée  $\tilde{t}_v^D$  (selon le taux d'infection instantanée de (3.30)) et l'instant  $t_v^D = \tilde{t}_v^D + \tau$  où elle est effective (instant d'observation et d'injection de nouvelle énergie dans le système). Cela permet de définir un processus asynchrone dans lequel les délais d'infection n'impactent pas trop fortement les probabilités d'infection. Comme dans [Saito et al., 2009], ce délai peut être généré selon une loi exponentielle, de paramètre  $r_v$ , spécifique à chaque utilisateur récepteur  $v$ .

L'apprentissage des représentations et paramètres de ce modèle apparaît néanmoins complexe, car il requiert la marginalisation sur tous les temps d'infection possibles pour tous les utilisateurs du réseau. On a en effet :

$$\log P_{\Theta}(D) = \sum_{v \in \mathcal{U}^D, t_v^D > 0} \log \int_0^{t_v^D} P^D(t_v^D - \tau, v) \frac{1}{r_v} \exp(-r_v \tau) d\tau - \sum_{v \notin \mathcal{U}^D} \log \int_0^T \lambda^D(s, v) ds \quad (3.31)$$

qui peut s'optimiser en inférant  $\tau$  pour chaque utilisateur infecté selon une distribution d'instrumentation  $q_v^D(\tau)$ , et où  $\log \int_0^T \lambda^D(s, v) ds$  peut être estimé par échantillonnage de Monte-Carlo.

Une possibilité pour simplifier le problème serait de modifier quelque peu le modèle et considérer que la chaleur reçue par chaque utilisateur s'accumule au fil du temps, jusqu'à déclencher l'infection.

On considère alors une chaleur cumulée  $U^D(t, v) = \int_0^t T^D(s, \omega_v) ds$ , définie pour tout utilisateur  $v$  et tout temps  $t$  de chaque épisode  $D$ . Cela permet de travailler avec une fonction  $U^D(t, v)$  croissante plus facilement manipulable, plutôt que d'avoir à gérer les variations de température  $T^D(s, \omega_v)$  conditionnant directement les probabilités d'infection du modèle initial. Soit un seuil  $\gamma_v$  échantillonné en début de chaque épisode pour chaque utilisateur  $v$ , selon une loi par exemple Log-Normale de paramètres  $\mu_v$  et  $\sigma_v$  appris pour chaque  $v$ . Soit la quantité  $u^D(t, v) = \sigma(\alpha_v U^D(t, v) + \beta_v)$ . Au cours du processus, tout utilisateur  $v$  est déclenché lorsque  $u^D(t, v) = \gamma_v$  (en pratique on travaille par pas de Euler et on inspecte si  $u^D(t, v) > \gamma_v$  après le pas, diverses stratégies peuvent être considérées pour optimiser l'exécution). La fonction de survie en  $T$  est donnée par  $P(\tilde{t}_v^D > T) = P(\gamma_v > u^D(T, v)) = 1 - F_{\gamma_v}(u^D(T, v))$ , avec  $F_{\gamma_v}$  la fonction de répartition de la loi Log-Normale pour le seuil  $\gamma_v$ , sachant qu'il suffit alors de considérer la probabilité que le seuil ne soit pas dépassé au temps  $t$ . Cela induit une vraisemblance bien plus facile à optimiser que dans le modèle initial, car  $U^D(t, v)$  peut s'obtenir efficacement par simulation, pour tout  $v$  simultanément.

### 3.4.2.3 RL / GANs / Inverse RL

Dans l'ensemble de ce chapitre, nous nous en sommes tenus à des approches principalement basées sur un apprentissage par maximum de vraisemblance (MLE). Néanmoins, il est bien connu que dans un cadre prédictif séquentiel, l'apprentissage par MLE souffre d'un certain nombre de limitations, au premier rang desquelles le biais d'exposition (*Exposure Bias*) [Bengio et al., 2015], très étudié pour des tâches de génération de la langue notamment [Ranzato et al., 2015]. L'apprentissage par MLE sur des séquences, introduit sous le nom de *Teacher Forcing* dans [Williams and Zipser, 1989], se fait naturellement en considérant pour chaque nouvel élément (e.g. *token*) l'historique des éléments passés selon les observations dans les séquences d'entraînement, et non les prédictions du modèle. Cela crée un décalage entre les conditions d'entraînement et les conditions d'utilisation, car l'objectif est un modèle génératif auto-conditionné. En génération pure les erreurs tendent à s'accumuler. Le modèle se trouve alors conditionné par des séquences jamais vues en entraînement, ce qui accroît les possibilités d'erreurs et tend à la divergence de la génération. Ce biais d'exposition est également présent pour les modèles de diffusion récurrents.

D'importants efforts de recherche ont été portés sur ce biais d'exposition. Inspiré par [Venkatesh et al., 2015], [Bengio et al., 2015] a par exemple défini une variation de *Teacher Forcing* dans laquelle les éléments d'observation sont graduellement remplacés par des éléments prédits. Une autre approche, *Professor Forcing* [Lamb et al., 2016], propose d'entraîner le générateur de manière à ce que les états de son réseau récurrent soient similaires, qu'il soient issus d'un conditionnement *Teacher Forcing* ou d'une génération libre auto-conditionnée, par une méthodologie d'apprentissage par renforcement avec réseau discriminatoire adverse à la manière des GANs (*Generative Adversarial Networks*) [Goodfellow et al., 2014].

Dans le cadre de la génération textuelle, de nombreuses approches se sont focalisées sur les séquences produites par *Beam Search* [Furcy and Koenig, 2005], qui est un processus glouton de recherche en faisceau qui permet d'approcher les séquences les plus probables d'un générateur. L'idée de [Wiseman and Rush, 2016] est d'optimiser une mesure de BLEU sur les textes générés par *Beam Search*. Dans la même veine, [Zhang et al., 2019] propose une variante de *Beam Search* maximisant la mesure BLEU, afin de travailler à augmenter la probabilité des séquences sélectionnées de cette manière. Notons que, dans le cadre du résumé abstraitif, nous avons contribué à cette lignée de travaux en proposant une procédure de *Beam Search* orientée par un discriminatoire, appris itérativement sur les séquences générées à chaque époque, afin de favoriser les séquences ressemblant à des résumés

humains. Cela permet de limiter le biais d'exposition, sans avoir recours à une métrique experte, et sans avoir à modifier les poids du générateur [Scialom et al., 2020b]. Ces méthodes basées sur des processus de recherche en faisceau paraissent cependant difficilement applicables en diffusion, où le modèle n'est pas amené à choisir à chaque pas de temps dans un ensemble de tokens possibles connaissant le passé, mais détermine des infections à des temps continus selon des dépendances arborescentes.

Diverses méthodes par apprentissage par renforcement ont été développées pour limiter le biais d'exposition. Dans le cadre des approches de génération textuelle encore, [Ranzato et al., 2015] propose notamment de considérer des métriques globales type BLEU sur les textes produits comme récompenses dans des méthodes *Policy Gradient* type *Reinforce*. [Paulus et al., 2017] adopte une approche similaire pour le résumé abstraitif, en considérant la métrique ROUGE sur les résumés produits. Enfin, dans notre contribution dans [Scialom et al., 2019], nous avons proposé de renforcer une métrique de réponse à des questions générées automatiquement à partir du texte initial à résumer. Dans le cadre de la prédiction de diffusion, on note l'approche de [Yang et al., 2019], qui contrôle les prédictions des processus temporels ponctuels à base de RNNs, via une supervision sur la taille des cascades prédites. Il serait intéressant d'étudier l'impact de ce genre de régularisation sur notre modèle recCTIC.

Plutôt que définir des métriques expertes à renforcer en fonction de la tâche considérée, divers travaux se basent sur les GANs pour contrôler les séquences générées. L'idée est de se baser sur des discriminateurs adverses qui apprennent à distinguer les séquences réelles des données de celles du générateur. La discrimination est réputée être une tâche bien plus simple que la génération, notamment car il suffit d'identifier des marqueurs particuliers des distributions comparées, plutôt que d'avoir à modéliser les distributions complètes, avec les biais d'exposition associés (voir par exemple dans [Zellers et al., 2019] qui reporte un taux de bonne discrimination de 97% sur de la génération d'articles longs, avec des générateurs appris par MLE selon une architecture identique à celle des discriminateurs considérés). Par ailleurs, les décodeurs séquentiels souffrent aussi de devoir prendre des décisions itérativement de manière gloutonne, là où les discriminateurs travaillent sur des séquences complètes. Dans le cadre des données temporelles continues, l'utilisation des modèles adverses se fait relativement bien, grâce à un flux de gradient qui traverse le discriminateur pour atteindre les paramètres du générateur [Luo et al., 2018], comme dans le cas classique sur des images statiques par exemple [Radford et al., 2016]. Pour des données séquentielles discrètes néanmoins, la tâche est autrement plus ardue car ce genre de retro-propagation du gradient à partir d'un coût de discrimination n'est pas possible. Il s'agit alors de se tourner vers des approches de renforcement dont la fonction de récompense est le score de sortie du discriminateur, soit la probabilité que la séquence générée soit issue de la distribution des données d'entraînement. Étant donnée la parcimonie des récompenses associées, une large part de la littérature pour les approches type GAN pour les données séquentielles discrètes s'est concentrée sur la densification de ces récompenses, afin de guider plus efficacement l'apprentissage du générateur. Des discriminateurs comparatifs [Li et al., 2017, Zhou et al., 2020], séquentiels (pour donner des récompenses à chaque pas de temps plutôt qu'en seule fin de séquence) [Semeniuta et al., 2018, de Masson d'Autume et al., 2019], ou encore spécifiquement adaptés à des tâches de complétion de séquence (avec parties masquées) [Fedus et al., 2018] ont ainsi été proposées. On note également l'approche intéressante MALIGAN décrite dans [Che et al., 2017], qui propose quant à elle de traiter le problème de la cible mouvante induite par l'apprentissage itératif de la fonction de récompense. L'idée est d'exploiter le fait qu'à l'optimum, pour un discriminateur  $D_\phi$  de capacité suffisante et un nombre suffisant de séquences de la distribution des données  $p_{data}$  et de la distribution du généra-

teur  $\pi$ , on a  $D_\phi(\tau) = \frac{p_{data}(\tau)}{p_{data}(\tau) + \pi(\tau)}$  pour toute trajectoire  $\tau$ . On a alors aussi  $p_{data}(\tau) = \frac{D_\phi(\tau)}{1 - D_\phi(\tau)} \pi(\tau)$ . Cela permet aux auteurs de définir un processus d'échantillonnage préférentiel (IS) permettant de minimiser la divergence  $D_{KL}(p_{data} || \pi_\theta)$  selon les paramètres  $\theta$  du générateur, ce qui est un objectif plus stable que de maximiser  $\mathbb{E}_{\tau \sim \pi_\theta} [D_\phi(\tau)]$ . Néanmoins, dans une contribution récente présentée dans [Scialom et al., 2020a], nous montrons que pour diverses tâches de génération textuelle, la définition de stratégies d'échantillonnage proches du mode de la distribution  $\pi_\theta$ , permet une meilleure convergence vers des générateurs efficaces (un mix de ces deux approches pourrait être néanmoins envisagé). Notons finalement que ces discussions dépassent largement le cadre des séquences textuelles, et ont des implications dans de nombreux domaines, y compris pour de l'apprentissage RL par imitation, comme dans le cadre du modèle populaire GAIL défini dans [Ho and Ermon, 2016].

Nous proposons d'envisager ce type d'approches dans le cadre de la génération de cascades de diffusion, où les GANs n'ont pas été du tout étudiés, à l'exception de l'approche récente *DiffusionGAN* [Zhuo et al., 2019] qui emploie une méthode avec discriminateur adverse pour déterminer des projections continues statiques d'utilisateurs selon des épisodes observés. Cette approche n'a pas d'objectif prédictif (uniquement de représentation) et n'est pas confrontée aux difficultés de retropropagation du gradient mentionnées ci-dessus pour le cas discret, car travaillant directement dans l'espace de représentation, sans prise de décision discrète nécessaire. Le cadre prédictif a contrario implique des décisions de sélection d'utilisateurs infectés, qui doivent être prises en compte au cours de l'apprentissage pour éviter des décalages entre conditions d'entraînement et d'utilisation, qui viendraient renforcer les difficultés de biais d'exposition associés à la prédiction d'évènements séquentiels. Dans le cadre de réseaux récurrents tels que *recCTIC*, ou bien également dans des processus de diffusion de chaleur tels que ceux présentés en section précédente, l'objectif sera de construire des politiques de diffusion menant à des cascades générées qu'un discriminateur ne sera pas capable de distinguer des épisodes d'entraînement. Ce genre de travail pour données temporelles avec dépendances arborescentes associées, n'a à notre connaissance jamais été abordé dans la littérature, dans le cadre de la diffusion et au delà. Une autre possibilité serait de s'inspirer des approches en renforcement inverse, très proches des GAN RL mais où l'accent est plus porté sur la découverte des fonctions de récompense expliquant les données d'entraînement, dont le travail dans [Li et al., 2018] a montré le potentiel pour la génération de séquences.

### 3.4.3 Modèles en ligne

Bien sûr, il serait dommage de conclure ce chapitre sur la diffusion sans discuter de la possibilité de modèles en ligne, se raffinant avec des données arrivant en continu, et ainsi faire le pont avec les travaux du chapitre 2. On distingue deux types de travaux autour de l'apprentissage en ligne de modèles de diffusion : des modèles producteurs de contenu d'une part, qui établissent leurs modèles en injectant du contenu dans le réseau puis en étudiant sa propagation via des algorithmes de bandits<sup>2</sup> (ce qui nous rapproche d'une implémentation de l'opérateur d'intervention *do* discuté à la section 3.4.1.1), et des modèles consommateurs de contenu d'autre part, qui n'ont pas la possibilité de poster du contenu pertinent et doivent s'appuyer sur des contenus se diffusant naturellement dans le réseau pour capturer les dynamiques de propagation en jeu.

2. Notons que les modèles *Cascading Bandits* du type de [Kveton et al., 2015], dont le nom peut porter à confusion, n'entrent pas dans ce cadre car ne traitant pas de cascades de diffusion mais travaillant sur le modèle de cascade en recommandation de produits, qui correspond à un modèle de comportement d'utilisateur face à une liste d'items.

### 3.4.3.1 Modèles producteurs : Apprentissage par injection de contenu

La très vaste majorité des travaux pour la modélisation de dynamiques de diffusion en ligne concerne des tâches de maximisation d'influence : comme dans le travail fondateur de [Kempe et al., 2003] dans un cadre hors-ligne, il s'agit de définir l'ensemble  $S$  d'utilisateurs sources à qui donner un contenu pour en maximiser la diffusion, selon un budget  $k$  et un réseau de probabilités de transmission de contenu  $\Theta$  :  $\arg\max_{S \subseteq \mathcal{U}, |S|=k} \sigma(S, \Theta)$ , avec  $\sigma(S, \Theta)$  l'espérance du nombre de nœuds atteints par un contenu donné aux utilisateurs dans  $S$  selon le réseau  $\Theta$ . Pour optimiser cet objectif en ligne, la plupart des approches passent par la modélisation des relations d'influence  $\Theta$ , inconnues a priori. Néanmoins, on peut citer l'approche de [Lagrée et al., 2017] qui, dans un soucis de passage à l'échelle dans le cas des très grands réseaux, cherche à maximiser l'impact d'une campagne de diffusion sans modéliser le graphe d'influence sous-jacent. Il s'agit de sélectionner uns à uns des individus, leur donner le contenu concerné par la campagne et observer les nœuds impactés par ce contenu au bout d'une certaine période. Se plaçant dans un cadre de persistance des infections (un nœud impacté sur un essai précédent dans la campagne ne peut pas l'être à nouveau), l'utilité des utilisateurs décroît avec leur nombre de sélections, menant à la définition d'une forme de *Rested Bandit* (voir section 2.3.3), où le score de sélection est déterminé selon une borne supérieure de l'intervalle de confiance d'un estimateur *Good-Turing* de masse manquante pour chaque individu. Dans une autre approche *model-free*, [Carpentier and Valko, 2016] propose de ne s'intéresser qu'à l'apprentissage de liens direct de diffusion long terme source-infectés finaux (i.e., avec la diffusion pouvant transiter par des nœuds intermédiaires sans modélisation explicite de ces chemins). Enfin, dans un cadre similaire, l'approche de [Vaswani et al., 2017] propose une approximation linéaire de liens d'infection longs terme, dans un algorithme de type LinUCB, selon des caractéristiques de relations définies heuristiquement en fonction de la structure du réseau. Toutes ces approches sont intéressantes d'un point de vue efficacité pour la maximisation d'influence, mais ne permettent pas une réelle modélisation des dynamiques de l'information qui nous intéresse dans ce manuscrit.

D'autres approches, basées donc sur des modèles de diffusion, nous intéressent plus particulièrement, car impliquant la construction d'un graphe d'influence en ligne, par le biais d'une recherche des meilleurs influenceurs du réseau. Dans un cadre combinatoire, la plupart de ces approches s'appuie à chape étape sur la construction d'un super-bras composé des  $k$  influenceurs sélectionnés par un algorithme oracle approximant  $\arg\max_{S \subseteq \mathcal{U}, |S|=k} \sigma(S, \Theta)$  selon les relations courantes  $\Theta$ . La fonction  $\sigma(S, \Theta)$  étant monotone et sous-modulaire en  $S$ , il est alors possible de déterminer une approximation du problème de maximisation associé en un temps polynomial [Nemhauser et al., 1978]. En particulier, l'algorithme de [Kempe et al., 2003] fournit une approximation  $\tilde{S}$  de la solution optimale  $S^*$  dont le score  $\sigma(\tilde{S}, \Theta)$  est au moins supérieur à  $(1 - \frac{1}{e} - \epsilon)\sigma(S^*, \Theta)$  avec une probabilité supérieure à  $1 - \frac{1}{|\Theta|}$ , avec  $|\Theta|$  le nombre de relations du graphe et  $\epsilon$  l'erreur d'estimation de  $\sigma(S, \Theta)$  par Monte-Carlo à chaque étape, diminuant avec le nombre de simulations effectuées. Basé sur cet oracle, [Chen et al., 2016a] applique l'algorithme de bandit combinatoire CUCB au problème de maximisation d'influence en ligne, démontrant un regret sous-linéaire par rapport à l'approximation selon l'oracle de  $\sigma(S^*, \Theta^*)$ , avec  $\Theta^*$  les "vraies" probabilités de diffusion, inconnues au démarrage du processus (mais s'appuyant sur une structure de graphe connue). Dans un esprit similaire, [Lei et al., 2015] modélise l'incertitude sur les relations d'influence selon une loi Beta, prior conjugué de la loi de Bernouilli, permettant l'exploration de l'espace des paramètres à la manière des approches type *Thompson Sampling*, tout en s'appuyant sur l'oracle de [Kempe et al., 2003] pour produire les nœuds sources à chaque étape. Le travail dans [Wen et al., 2017] propose également une approche similaire à [Chen et al., 2016a], mais avec généralisation linéaire en fonction de caractéristiques disponibles sur les relations du graphe,

afin de proposer un algorithme de type LinUCB pour la maximisation d'influence.

Cependant, l'ensemble de ces travaux sont basés sur l'hypothèse très forte de l'observabilité des activations des liens du graphe. En plus de connaître l'ensemble des noeuds qui ont été impactés par un contenu posté sur les sources  $S_t$  à l'itération  $t$ , ces approches nécessitent de savoir pour chacun d'entre eux quels liens ont été suivis par le contenu pour les atteindre, afin de mettre à jour les probabilités d'activation  $\Theta$  correspondantes. Cette information étant rarement disponible sur les réseaux sociaux, l'approche dans [Vaswani and Lakshmanan, 2015] propose d'intégrer un mécanisme d'assignation de crédit à partir de la seule information de l'identité des nœuds infectés à chaque itération. Trois variantes sont considérées, dont une version en-ligne de l'apprentissage par maximum de vraisemblance de l'algorithme IC, considérant les épisodes de diffusion les uns après les autres sans stockage<sup>3</sup>, et une approche fréquentiste simpliste qui choisit d'attribuer tout le crédit de chaque infection à l'un des candidats échantillonné uniformément à chaque itération. Étonnamment, c'est cette dernière approche fréquentiste qui semble obtenir les meilleurs résultats, à la fois pour la maximisation d'influence et pour l'estimation du graphe de diffusion. Il est probable que ceci soit dû à un sur-apprentissage des probabilités de diffusion via la version MLE en début de processus, ce qui réduit largement les capacités d'exploration. Prenons l'exemple d'un nœud  $v$  avec deux prédécesseurs  $u_1$  et  $u_2$ . Si en début de processus  $u_1$  est infecté à plusieurs reprises avant  $v$  sans que  $u_2$  le soit, la probabilité  $\theta_{u_1,v}$  devient si élevée que lorsque l'on observe  $u_2$  en même temps que  $u_1$  avant  $v$  par la suite, le crédit de cette possible diffusion de  $u_2$  vers  $v$  est très fortement "cachée" par  $\theta_{u_1,v}$  (cela rejoint les discussions de la section 3.1.3). Pour contrer cela il faut accorder des bonus d'exploration élevés pour les liens peu encore considérés, du type de ceux de CUCB qui garantissent que l'oracle finisse par sélectionner les noeuds sources de ces liens. Or, les comparaisons des mécanismes de *feedback* dans [Vaswani and Lakshmanan, 2015] ne sont pas effectuées selon CUCB, jugé très bon en théorie, mais menant à une exploration bien trop importante pour être compétitif en pratique.

Une première piste pour dépasser cela serait de travailler sur des versions avec représentations distribuées des nœuds du graphe, avec probabilités de diffusion dépendant des distances entre représentations des nœuds concernés, comme dans notre approche embIC présentée à la section 3.2.2, afin d'éviter ces biais de sur-apprentissage, d'autant plus problématiques dans un cadre en ligne (notons que l'aspect sous-modulaire de la fonction  $\sigma(S, \Theta)$  n'est pas affecté par cette modification, sachant que des relations  $\Theta$  explicites peuvent être extraites à partir des représentations - des bonus d'exploration peuvent aussi être ajoutés aux probabilités  $\Theta$  ainsi extraites, avant sélection des sources). D'autre part, les bonus d'exploration accordés dans ces approches existantes sont appliqués sur chaque lien de manière individuelle. Une possibilité serait d'attribuer ces bonus, non seulement en fonction de leur incertitude propre, mais également en fonction de l'incertitude des liens auxquels ils ont tendance à donner accès, afin d'améliorer l'exploration en se focalisant sur la sélection de nœuds hubs. Cela serait par ailleurs particulièrement utile dans des cas où il n'est pas possible de sélectionner certains nœuds comme sources, du fait de contraintes d'accès à ces nœuds.

Une limite des approches existantes pour la découverte du graphe de diffusion d'une communauté est à mon sens de se focaliser sur un oracle visant à la sélection des sources les plus influentes. Si cette approche est pertinente pour des problématiques de maximisation de diffusion, par exemple pour une campagne publicitaire, elle l'est moins si l'objectif est principalement de comprendre les dynamiques de transmission de l'information dans un réseau. La découverte des relations d'influence

3. S'appuyant sur des garanties théoriques pour problèmes convexes (ce qui est le cas pour IC selon la reparamétrisation de [Netrapalli and Sanghavi, 2012] discutée en section 3.2.2), garantissant une erreur moyenne au bout de  $T$  itérations en  $\mathcal{O}(\frac{1}{\sqrt{T}})$  par rapport à une optimisation classique hors-ligne [Zinkevich, 2003].

n'est en effet qu'une annexe dans les approches existantes, menant à des graphes de diffusion appris très incomplets : les zones de faible potentiel en terme de propagation sur le réseau sont délaissées au profit des zones plus denses. Plutôt que de s'intéresser à combien d'utilisateurs seront impactés par un contenu se diffusant, une voie potentiellement utile serait de chercher à prédire quels nœuds (ou groupes de nœuds) le seront, et définir ainsi une méthode de sélection active des nœuds à cibler, afin d'atteindre des nœuds spécifiques du réseau au cours de la diffusion. Soit une fonction Oracle  $S(X, \Theta)$ , prenant en entrée un ensemble de nœuds du réseau à atteindre  $X \in \mathcal{U}$  et un ensemble de paramètres de diffusion  $\Theta$ , qui retourne une approximation de l'ensemble de nœuds  $S^*$  à partir desquels la probabilité d'atteindre tous les nœuds cible de  $X$  est maximisée :  $S^* = \arg \max_{S \in \mathcal{S} \subseteq \mathcal{U}, |S|=k} F_{\Theta}(S, X)$ , avec  $F_{\Theta}(S, X) = \prod_{x \in X} F_{\Theta}(S, x)$ , où  $F_{\Theta}(S, x)$  correspond à la probabilité d'atteindre le nœud  $x$  à partir des nœuds de  $S$  selon les probabilités de diffusion  $\Theta$ .  $\mathcal{S} \subseteq \mathcal{U}$  correspond à l'ensemble des nœuds auxquels on a accès (ceux à qui on peut donner un contenu). Bien sûr si  $|X| \leq k$  et que  $X \subseteq \mathcal{S}$ , le problème est trivial car il suffit de sélectionner les éléments de  $X$  comme sources. Cependant, si  $|X| \gg k$  ou si  $|X \setminus \mathcal{S}| > 0$ , il s'agit de bien connaître les chemins reliant probablement ces différents nœuds. Ce genre d'Oracle peut être défini par une méthode glouton sur des approximations Monte-Carlo de  $F_{\Theta}(S, x)$ , cette fonction étant également monotone et sous-modulaire en  $S$  pour tout  $x$ , pour atteindre le même genre d'approximation que pour la maximisation d'influence. L'objectif est alors de minimiser le pseudo-regret  $\sum_{t=1}^T \mathbb{E}_{X_t \sim P_X(X_t)} [F_{\Theta^*}(S(X_t, \Theta^*), X_t) - F_{\Theta_t^*}(S(X_t, \Theta_t), X_t)]$ , où  $\Theta^*$  correspond aux vrais paramètres diffusion,  $\Theta_t$  correspond à leur estimation après  $t$  itérations et où  $P_X$  est une distribution sur les ensembles de nœuds  $X$  à atteindre. Selon la définition de la distribution  $P_X$ , on pourra se focaliser sur la diffusion vers des nœuds cibles spécifiques avec  $P_X$  une distribution resserrée sur ces nœuds, ou bien au contraire s'intéresser à apprendre au mieux la totalité du graphe de diffusion par l'utilisation d'une distribution  $P_X$  uniforme sur l'ensemble des  $X$  possibles. Notons par ailleurs que, selon [Narasimhan et al., 2015], on sait que  $F_{\Theta}(S, x)$  est une fonction Lipsitchienne par rapport aux paramètres  $\Theta$ , ce qui constitue un bon point de départ pour la quantification théorique du regret associé au processus. Enfin, ce genre de travail pourrait également être réalisé en considérant différents types de contenus diffusés, comme cela est fait par exemple dans le modèle TIM [Chen et al., 2015] dans le cadre de la maximisation d'influence hors-ligne.

### 3.4.3.2 Modèles consommateurs : Apprentissage par suivi de flux

L'injection de contenu dans le réseau pour en étudier les dynamiques n'est pas toujours possible. Il faut avoir accès à un ensemble de sources potentielles à qui donner du contenu. En outre il faut être à même de formuler du contenu correspondant au type de données dont on veut étudier la propagation sur le réseau. Selon les cas et le contenu injecté sur les sources, il se peut que les dynamiques observées ne correspondent pas aux dynamiques naturelles du réseau. Pour ces cas, il n'est pas possible d'avoir recours aux modèles producteurs discutés à la section précédente. Nous nous tournons alors vers des modèles plutôt consommateurs de données, qui travaillent par analyse de flux et suivi de contenus dans ces flux.

L'un des modèles de la littérature les plus intéressants dans ce cadre est celui proposé dans [Ghalebi et al., 2018], qui modélise des cascades de diffusion selon des temps d'infection arrivant en flux : à un instant  $t$  du processus, on possède un ensemble d'épisodes  $\mathcal{D}_t$ , possiblement incomplets, collectés depuis le début du processus. L'idée de [Ghalebi et al., 2018] est de se baser sur le modèle génératif de réseaux de [Williamson, 2016] pour capturer les distributions postérieures des différentes relations de diffusion du réseau, qui peuvent être mises à jour incrémentalement au fur et à mesure que de nouvelles observations arrivent. Le modèle de [Williamson, 2016] sur lequel [Ghalebi et al., 2018] se base

est un modèle non-paramétrique de mixture de processus de Dirichlet, permettant de générer des réseaux avec propriétés observables dans la plupart des graphes du Web (i.e., fort degré de clustering, distribution des degrés des noeuds en loi de puissance), sous la forme de séquences échangeables de liens du réseau. Les diffusions n'étant cependant observées que sous la forme de séries de temps d'infection (i.e., les liens activés sont inconnus), [Ghalebi et al., 2018] intègre un processus supplémentaire d'échantillonnage de liens à partir de ces observations, avant d'appliquer le processus d'inférence de probabilités de postérieures selon le modèle de [Williamson, 2016]. L'approche démontre de bonnes capacités de modélisation de la diffusion, y compris dans un contexte dynamique avec structure de réseau pouvant évoluer. Néanmoins, sa complexité est fortement croissante avec le temps, car il nécessite de stocker tous les temps d'infection observés depuis le début du processus et de reconsidérer les relations associées à chaque étape d'inférence du modèle, ce qui limite grandement son applicabilité dans un contexte en ligne. De la même manière qu'on l'a fait en section 2.2.3.1 pour nos modèles de collecte dynamiques, il serait utile de limiter ces inférences coûteuse à une fenêtre d'historique, en se basant sur des statistiques suffisantes du passé, agissant sous la forme de prior. Bien sûr cela reviendrait à une approximation du modèle car ces priors sont basés sur des estimations ignorant les observations qui suivent leur sortie de la fenêtre d'historique, mais il serait possible de pondérer leur contribution en fonction de l'incertitude du modèle par rapport à leur validité. Par ailleurs, le fait d'"oublier" des observations trop lointaines rajouterait à l'aspect dynamique du modèle. Là encore l'usage de représentations distribuées des noeuds pourrait être bénéfique pour alléger le modèle et gagner en généralisation.

Un autre aspect limitant du modèle de [Ghalebi et al., 2018] est qu'il ne considère pas l'absence possible d'observation de certains temps d'infection dans les cascades, et infère les relations impliquées dans une cascade donnée selon l'arbre couvrant passant par les noeuds infectés observés uniquement. Il serait utile d'étendre le modèle aux données avec infections manquantes comme discuté en section 3.4.1.2, qui est un cadre plus réaliste quand on s'intéresse aux données issus de flux fournis par les réseaux sociaux (voir chapitre 2). L'approche de [Taxidou and Fischer, 2014a] considère ce cas mais dans un cadre très heuristique, qui gère le problème d'assignation de crédit d'infection selon des règles arbitraires simplistes (e.g., parmi les individus qu'un noeud infecté suit : dernier ou premier à avoir reposté le contenu, ou encore l'influenceur le plus suivi) pour déterminer les chemins de diffusion du contenu. Il conviendrait de reconsidérer cela dans un cadre probabiliste.

Enfin bien sûr un objectif plus long terme est d'inclure ce travail dans le cadre des approches du chapitre 2, pour la sélection active des flux de données, en fonction des utilisateurs à suivre pour améliorer le modèle. L'approche en ligne de [Taxidou and Fischer, 2014a] se base sur des threads de messages, construits à partir de retweets uniquement, en se concentrant sur des contenus semblant avoir une forte popularité selon l'API *Sample Streaming* de Twitter (voir section 2.1.1). Pour ces contenus identifiés comme objets de diffusion, la méthode suit leur propagation dans le réseau en traquant leurs mots-clé par l'API *Track Streaming*, afin de modéliser les relations de diffusion en jeu. Du fait des limites associées à une sélection de données selon cette API par mots-clé, déjà discutées en section 2.1.1, et tout particulièrement pour le cas d'une collecte à des fins de modélisation de relations entre utilisateurs, il paraît plus pertinent de travailler par sélection d'utilisateurs à suivre selon une API telle que l'API *Follow Streaming* de Twitter. Cela peut permettre au modèle de sélectionner les utilisateurs pour lesquels il est nécessaire de collecter de l'information dans un processus d'apprentissage actif.

Cependant la mise en place de stratégies efficaces dans ce cadre constitue un réel challenge. Deux possibilités sont envisageables dans ce contexte. La première, indirecte, est la considération dans des approches de bandits combinatoires de fonctions de récompenses statiques qui correspondent à des objectifs annexes, menant de manière secondaire à l'établissement d'un modèle de diffusion comme

c'est le cas pour les problématiques de maximisation d'influence discutées à la section précédente. Par exemple, on pourrait chercher à maximiser le volume de diffusions observées sur des ensembles d'utilisateurs suivis à chaque temps  $t$  du processus. Une seconde voie serait de considérer des modèles de récompenses dynamiques, comme déjà envisagé en section 2.3.3 dans un cadre plus général, où la fonction de récompense prend en compte les connaissances et incertitudes du modèle en fonction des données déjà collectées, pour sélectionner les ensembles d'utilisateurs pour lesquels on a besoin de plus d'informations sur leurs dynamiques de transmission de contenu. L'idée est d'éviter de re-sélectionner à chaque étape des ensembles d'utilisateurs dont les dynamiques de diffusion sont parfaitement connues. Bien entendu il faudra être vigilant au passage à l'échelle des méthodes considérées, et restreindre les graphes de diffusion à apprendre à des sous-ensembles d'utilisateurs cibles, pouvant être par exemple pré-sélectionnés par des méthodes plus légères telles que celles de la section 2.1.2 [Gisselbrecht et al., 2015c] pour cibler des communautés particulières par exemple, ou que celle de [Carpentier and Valko, 2016] si l'on souhaite se focaliser sur des leaders d'opinion.



## Chapitre 4

# Évolution temporelle dans les communautés d’auteurs

### Sommaire

---

<b>4.1</b>	<b>Modélisation de l’évolution langagière au cours du temps . . . . .</b>	<b>104</b>
<b>4.2</b>	<b>Modèle de Langue Dynamique Récurrent . . . . .</b>	<b>106</b>
4.2.1	Modèle à États Temporels pour la Génération de Documents Textuels . . . . .	107
4.2.2	Apprentissage par Inférence Variationnelle Temporelle . . . . .	108
<b>4.3</b>	<b>Apprentissage de représentations dynamiques dans les communautés d’auteurs . .</b>	<b>111</b>
4.3.1	Trajectoires Individuelles Déterministes . . . . .	112
4.3.2	Trajectoires Individuelles Stochastiques . . . . .	114
<b>4.4</b>	<b>Conclusions et Perspectives . . . . .</b>	<b>117</b>
4.4.1	Structuration de l’espace latent . . . . .	118
4.4.1.1	Auto-encodage textuel variationnel . . . . .	118
4.4.1.2	Biais d’exposition en prédiction . . . . .	121
4.4.1.3	Inférence de relations . . . . .	121
4.4.2	Prédiction stochastique inductive . . . . .	122
4.4.3	Modèles en ligne . . . . .	124

---

Dans le chapitre précédent nous nous sommes intéressés à la modélisation de phénomènes de diffusion à événements binaires ponctuels : des utilisateurs du réseau participent à des processus de diffusion à des instants identifiés, de manière abrupte, par exemple par re-partage de contenus. Lorsqu'ils participent à une diffusion, on considère que les utilisateurs sont entièrement et définitivement infectés pour le restant de l'épisode de diffusion. On suppose également généralement que des épisodes de diffusion sont directement identifiables à partir des données. C'est dans ce contexte que la plupart des travaux de relations d'influence entre auteurs de contenu ont été établis. Or, l'influence peut largement dépasser ce cadre simpliste, menant à des modifications plus diffuses des comportements des utilisateurs en relation, telles que des adoptions de tendances langagières, sans que l'on puisse nécessairement extraire ce genre de séquences d'événements discrets.

Dans ce chapitre nous nous intéressons à des évolutions temporelles de la langue dans des communautés d'auteurs. La modélisation du langage est un domaine très actif en apprentissage statistique, avec de très nombreuses applications. Classiquement, l'objectif est de déterminer des distributions de mots selon leur contexte, correspondant généralement aux mots de positions proches dans les textes considérés. Néanmoins, sur les réseaux du Web les textes sont souvent associés à des informations additionnelles, telles que leur auteur ou leur date de publication, qui peuvent être exploités pour améliorer les performances des modèles, en considérant les dynamiques structuro-temporelles des textes publiés. La langue des auteurs évolue au cours du temps, en fonction des sujets d'actualité du moment ou d'influences intra ou inter-communautés. Les textes publiés sont alors sujets à des dérives temporelles reflétant ces évolutions : le sens des mots peut changer, certains mots nouveaux peuvent apparaître alors que d'autres disparaissent, les dépendances syntaxiques et sémantiques évoluer.

Ce chapitre reprend certaines contributions établies sur ce thème au cours de la thèse d'Edouard Delasalles. La section 4.1 présente un survol des travaux existants pour la modélisation de la langue avec évolutions temporelles. La section 4.2 présente ensuite une première contribution pour cette problématique, où nous nous intéressons aux évolutions globales de la langue dans des communautés d'auteurs, afin de capturer la dérive temporelle générale de la langue sur ces communautés. Nous définissons enfin en section 4.3 un modèle pour la prise en compte de dérives individualisées, pouvant capturer certaines dynamiques d'influence diffuses du réseau. Ces deux approches, ouvrant sur des perspectives de recherche décrites en section 4.4, sont appliquées à des tâches de modélisation pure, de complétion de données ou encore de prédiction temporelle.

## 4.1 Modélisation de l'évolution langagière au cours du temps

Alors que le domaine du Traitement Automatique du Langage naturel (TAL) se place à un niveau d'analyse fine des relations de dépendances syntaxiques des textes, les modèles de langue statistiques sont essentiellement basés sur des comptages de mots (ou n-grammes), avec prise en compte de dépendances plus ou moins long terme. Depuis les premiers travaux autour des modèles unigrammes multinomiaux [Song and Croft, 1999], le domaine a largement évolué avec l'avènement des méthodes neuronales et des approches type *Word2Vec* avec représentations distribuées des mots [Bengio et al., 2003, Mikolov et al., 2013]. La recherche sur les modèles de langue par apprentissage profond est très active [Vaswani et al., 2017, Merity et al., 2018b, Bai et al., 2018, Melis et al., 2018, Merity et al., 2018a], avec des applications sur des tâches telles que la reconnaissance de la parole [Chiu et al., 2018], le sous-titrage d'image [Vinyals et al., 2017], la génération de textes [Fedus et al., 2018] ou la traduction automatique [Lample et al., 2018]. Récemment, la modélisation statistique de la langue a fortement

gagné en intérêt pour la communauté du TAL, avec la mise à disposition de réseaux pré-entraînés pour de multiples tâches [Devlin et al., 2019, Howard and Ruder, 2018, Peters et al., 2018], qui peuvent être utilisés efficacement comme brique de divers modèles sans ré-entraînement coûteux nécessaire.

Au début des années 2000, quelques travaux ont commencé à s'intéresser à l'évolution langagière au cours du temps par des modèles statistiques de suivi des tendances thématiques dans les documents textuels. Notamment le travail de [Kabán and Girolami, 2002], basé sur un modèle de Markov caché (HMM), a pour objet de visualiser les évolutions temporelles thématiques dans des flux de données textuelles. Cette approche, qui entre dans le domaine du *Topic Detection and Tracking* déjà discuté en section 2.1.1, étend les travaux de cartographie topographique temporelle dans les textes de [Bishop et al., 1997], et permet de visualiser les changements thématiques via des trajectoires sur une grille 2D. Cependant, ce genre d'approche ne permet pas l'établissement de modèles génératifs langagiers, avec prises en compte des tendances de la communauté étudiée. L'approche non-markovienne dans [Wang and McCallum, 2006], restreinte à des représentations en sacs de mots n'a pas non plus de capacité de génération langagière, mais s'avère performante pour détecter les évolutions thématiques au cours du temps. Divers autres travaux ont étudié l'évolution du vocabulaire au fil du temps, selon par exemple de transformations de graphes sémantiques dans [Kenter et al., 2015], ou les dérives thématiques dans des communautés d'auteurs, selon des modélisations des sujets dominants par période de temps [Hall et al., 2008]. On note également les travaux sur l'émergence du langage dans les communautés, montrant par exemple comment la compositionnalité émerge dans le langage par des modèles d'apprentissage itérés [Griffiths and Kalish, 2007, Ren et al., 2020], mais qui sortent largement du cadre de ce chapitre, n'ayant pas d'objectif d'apprentissage à partir de données réelles.

Plus proche de notre objectif de modélisation générative langagière, le modèle thématique dynamique de [Blei and Lafferty, 2006] propose une modélisation de type LDA avec mélanges de distributions de mots selon des thématiques [Blei et al., 2003], où les distributions de thématiques et les distributions de mots dans ces thématiques sont amenées à évoluer au cours du temps. Les transformations entre distributions multinomiales successives sont dirigées selon des mouvements browniens de leurs paramètres naturels, à la manière des filtres de Kalman, et optimisées par inférence variationnelle. Cependant, cette approche requiert de déterminer manuellement le nombre de thématiques à modéliser et la modélisation de la langue est limitée à des simples distributions d'occurrence de mots, sans prise en compte de dépendances entre mots générés. Par ailleurs, ce genre de modèle est contraint par l'utilisation de distributions conjuguées spécifiques pour l'inférence de leurs variables cachées modélisant l'évolution temporelle. On note des extensions du modèle de [Blei and Lafferty, 2006] pour une version temporelle multi-niveaux [Iwata et al., 2012] ou pour une version en temps continu [Wang et al., 2012a]. Dans une autre veine, [Gerrish and Blei, 2010] introduit le concept d'influence entre documents, qui pourrait sembler plus proche de nos objectifs mais est limité à des tâches d'analyse. Enfin, [Wang et al., 2011] propose une approche temporelle qui considère des relations entre documents via un graphe connu de dépendances, ce qui sort quelque peu du cadre de ce manuscrit où l'on suppose qu'aucune donnée structurelle n'est connue a priori.

Après l'introduction du modèle Word2Vec [Mikolov et al., 2013], de nombreux travaux ont proposé des dérivations de l'algorithme Skip-gram pour textes avec annotations temporelles [Frermann and Lapata, 2016]. Toutes ces approches visent à acquérir une meilleure compréhension des évolutions de la langue en étudiant des dérives temporelles des sémantiques des mots au cours du temps, via des dérives temporelles de leurs représentations distribuées. Parmi elles, [Eger and Mehler, 2016] apprend des dépendances linéaires entre représentations de mots de périodes de temps successives. [Yao et al., 2018] apprend des représentations de mots diachroniques par factorisation matricielle

avec contraintes d'alignement temporel. [Bamler and Mandt, 2017] propose un modèle Skip-gram temporel probabiliste, avec prior de diffusion dans l'espace de représentation. Dans la même veine, [Rudolph and Blei, 2017] propose une architecture utilisant des plongements stochastiques des mots, avec contraintes probabilistes entre pas de temps successifs. Comparées aux approches basées sur des HMM ou des représentations LDA, les approches de type Skip-gram utilisent des techniques d'optimisation par descente de gradient stochastique standard, qui peuvent être aisément parallélisées pour passer à l'échelle sur des corpus de données massifs. Le but de ces travaux est d'apprendre des représentations sémantiques des mots pouvant être directement utilisées dans divers modèles prédictifs. Les dépendances temporelles sont définies sur les représentations des mots : chaque période de temps considérée est associée à sa propre représentation du vocabulaire, respectant diverses contraintes de régularité temporelle.

Cependant, toutes ces approches basées sur des représentations évolutives des mots du vocabulaire présentent la limite majeure d'impliquer un apprentissage de  $|V|$  représentations par pas de temps, avec  $|V|$  la taille du vocabulaire. Cela conduit à des procédures d'apprentissage complexes, à la fois en temps et en mémoire, requérant de nombreuses approximations pour être utilisées sur des vocabulaires de taille raisonnable, telles que des optimisations alternées dans [Yao et al., 2018], ou des approximations des gradients dans [Bamler and Mandt, 2017]. Une exception notable est donnée dans [Rosenfeld and Erk, 2018], qui propose un réseau d'encodage qui produit une représentation temporelle d'un mot à partir d'une représentation statique de ce mot et d'un token représentant le temps de publication du texte considéré. Ce genre d'approche paraît néanmoins difficile dans un cadre multi-auteurs tel qu'on le considérera à la section 4.3, pour lequel des représentations différentes doivent être apprises pour chaque auteur aux différents pas de temps. On note l'approche de [Rudolph et al., 2017] pour données groupées, qui propose de réduire le nombre de paramètres en partageant des vecteurs de contexte entre groupes de textes, mais dont la transposition au cadre multi-auteurs paraît difficile (très grand nombre de groupes, doubles dépendances temporelles-structurelles). Toutes ces approches ne permettent en outre pas d'apprentissage de bout en bout de modèles de langue génératifs, et leur extension pour ce cadre, tel qu'on l'a considéré dans notre contribution [Delasalles et al., 2019a], ne permet pas de définir des distributions conditionnelles de mots réalistes en pratique, bien que très performantes pour des tâches annexes de classification par exemple.

## 4.2 Modèle de Langue Dynamique Récurrent

Une alternative aux différents modèles de la section précédente est de s'appuyer sur un conditionnement au temps de modèles de langue neuronaux, type RNNs. Les réseaux de neurones récurrents (présentés brièvement en section 3.3.1) sont très adaptés pour traiter des séquences de taille variable, ce qui en fait de très bons candidats pour la modélisation de la langue. Comparées au modèle Skip-gram qui utilise une fenêtre de contexte de taille limitée, les modèles de langue basés sur des réseaux récurrents travaillent avec des séquences de longueur arbitraire et sont capables de capturer des dépendances long terme dans les textes.

Les réseaux LSTM [Hochreiter and Schmidhuber, 1997, Mikolov et al., 2010] étaient jusqu'à récemment état de l'art pour la modélisation de la langue [Melis et al., 2018, Merity et al., 2018b]. Depuis, des architectures transformeur type BERT ou GPT-2 ont dépassé leurs performances par empilements de couches d'attention neuronales [Bai et al., 2018, Devlin et al., 2019, Vaswani et al., 2017, Radford et al., 2019]. Cependant, ces réseaux restent compétitifs, tout en étant généralement plus légers, pour la plupart des cas. Nous les considérons dans l'ensemble de ce chapitre, pour établir des modèles

dynamiques de la langue. Les méthodes présentées pourraient être étendues simplement en conservant les idées générales énoncées dans la suite, tout en substituant des architectures transformeur aux décodeurs LSTM utilisés dans nos contributions.

#### 4.2.1 Modèle à États Temporels pour la Génération de Documents Textuels

Dans ce qui suit, on considère des documents textuels définis sur un vocabulaire  $V$  et annotés selon des temps discrets successifs  $t \in \{1, 2, \dots, T\}$ . Soit  $X = (d^{(i)}, t^{(i)})_{i=1..N}$  un corpus de  $N$  documents associés à leur date de publication. Un document  $d^{(i)}$  est une séquence de mots de taille  $n^{(i)}$  de la forme  $d^{(i)} = \{w_1^{(i)}, w_2^{(i)}, \dots, w_{n^{(i)}}^{(i)}\}$ . Dans les tâches de modélisation de la langue standard, l'objectif est de trouver l'ensemble de paramètres  $\theta^*$  qui maximise la vraisemblance des mots selon leurs prédécesseurs dans tous les documents du corpus selon le modèle considéré :

$$\theta^* = \arg \max_{\theta} \prod_{i=1}^N \prod_{k=0}^{n^{(i)}-1} P_{\theta}(w_{k+1}^{(i)} | w_{0:k}^{(i)}) \quad (4.1)$$

où  $w_0^{(i)}$  et  $w_{n^{(i)}}^{(i)}$  sont respectivement les tokens de début et de fin de document, identiques dans chaque document.  $w_{0:k}^{(i)}$  est la séquence des premiers  $k+1$  tokens du document  $d^{(i)}$ , et  $P_{\theta}$  est modélisé par un RNN de paramètres  $\theta$  qui produit la distribution de probabilités catégorielle du prochain token. Plus spécifiquement, on a  $P_{\theta}(w_{k+1}^{(i)} | w_{0:k}^{(i)})$  définie selon  $\text{softmax}(Wh_k^{(i)} + b)$  où  $W \in \mathbb{R}^{V \times d_h}$  et  $b \in \mathbb{R}^V$  sont des paramètres de décodage à apprendre,  $h_k^{(i)} = f(w_k^{(i)}, h_{k-1}^{(i)}; \nu)$  est un vecteur caché de taille  $d^h$  pour tout  $k \in \{1, \dots, n^{(i)} - 1\}$  ( $h_0^{(i)}$  étant un vecteur nul pour tout  $i$ ), et  $f$  est la fonction récurrente d'un LSTM de paramètres  $\nu$ . On a alors  $\theta = \{U, W, b, \nu\}$  où  $U$  est la matrice de représentations des mots (ou tokens).

Le modèle général proposé dans cette section est un modèle à état (*State Space Model* - SSM) qui repose sur une variable globale dont les mouvements au cours des pas de temps successifs permettent de conditionner le modèle de langue LSTM. Notons que ce genre de conditionnement de RNNs par des variables aléatoires a déjà été considéré avec succès dans [Le and Mikolov, 2014, Serban et al., 2017a, Zaheer et al., 2017] pour des contextes différents. Contrairement aux approches existantes qui apprennent des matrices de représentations de mots complètes à chaque pas de temps, le modèle à état que l'on propose dans [Delasalles et al., 2019a] n'implique l'apprentissage que d'une seule représentation par mot, augmentée par l'état du SMM. Le modèle LSTM de décodage capture les dynamiques de langage générales, et utilise les états temporels pour adapter ses dynamiques dépendant de biais spécifiques à chaque pas de temps. On apprend également une fonction de transition entre états successifs, afin de permettre l'utilisation du modèle à des fins de prédiction future. Une représentation schématique du modèle est donnée en figure 4.1.

Soit  $z_t \in \mathbb{R}^{d_z}$  la variable latente correspondant à l'instant  $t$ . La probabilité d'un document  $d^{(i)}$  publié à l'instant  $t^{(i)}$  est alors obtenue selon :

$$P_{\theta}(d^{(i)} | t^{(i)}) = P_{\theta}(d^{(i)} | z_{t^{(i)}}) = \prod_{k=0}^{n^{(i)}-1} P_{\theta}(w_{k+1}^{(i)} | w_{0:k}^{(i)}, z_{t^{(i)}}).$$

Notons que  $z_{t^{(i)}}$  dépend seulement de l'instant auquel  $d^{(i)}$  a été publié, et n'est pas spécifique à  $d^{(i)}$ . Diverses possibilités sont envisageables pour conditionner la probabilité du document  $d^{(i)}$  selon  $z_{t^{(i)}}$ . Dans l'architecture considérée, on concatène  $z_{t^{(i)}}$  aux représentations de chaque mot  $w_k^{(i)}$ , ce qui nous a permis d'obtenir les meilleurs résultats expérimentaux.

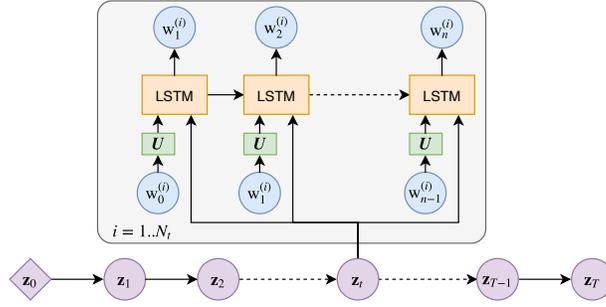


FIGURE 4.1 – Représentation schématique du modèle de langage dynamique récurrent.

Les états  $z_t$  considérés sont des variables aléatoires gaussiennes structurées dans le temps via une fonction de transition  $g$  de l'état précédent :

$$z_{t+1}|z_t \sim \mathcal{N}(g(z_t; \omega), \sigma^2 \mathbf{I}),$$

où  $\omega$  est le vecteur de paramètres de la fonction de transition  $g$  et où  $\sigma^2 \mathbf{I}$  correspond à une matrice de covariance diagonale apprise par le modèle. L'apprentissage de la fonction  $g$  donne au système plus de liberté pour définir des trajectoires temporelles utiles pour les états cachés. De plus, cela donne la possibilité d'estimer des états futurs du système, pour des instants pour lesquels aucune donnée n'est observable pendant l'entraînement. Le prior de moyenne sur le premier pas de temps est un vecteur appris  $z^0$  qui agit en tant que condition initiale du système.

La distribution jointe du modèle se factorise selon :

$$P_{\theta, \psi}(X, Z) = \prod_{i=1}^N P_{\theta}(d^{(i)} | z_{t^{(i)}}) \prod_{t=0}^{T-1} p_{\psi}(z_{t+1} | z_t) \quad (4.2)$$

où  $\psi = (\omega, \sigma^2, z_0)$  sont des paramètres de priors temporels, et  $Z \in \mathbb{R}^{T \times d_z}$  est la matrice contenant les vecteurs latents  $z_t$ .

#### 4.2.2 Apprentissage par Inférence Variationnelle Temporelle

Apprendre le modèle génératif donné par (4.2) requiert l'inférence des variables  $z^t$ , pas observables dans les données. Selon les principes d'inférence bayésienne, cela peut se faire en estimant la distribution postérieure  $p_{\theta, \psi}(Z|X) = \frac{P_{\theta, \psi}(X, Z)}{\int P_{\theta, \psi}(X, Z) dZ}$ . Malheureusement, la marginalisation sur  $Z$  requiert l'estimation d'une intégrale de normalisation incalculable. On a alors encore une fois recours à l'inférence variationnelle pour optimiser notre problème d'apprentissage. Dans ce qui suit, on considère une distribution variationnelle  $q_{\phi}(Z)$  qui se factorise en composantes indépendantes sur l'ensemble des pas de temps :

$$q_{\phi}(Z) = \prod_{t=1}^T q_{\phi}^t(z_t),$$

où les différents facteurs  $q_{\phi}^t$  sont des distributions gaussiennes indépendantes  $\mathcal{N}(\mu_t, \sigma_t^2 \mathbf{I})$ , de matrices de covariance diagonales  $\sigma_t^2 \mathbf{I}$ .  $\phi = ((\mu_t, \sigma_t))_{t \in \{1, \dots, T\}}$  correspond à l'ensemble des paramètres variationnels du modèle.

Cette factorisation est possible car la modélisation récurrente de la langue est une tâche auto-régressive (c.f. (4.1)) qui ne requiert pas l'auto-encodage des textes considérés et qu'une seule variable  $z_t$  est partagée par l'ensemble des documents publiés au temps  $t$ . On est alors capable d'apprendre un modèle avec relativement peu de paramètres, tout en évitant les difficultés liées à l'auto-encodage variationnel bayésien pour les données textuelles (e.g., KL vanishing) [Bowman et al., 2016]. Nous rediscutons de ce point en section 4.4.1. Une particularité de notre approche est que nous disposons donc de multiples documents pour chaque état du système. Nous adaptons la méthode de [Krishnan et al., 2017] à ce cas pour former une ELBO  $\mathcal{L}(\theta, \psi, \phi)$  à maximiser :

$$\begin{aligned}
 \log P_{\theta, \phi}(X) &= \log \int_Z P_{\psi}(Z) \prod_{t=1}^T P_{\theta}(X_t | z_t) dZ \geq \int_Z q_{\phi}(Z) \log \left( p_{\psi}(Z) \frac{\prod_{t=1}^T P_{\theta}(X_t | z_t)}{q_{\phi}(Z)} \right) dZ \\
 &= \sum_{t=1}^T \int_{z_t} q_{\phi}^t(z_t) \log P_{\theta}(X_t | z_t) dz_t + \int_{z_{t-1}} q_{\phi}^{t-1}(z_{t-1}) \log \frac{p_{\psi}(z_t | z_{t-1})}{q_{\phi}^t(z_t)} dz_{t-1} dz_t \\
 &= \sum_{t=1}^T \mathbb{E}_{q_{\phi}^t(z_t)} [\log P_{\theta}(X_t | z_t)] - \mathbb{E}_{q_{\phi}^{t-1}(z_{t-1})} [\text{KL}(q_{\phi}^t(\cdot) || p_{\psi}(\cdot | z_{t-1}))], \tag{4.3}
 \end{aligned}$$

où  $X_t$  est l'ensemble des documents publiés au temps  $t$ , et où l'inégalité est obtenue en appliquant le théorème de Jensen sur les fonctions concaves.

Cette borne inférieure de l'évidence exhibe deux termes d'espérance. Le premier correspond à la log-vraisemblance conditionnelle des observations à  $t$  selon l'état  $z_t$ . Le second correspond à l'espérance de divergence de Kullback-Leibler de la distribution variationnelle selon un prior gaussien dépendant de l'état précédent  $z_{t-1}$ . Notons que cette KL, entre deux gaussiennes, possède une forme close analytique (que l'on donne en annexe de [Delasalles et al., 2019a]). Cette ELBO peut être classiquement optimisée par montée de gradient stochastique, en utilisant l'astuce de reparamétrisation [Kingma and Welling, 2013, Rezende et al., 2014], déjà discutée en section 2.3.2.

Les états temporels globaux considérés dans cette approche, couplés à des distributions variationnelles indépendantes à travers le temps, offrent un certain nombre d'avantages pour l'apprentissage de dynamiques dans les textes d'une communauté d'auteurs. Cela permet notamment au système de faire face aux disruptions importantes des dérives du langage, pour lesquelles les régularités de transitions observées entre les autres pas de temps ne peuvent pas s'appliquer. Plutôt que de déstabiliser considérablement la fonction de transition, et par là impacter fortement les états consécutifs à la disruption, l'algorithme d'apprentissage peut choisir d'ignorer ces transitions difficiles, à un prix dépendant de la variance  $\sigma^2$ . Cette variance  $\sigma^2$ , apprise conjointement avec le modèle, permet à l'algorithme de s'adapter aux transitions stochastiques selon le niveau de régularité présent dans les données. De plus, grâce au découpage des dépendances temporelles, l'optimisation sur l'ensemble des pas de temps peut se faire de manière parallèle selon des mini-batches contenant uniquement des textes publiés au temps  $t$  (cela entraîne néanmoins quelques difficultés de convergence, reconsidérées en section 4.3.2).

**Résultats** Les tableaux 4.1 et 4.2 donnent des résultats en perplexité micro (i.e., perplexité sur l'ensemble des textes de test) et macro (i.e., moyenne des perplexités sur les différentes périodes de test), respectivement pour une tâche de prédiction et une tâche de modélisation :

- Prédiction : l'objectif est de prédire les évolutions futures. on se sert de toutes les données issues des premières périodes de temps des corpus pour l'entraînement (75% des périodes de temps

Model	S2		NYT		Reddit	
	<i>micro</i>	<i>macro</i>	<i>micro</i>	<i>macro</i>	<i>micro</i>	<i>macro</i>
LSTM	84.7	82.7	128.5	128.4	125.8	126.1
DT	92.0	89.6	137.1	137.0	151.1	151.6
DWE	87.0	84.8	140.1	140.0	136.5	139.9
DRLM-Id	81.2	79.2	123.7	123.6	124.7	125.0
DRLM	<b>79.7</b>	<b>77.8</b>	<b>123.3</b>	<b>123.1</b>	<b>123.9</b>	<b>124.3</b>

TABLEAU 4.1 – Perplexité en Prédiction

Model	S2		NYT		Reddit	
	<i>micro</i>	<i>macro</i>	<i>micro</i>	<i>macro</i>	<i>micro</i>	<i>macro</i>
LSTM	62.8	66.2	109.9	110.4	116.7	123.0
DT	70.7	73.9	125.6	120.4	136.8	147.7
DWE	65.9	69.8	119.9	120.4	129.4	139.6
DRLM-Id	60.6	<b>61.3</b>	104.0	104.4	115.5	121.5
DRLM	<b>60.2</b>	<b>61.2</b>	<b>103.5</b>	<b>103.9</b>	<b>114.7</b>	<b>120.4</b>

TABLEAU 4.2 – Perplexité en Modélisation

environ) et on reporte les résultats pour les périodes de temps suivantes, de  $T + 2$  à  $T + 8$  environ (pour lesquelles aucun texte n'a été observé en apprentissage),  $T + 1$  étant réservé à la validation ;

- Modélisation : l'objectif est de modéliser au mieux les distributions de texte sur l'ensemble des pas de temps. Toutes les périodes de temps sont utilisées en entraînement, mais sur seulement 60% des données, les données restantes servant pour 10% à la validation et à 30% pour le test.

On considère les jeux de données suivants :

- Le corpus **Semantic Scholar**<sup>1</sup> corpus (S2) contient 50K titres publiés dans des journaux et conférences en apprentissage statistique entre 1985 et 2017, découpés en années (33 périodes).
- Le corpus **New York Times** [Yao et al., 2018] (NYT) contient 50K titres d'articles du journal (500K mots) publiés entre 1990 et 2015, découpés en années (26 périodes).
- Le corpus **Reddit** [Tan and Lee, 2015] regroupe un échantillon correspondant à 3% des messages publiés sur le réseau social *Reddit*. Il est composé de 100K messages publiés entre Janvier 2006 et Décembre 2013, découpés en quarts d'année (32 saisons).

On compare notre modèle DRLM à un LSTM classique qui n'intègre pas la période de publication des textes dans son modèle, ainsi qu'à une version de notre modèle où le réseau de neurones de la fonction de transition  $g$  est remplacé par la fonction identité, afin d'évaluer l'apport de cette fonction de transition. On se compare également à des adaptations pour la modélisation de textes des approches DT [Rosenfeld and Erk, 2018] et DWE [Bamler and Mandt, 2017] discutées en section 4.1. Toutes les approches sont basées sur le LSTM à 2 niveaux AWD-LSTM de [Merity et al., 2018b], avec espaces de représentation de dimension 400. On utilise des techniques de *weight dropout*, *embedding dropout*, *variational dropout* [Gal and Ghahramani, 2016] et *embedding weight-tying* [Inan et al., 2017], pour régulariser les modèles.

Notons qu'en test on utilise directement la moyenne des distributions variationnelles des états à chaque pas de temps  $z_t = \mu_t$ , plutôt que d'échantillonner l'état selon la distribution, afin de réduire la variance des résultats. Pour la tâche de prédiction, il faut produire des états pour les temps futurs. Pour notre modèle DRLM, on utilise la fonction de transition  $g$  que l'on applique à partir de l'état moyen du dernier temps d'apprentissage  $T$ . Pour le modèle DRLM-id on a  $z_{T+k} = \mu_T$  pour tout  $k > 0$ . Pour DT et DWE on utilise les représentations de mots apprises pour la période  $T$  pour toute période de test  $T + k$ .

Les résultats montrent que la prise en compte de la dimension temporelle est bénéfique pour les deux tâches et pour les trois corpus, il existe bien une dérive temporelle qu'il est possible d'exploiter. Les résultats confirment aussi la difficulté d'adapter les approches basées sur *Skip-gram* pour produire des modèles génératifs de langue. Pour la prédiction, l'apprentissage d'une fonction de transition paraît bénéfique, avec une amélioration significative des performances sur les corpus S2 et Reddit. En modélisation cette fonction est principalement bénéfique sur le corpus Reddit, où elle permet

1. <http://labs.semanticscholar.org/corpus/>

Année	a framework for...	unsupervised learning of...	a comparison of...
1985	shape recovery from images	hidden markov models	smoothing techniques for statistical machine translation
1995	shape recovery from images	gaussian graphical models	smoothing techniques for word sense disambiguation
2005	automatic evaluation of statistical machine translation	named entity recognizers	smoothing techniques for statistical machine translation
2015	unsupervised feature selection	deep convolutional neural networks	convolutional neural networks for action recognition
2016	unsupervised learning of deep neural networks	convolutional neural networks	convolutional neural networks for action recognition
2017	training deep convolutional neural networks	generative adversarial networks	convolutional neural networks for action recognition

 TABLEAU 4.3 – Textes générés par *beam search* avec DRLM pour différentes années sur le corpus S2.

de structurer l'espace de manière beaucoup plus régulière qu'avec DRLM-Id, et gagner alors en généralisation. Des résultats supplémentaires donnés dans [Delasalles et al., 2019a] montrent en effet l'apport de la fonction de transition sur l'organisation de l'espace de représentation. On y observe également son intérêt pour la prédiction long terme, les régularités de dynamique qu'elle a pu capturer sur les données d'entraînement lui permettant de suivre certaines évolutions de la langue dans le futur. La table 4.3 donne des exemples de titres générés par notre modèle DRLM pour différentes années sur le corpus S2, selon un processus de *beam search* et trois différents débuts de séquence. On observe pour les trois exemples une évolution du langage généré correspondant bien aux évolutions du domaine de l'apprentissage statistique, avec l'apparition sur les dernières années d'expressions comme *deep convolutional neural networks* ou *generative adversarial networks*.

#### Publications associées :

- Delasalles, E., Lamprier, S., and Denoyer, L. (2019a). Dynamic neural language models. In *Neural Information Processing - 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12-15, 2019, Proceedings, Part III*, pages 282–294

### 4.3 Apprentissage de représentations dynamiques dans les communautés d'auteurs

Pour des objectifs de prédiction d'auteurs, divers travaux se sont intéressés au style d'écriture des textes dans des méthodes d'apprentissage statistique [Ding et al., 2017], mais très peu se sont intéressés à leur considération dans un cadre temporel, pour capturer les dynamiques de diffusion de comportements langagiers dans les communautés d'auteurs. Dans cette section, on propose d'étendre le travail sur l'évolution langagière initiée à la section précédente, en l'adaptant au cas individuel, toujours par le biais d'un modèle à états, mais où chaque auteur possède son propre état de conditionnement à chaque étape, et donc sa propre trajectoire d'évolution. On commence par proposer une version déterministe des dynamiques du modèle en section 4.3.1, pour s'intéresser à une extension stochastique donnant plus de flexibilité en section 4.3.2.

### 4.3.1 Trajectoires Individuelles Déterministes

L'objectif de cette section est donc de produire un modèle permettant une individualisation du conditionnement d'un modèle de langue LSTM modélisant  $P_{\theta}(d^{(i)}|a^{(i)}, t^{(i)}) = P_{\theta}(d^{(i)}|z_{a^{(i)}, t^{(i)}}, z_a^{(i)})$ , où  $a^{(i)}$  correspond à l'auteur du texte  $d^{(i)}$ ,  $z_{a^{(i)}, t^{(i)}}$  le vecteur de conditionnement dynamique à  $t^{(i)}$  pour  $a^{(i)}$  et  $z_a^{(i)}$  son vecteur de conditionnement statique. L'ajout de ce vecteur statique permet au modèle d'états dynamiques, et donc à la fonction de transition, de se focaliser sur les dérives temporelles, plutôt que d'avoir à encoder également les spécificités statiques de chaque auteur. On montre dans [Delasalles et al., 2019b] que ce conditionnement statique permet en effet d'améliorer significativement la stabilité du modèle.

Toute la difficulté revient donc à apprendre des vecteurs de conditionnement dynamiques efficaces, capables de modéliser les trajectoires langagières de chaque individu dans un même espace de représentation. L'idée est de modéliser des tendances générales (e.g., dérive générale des thématiques de la communauté étudiée), via une fonction de transition commune à tous les auteurs, tout en permettant la capture de trajectoires individualisées, modélisant implicitement les jeux d'influence de la communauté. On propose de considérer l'architecture représentée en figure 4.2, où une fonction d'initialisation  $g_{\psi}$  utilise tout d'abord la représentation statique de l'auteur  $a$ ,  $z_a$ , pour produire le vecteur initial  $z_{a,0}$ . Pour obtenir le vecteur de conditionnement dynamique pour tout texte  $d = (w_1, \dots, w_{|d|})$  publié par un auteur  $a$  au temps  $t$ , on applique récursivement  $t$  fois une fonction de transition  $f_{\phi}$  à partir du vecteur initial  $z_{a,0}$ .

Diverses possibilités peuvent être envisagées pour  $f_{\phi}$ . Nous avons choisi dans [Delasalles et al., 2019b] d'utiliser une architecture résiduelle, avec fonction de transition markovienne, qui ne considère que le dernier état produit  $z_{a,t}$  pour induire le suivant  $z_{a,t+1}$ . Ce choix paraît former un bon compromis entre stabilité et robustesse selon nos expérimentations. L'utilisation de modèles séquentiels plus puissants, tels que des RNNs qui maintiennent une mémoire des états passés, se heurte en effet à des problèmes de sur-apprentissage. Les nombres d'auteurs et de pas de temps sont généralement petits comparés au nombre de documents dans les collections, et de nombreuses paires auteur-pas de temps sont manquantes dans les données. L'utilisation d'une fonction résiduelle permet l'apprentissage de trajectoires plus régulières, où la magnitude et la direction des résidus produits peuvent être aisément contrôlés en régularisant par exemple  $\phi$  selon sa norme L2. L'état dynamique pour un auteur  $a$  au temps  $t$  s'obtient alors selon  $z_{a,t-1}$  et  $z_a$  par :

$$z_{a,t} = z_{a,t-1} + f_{\phi}(z_{a,t-1}, z_a). \quad (4.4)$$

où  $f_{\phi}$  est un réseau de neurones multi-couches (MLP) avec activations ReLU. L'ajout de la représentation statique  $z_a$  de l'auteur  $a$  en entrée de  $f_{\phi}$  permet d'encourager différentes dynamiques pour les différents auteurs, tout en partageant un même espace et une même fonction de transition. Sans cela, deux états identiques pour deux auteurs différents à un temps  $t$  donné impliqueraient nécessairement deux trajectoires futures identiques. Une étude d'ablation dans [Delasalles et al., 2019b] montre que l'ajout du vecteur statique en entrée de  $f_{\phi}$  joue en effet un rôle important dans la capture de dynamiques individuelles efficaces.

**Résultats** La figure 4.3 donne les résultats en terme de gains par rapport à la baseline LSTM (pourcentages de diminution de la perplexité du modèle LSTM) pour les corpus S2 et NYT. On considère trois tâches : 1) modélisation, où le corpus d'entraînement contient des textes publiés à tous les temps, 2) complétion, où pour chaque auteur on réserve 20% des périodes de publication pour le test et 10%

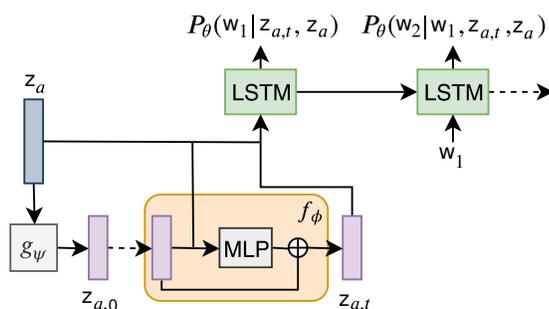


FIGURE 4.2 – Description de l’architecture du modèle dynamique individuel.

pour la validation, et 3) prédiction, où les 70% des documents de chaque auteur, pris par ordre chronologique de publication, sont utilisés en entraînement, les 10% suivants pour la validation et enfin les 20% restants pour le test. On considère une baseline LSTM-A conditionnée uniquement sur le vecteur statique  $z_a$  de chaque auteur et une autre LSTM-iAT conditionnée par des vecteurs  $z_a$  et  $z_{a,t}$ , avec tous  $z_{a,t}$  libres (i.e., où chaque  $z_{a,t}$  correspond à une variable apprise indépendante plutôt que résultant d’une fonction de transition comme dans le modèle proposé). Le modèle LSTM-AT suit le même principe que LSTM-iAT mais où l’on contraint les états successifs à être proches dans l’espace de représentation par une régularisation L2 de leurs déplacements. Pour ces deux modèles ne possédant pas de fonction de transition, pour tout temps  $t$  et tout auteur  $a$  pour lequel on ne dispose d’aucun exemple de publication à  $t$ , on utilise le vecteur de conditionnement  $z_{a,t'}$ , où  $t' < t$  correspond au temps passé le plus récent à  $t$  auquel l’auteur  $a$  a publié. La figure 4.3 ne rapporte pas les résultats pour LSTM-iAT, trop mauvais, car très fortement sujet au sur-apprentissage. Il faut contraindre les déplacements des états successifs sans quoi on perd toute cohérence temporelle et le modèle n’est pas capable de généraliser correctement, quelle que soit la tâche. Les résultats de la figure montrent néanmoins que lorsque ces vecteurs dynamiques sont correctement organisés dans le temps, la prise en compte de cet aspect temporel est bénéfique dans la plupart des cas, les courbes *Ours* et *LSTM-AT* se trouvant souvent significativement au dessus de la courbe *LSTM-A*. En outre, on observe l’apport de la fonction de transition résiduelle, permettant de gagner en prédiction et complétion sur les deux corpus, mais aussi en modélisation dans le cas du corpus S2. On note le gain long terme significatif en prédiction pour les deux corpus, particulièrement après la barre pointillée verticale noire symbolisant le temps après lequel aucun texte n’a été observé en apprentissage.

La figure 4.4 représente une visualisation par analyse en composantes principales des trajectoires apprises par notre modèle pour  $z_{a,t}$  pour tout  $a$  sur les corpus S2 et NYT. Alors qu’on observe des trajectoires rectilignes sur S2, ce qui dénote que le gain observé sur ce corpus par la prise en compte de l’aspect temporel est principalement dû à une dérive globale des thématiques de la communauté (i.e., vers le *deep learning* dans ce cas), les trajectoires sont bien plus individualisées dans le cas du corpus NYT. Il semble que ce sur ce corpus, les auteurs suivent beaucoup moins une tendance commune que sur S2, chaque auteur du journal possédant sa spécialité propre, diverses dynamiques émergent des données, correspondant à diverses sous-communautés à l’intérieur de la communauté globale considérée. Des résultats d’analyse complémentaires donnés dans [Delasalles et al., 2019b] montrent en effet différents groupes d’auteurs aux centres d’intérêts bien distincts sur ce corpus.

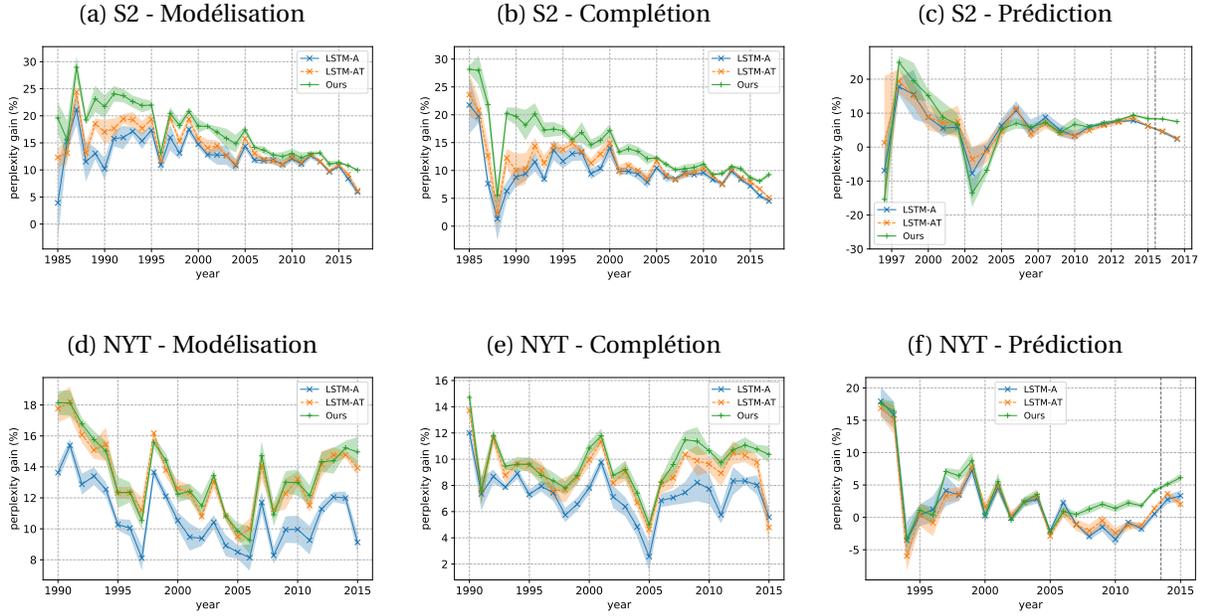


FIGURE 4.3 – Gains en perplexité par rapport à la baseline LSTM pour les corpus S2 et NYT.

### 4.3.2 Trajectoires Individuelles Stochastiques

Dans cette section, on cherche à donner un peu plus de souplesse au modèle, par l'utilisation d'états dynamiques stochastiques, comme effectué à la section 4.2.2, mais dans un cadre individualisé. Plutôt que de restreindre les états successifs  $z_{a,t}$  à suivre strictement des dynamiques définies de manière déterministe par la fonction de transition, on considère ici chaque  $z_{a,t}$  comme une variable aléatoire suivant une distribution gaussienne diagonale centrée sur  $\mu_{a,t} = z_{a,t-1} + f_{\phi}(z_{a,t-1}, z_a)$  :

$$z_{a,t} \sim \mathcal{N}(\mu_{a,t}, \sigma^2 \mathbf{I}) \quad (4.5)$$

où  $\mathbf{I}$  est la matrice identité et  $\sigma^2$  est un hyper-paramètre permettant de contrôler la tolérance à la déviation de la position de l'état par rapport à la moyenne  $\mu_{a,t}$  lors de l'apprentissage du modèle. Comme pour le modèle de la section 4.2.2, cette relaxation stochastique a pour objectif de permettre au modèle de faire face aux fortes disruptions de dynamiques pouvant survenir à certains pas de temps (e.g., correspondant à un évènement externe important). Si, par exemple, un auteur change de domaine de prédilection de manière abrupte à un pas de temps donné, cela pourrait induire de fortes instabilités dans l'apprentissage des dynamiques dans le cadre déterministe. L'utilisation d'états stochastiques répond à ce problème, en donnant plus de flexibilité en apprentissage, car permettant au modèle d'ignorer ce genre de transition difficile, à un prix dépendant de la variance  $\sigma^2$ .

Le modèle génératif décrit par (4.5) implique la maximisation de la log-vraisemblance suivante :

$$\begin{aligned} \log P_{\theta}(X) &= \log \int_{\mathcal{Z}} \prod_{i=1}^N P_{\theta}(d^{(i)} | z_{a^{(i)}, t^{(i)}}, z_{a^{(i)}}) \prod_{a \in \mathcal{A}} \prod_{t=1}^T p(z_{a,t} | \mu_{a,t}, \sigma^2) dZ \\ &= \sum_{a \in \mathcal{A}} \log \int_{\mathcal{Z}_a} \prod_{t=1}^T p(z_{a,t} | \mu_{a,t}, \sigma^2) \prod_{d \in X_{a,t}} P_{\theta}(d | z_{a,t}, z_a) dZ_a, \end{aligned} \quad (4.6)$$



FIGURE 4.4 – PCA des trajectoires  $z_{a,t}$  pour S2 et NYT. Les couleurs représentent le temps, avec un dégradé chronologique de violet foncé à jaune.

où  $Z$  correspond à l'ensemble des vecteurs  $z_{a,t}$  du modèle et  $Z_a$  à la séquence d'états  $(z_{a,t})_{t \in \{1 \dots T\}}$  pour l'auteur  $a$ . La seconde équation de (4.6) est obtenue grâce à l'indépendance mutuelle des séquences  $Z_a$  des différents auteurs  $a$  du modèle. Néanmoins, la log-vraisemblance à optimiser présente une marginalisation sur  $Z_a$  impossible à résoudre analytiquement. Une possibilité serait alors d'introduire une distribution variationnelle se factorisant selon un découpage par pas de temps comme à la section 4.2.2. Une autre possibilité serait d'utiliser la distribution variationnelle  $q_\Psi(Z)$  qui se factorise de la façon suivante :

$$q_\Psi(Z) = \prod_{a \in \mathcal{A}} \prod_{t=1}^T q_\Psi^{a,t}(z_{a,t} | z_{a,t-1}, z_a), \quad (4.7)$$

où tout  $q_\Psi^{a,t}(\cdot)$  correspond à un facteur individuel de la distribution, avec  $\Psi$  l'ensemble des paramètres variationnels du modèle. Suivant [Krishnan et al., 2017], même dans le cas de ce genre de distribution conditionnée selon l'état passé, il est possible d'obtenir une procédure d'apprentissage variationnel stable en se ramenant à des échantillonnages uniquement sur  $z_{a,t-1}$  pour chaque couple  $(a, t)$ . Cependant pour ces deux schémas d'inférence, nous avons rencontré des difficultés de convergence, que nous expliquons par le fait que la retro-propagation à travers le temps n'est pas assurée de bout en bout pour chaque exemple. L'optimisation est seulement réalisée par adaptation d'états successifs à chaque étape. La propagation de ces adaptations successives ne se fait alors que de proche en proche, ce qui rend la convergence lente et sujette à de nombreux optima locaux et points selle. Aussi, nous préférons nous référer à des approches, telles que [Chung et al., 2015] or [Fraccaro et al., 2016], qui considèrent chaque état comme une fonction déterministe de l'état précédent et d'une variable de bruit gaussien  $\epsilon_{a,t}$  :

$$z_{a,t} = \mu_{a,t} + \epsilon_{a,t}, \text{ with } \epsilon_{a,t} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}). \quad (4.8)$$

où  $\mu_{a,t}$  est toujours égal à  $z_{a,t-1} + f_\phi(z_{a,t-1}, z_a)$ . Ceci est tout à fait équivalent au fait de considérer le prior donné en (4.5), mais permet d'isoler la stochasticité sur des variables de bruit blanc lors de l'inférence des états. Plutôt que de considérer des distributions  $q_\Psi^{a,t}(\cdot)$  produisant une moyenne et une variance pour  $z_{a,t}$ , on considère des réseaux de neurones  $q_\Psi^{a,t}$  qui produisent la moyenne  $\tilde{\mu}_{a,t}$  et la variance  $\tilde{\sigma}_{a,t}^2$  de chaque bruit  $\epsilon_{a,t} = z_{a,t} - \mu_{a,t}$  selon les données. On a alors :

$$q_\Psi^{a,t}(z_{a,t} | z_{a,t-1}, z_a) = \mathcal{N}(z_{a,t}; \tilde{\mu}_{a,t} + \mu_{a,t}, \tilde{\sigma}_{a,t}^2 \mathbf{I}). \quad (4.9)$$

Cela implique d'échantillonner des trajectoires complètes à chaque étape d'optimisation, au prix d'un accroissement du temps d'apprentissage (environ le double de temps dans nos expérimentations),

mais permet d'obtenir de bien meilleurs résultats pour notre modèle avec trajectoires individualisées, grâce à un meilleur flot de retro-propagation du gradient à travers le temps.

Multiplier et diviser la partie droite de (4.6) par chaque facteur correspondant  $q_\Psi(Z_a)$ , et en utilisant l'inégalité de Jensen, on obtient l'ELBO  $\mathcal{L}(X; \Theta, \Psi)$  selon :

$$\begin{aligned} \log P_{\Theta, \Psi}(X) &\geq \sum_{a \in \mathcal{A}} \int_{Z_a} q_\Psi(Z_a) \left( \sum_{t=1}^T \log \frac{p_z(z_{a,t} | \mu_{a,t}, \sigma^2)}{q_\Psi^{a,t}(z_{a,t} | z_{a,t-1}, z_a)} + \sum_{d \in X_{a,t}} \log P_\theta(d | z_{a,t}, z_a) \right) dZ_a \\ &= \sum_{a \in \mathcal{A}} \sum_{t=1}^T \mathbb{E}_{q_\Psi(Z_{a,1:t-1})} \left[ \mathbb{E}_{q_\Psi^{a,t}(z_{a,t} | z_{a,t-1}, z_a)} \left[ \log P_\theta(d | z_{a,t}, z_a) \right] \right. \\ &\quad \left. - D_{\text{KL}}(q_\Psi^{a,t}(\cdot | z_{a,t-1}, z_a) \| p_z(\cdot | \mu_{a,t}, \sigma^2)) \right] \end{aligned} \quad (4.10)$$

où  $p_z(z_{a,t} | \mu_{a,t}, \sigma^2) = \mathcal{N}(z_{a,t}; \mu_{a,t}, \sigma^2 \mathbf{I})$  et  $Z_{a,1:t-1}$  correspond à la séquence des états précédant  $t$  pour l'auteur  $a$  :  $Z_{a,1:t-1} = (z_{a,t})_{t \in \{1 \dots t-1\}}$ . L'égalité dans (4.10) est obtenue en utilisant l'indépendance des probabilités de décodage par rapport aux états futurs.  $D_{\text{KL}}(q \| p)$  correspond à la divergence de Kullback-Leibler de  $q$  par rapport à  $p$ , qui possède une forme close pour  $q$  et  $p$  gaussiennes. Dans notre cas, on considère alors la formulation :

$$D_{\text{KL}}(q_\Psi^{a,t}(\cdot | z_{a,t-1}, z_a) \| p_z(\cdot | \mu_{a,t}, \sigma^2)) = \frac{1}{2} \left[ \log \frac{|\sigma^2 \mathbf{I}|}{|\tilde{\sigma}_{a,t}^2 \mathbf{I}|} + \text{tr}\{(\sigma^2 \mathbf{I})^{-1} (\tilde{\sigma}_{a,t}^2 \mathbf{I})\} + \tilde{\mu}_{a,t}^\top (\sigma^2 \mathbf{I})^{-1} \tilde{\mu}_{a,t} - d_z \right], \quad (4.11)$$

où  $d_z$  correspond à la taille de l'espace de représentation de  $z_{a,t}$ . Les paramètres variationnels  $\Psi = ((\tilde{\mu}_{a,t})_{a \in \mathcal{A}, t \in \{1 \dots T\}}, (\tilde{\sigma}_{a,t}^2)_{a \in \mathcal{A}, t \in \{1 \dots T\}})$  sont optimisés conjointement aux paramètres du modèle  $\theta$ , par la maximisation de  $\mathcal{L}(X; \Theta, \Psi)$  via montée de gradient stochastique en utilisant l'astuce de reparamétrisation pour faire de  $z_{a,t}$  une transformation déterministe d'un bruit standard gaussien, comme discuté à la section 2.3.2.

**Résultats** La figure 4.5 reporte des résultats en perplexité obtenus avec cette version stochastique du modèle. Elle donne l'évolution des résultats en fonction de la valeur de l'hyper-paramètre  $\sigma^2$  qui contrôle la tolérance à la déviation des positions des états par rapport à leur prior calculé en fonction de l'état précédent et de la fonction de transition.  $\sigma^2 = 0$  correspond à la version déterministe présentée en section précédente. Plus  $\sigma^2$  est élevé, plus le modèle a de liberté pour inférer  $\tilde{\mu}_{a,t}$ . Cependant, pour  $\sigma^2$  élevé,  $\tilde{\sigma}_{a,t}^2$  tend à être élevé également, ce qui perturbe l'apprentissage en y injectant trop de bruit. Lors de l'utilisation du modèle en test, on utilise le mode des distributions variationnelles pour déterminer les états plutôt que d'échantillonner divers  $z_{a,t}$ , afin de stabiliser les résultats. Les améliorations par rapport à la version déterministe ne sont pas gigantesques, mais il y a tout de même un gain significatif dans la plupart des cas pour  $\sigma^2 = 0.1$ . Pour des valeurs plus faibles, le modèle est quelque peu déstabilisé par des auteurs aux dynamiques présentant des ruptures temporelles. A contrario, pour des valeurs élevées (e.g.,  $\sigma^2 = 1$ ), on observe de catastrophiques baisses de performances, dues à la très forte variance durant l'apprentissage, empêchant une bonne convergence du modèle vers des tendances de dynamiques stables. Notons que des améliorations significatives ont été observées en contraignant la variance inférée  $\tilde{\sigma}_{a,t}^2$  (i.e., par une régularisation L2). Dans ce cas, il peut y avoir des tendances au sur-apprentissage cependant, particulièrement pour de fortes valeurs de  $\sigma^2$ , pour lesquelles le modèle a alors beaucoup de liberté pour expliquer les données d'entraînement. Ces résultats sont prometteurs, car ils valident la nécessité de relaxer le modèle de manière à ce qu'il puisse se focaliser sur les dynamiques principales du réseau, en ignorant les disruptions abruptes

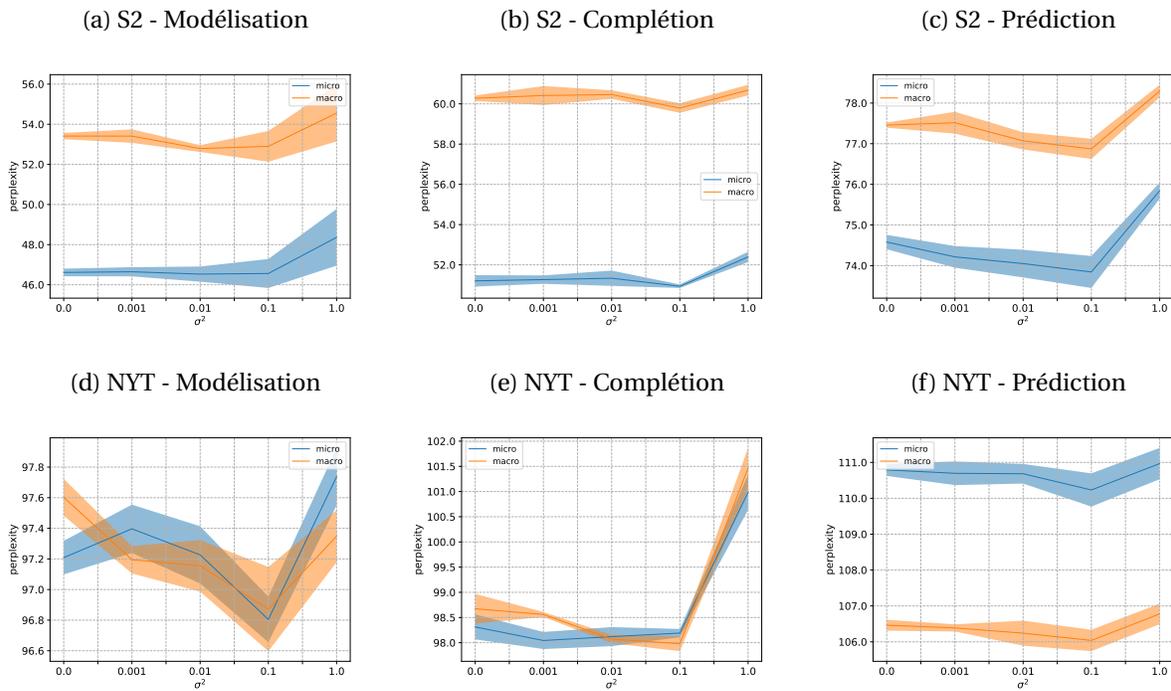


FIGURE 4.5 – Micro et Macro-perplexité selon la variance  $\sigma^2$  du prior des états, pour les corpus *Semantic Scholar* (en haut) et *New-York Times* (en bas).

dans les évolutions des auteurs, pour permettre l'apprentissage de trajectoires régulières des états de conditionnement. Diverses perspectives sont envisageables pour aller plus loin dans ce cadre, comme discuté en section 4.4.1.

**Publications associées :**

- Delasalles, E., Lamprier, S., and Denoyer, L. (2020). Deep dynamic neural networks for temporal language modeling in author communities. *Knowledge and Information Systems*, to appear
- Delasalles, E., Lamprier, S., and Denoyer, L. (2019b). Learning dynamic author representations with temporal language models. In *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*, pages 120–129
- Delasalles, E., Lamprier, S., and Denoyer, L. (2018). Apprentissage de l'évolution langagière dans des communautés d'auteurs. In *Conférence en Recherche d'Informations et Applications - CO-RIA 2018, 15th French Information Retrieval Conference, Rennes, France, May 16-18, 2018. Proceedings*

**4.4 Conclusions et Perspectives**

Ce chapitre a présenté un travail important sur la modélisation de l'évolution de la langue dans des communautés d'auteurs. Nous avons montré qu'il est possible d'extraire des trajectoires de représentations langagières dans un espace latent à partir de données textuelles, utiles pour des tâches

de modélisation et/ou prédiction de la langue dans la communauté étudiée. Ce travail a concerné un modèle d'évolution globale (section 4.2) et un modèle de dynamiques à l'échelle des individus (section 4.3), qui détermine des trajectoires individuelles dans un même espace pour les différents auteurs. Des résultats intéressants pour la modélisation de la langue, la complétion de données et la prédiction de publications futures ont été obtenus. Ce travail constitue un premier pas vers l'établissement de modèles d'extraction de dynamiques de l'information à partir de données complexes comme les publications datées d'une communauté d'auteurs, où l'on ne possède pas de marqueurs clairs de participation à des épisodes de diffusion identifiés comme c'était le cas au chapitre précédent.

Une première possibilité pour étendre ce travail serait d'imaginer un modèle à deux niveaux hiérarchiques, un niveau prenant en charge les dynamiques globales de la communauté et un autre se concentrant sur les dynamiques spécifiques à chaque auteur. Cela pourrait permettre de décharger la partie individuelle de la modélisation de la dérive globale de la communauté, qui empêche par exemple dans le corpus S2 d'observer une individualisation des trajectoires, car trop importante au regard des spécificités individuelles d'évolution. Une autre possibilité d'extension assez directe serait de s'intéresser à des données textuelles en temps continu (ou quasi-continu), où les mouvements browniens des états de conditionnement pourraient être modélisés à l'aide d'un processus de Wiener :

$$dz_{a,t} = \underbrace{f_{\phi}(z_{a,t})}_{\text{Dérive}} dt + \underbrace{\sigma(z_{a,t})}_{\text{Diffusion}} dW(t)$$

avec  $f_{\phi}(z_{a,t})$  la fonction de dérive temporelle de  $z_{a,t}$ , implémentée par un réseau de neurones comme dans nos approches des sections précédentes,  $\sigma(z_{a,t})$  contrôlant l'amplitude des écarts stochastiques, possiblement un réseau dépendant de l'état courant (mais peut aussi correspondre à un simple hyperparamètre constant comme dans nos approches à temps discrets) et  $(W(t))_{t>0}$  un processus de Wiener. Les nombreux travaux récents sur la modélisation par réseaux de neurones d'équations différentielles [Chen et al., 2018], et plus particulièrement d'équations différentielles stochastiques [Tzen and Raginsky, 2019, Deng et al., 2020], constitueront des sources d'inspiration importantes pour la mise en œuvre de cette extension à temps continu.

D'autres perspectives de travail, dont nous discutons en section 4.4.1, concernent une meilleure structuration de l'espace latent, de manière à être à même de véritablement extraire des relations d'influence à partir des trajectoires observées. Selon les résultats observés lors des expérimentations des modèles présentés dans ce chapitre, il apparaît en effet que si des tendances communes se retrouvent dans l'espace latent appris, par des corrélations des trajectoires pour les auteurs aux centres d'intérêts proches, on peut difficilement en déduire des relations d'influence à proprement parler. Par ailleurs, la structuration de l'espace a pour objectif de limiter les problèmes éventuels de biais d'exposition (notamment en prédiction) et de vacance de zones de l'espace latent (i.e., zones de l'espace ne codant pour aucune donnée réaliste). La section 4.4.2 présente quant à elle des perspectives de travail en prédiction stochastique inductive, où l'on ne modélise plus une seule trajectoire unique mais où l'on cherche à encoder des ensembles de trajectoires à partir de séquences de textes observés (fils de discussion par exemple). Enfin, la section 4.4.3 discute de possibilités d'application en ligne des modèles proposés.

## 4.4.1 Structuration de l'espace latent

### 4.4.1.1 Auto-encodage textuel variationnel

Une piste pour structurer l'espace latent et obtenir des trajectoires plus informatives dans cet espace serait en premier lieu de travailler par auto-encodage des textes considérés. L'idée serait

de considérer des codes individuels  $z_i$  pour chaque document  $d^{(i)}$  du corpus de texte, issus par exemple d'une gaussienne centrée sur l'état de l'auteur de ce document à son instant de publication  $p_{z_d}(z_i|z_{a^{(i)},t^{(i)}},\sigma_d^2) = \mathcal{N}(z_i; z_{a^{(i)},t^{(i)}}, \sigma_d^2 \mathbf{I})$ . Selon les principes de l'auto-encodage variationnel [Kingma and Welling, 2013], on pourrait alors considérer un encodeur de texte (e.g., un LSTM) émettant les paramètres d'une gaussienne dans l'espace latent  $q_\xi(z_i|d^{(i)})$  pour tout texte d'entrée  $d^{(i)}$ , résultant à l'ELBO suivante à maximiser :

$$\begin{aligned} \mathcal{L}(\mathbf{X}; \theta, \Psi, \xi) = & \sum_{a \in \mathcal{A}} \sum_{t=1}^T \mathbb{E}_{q_\Psi(Z_{a,1:t})} \left[ \sum_{d \in X_{a,t}} \mathbb{E}_{q_\xi(z|d)} [\log P_\theta(d|z, z_a)] - \text{D}_{\text{KL}}(q_\xi(\cdot|d) \| p_{z_d}(\cdot|z_a, t, \sigma_d^2)) \right. \\ & \left. - \text{D}_{\text{KL}}(q_\Psi^{a,t}(\cdot|z_{a,t-1}, z_a) \| p_z(\cdot|\mu_{a,t}, \sigma^2)) \right] \end{aligned} \quad (4.12)$$

où  $\xi$  correspond à l'ensemble des paramètres variationnels d'encodage des textes. L'objectif de ce genre d'encodage est d'amener le modèle à attribuer des états  $z_{a,t}$  déductibles des groupes de textes considérés pour toute paire  $(a, t)$ , via l'encodeur  $q_\xi$ , pour renforcer la sémantique de l'espace de représentation. Cependant les premières expérimentations dans cette voie n'ont pas donné de résultats probants, pour diverses raisons.

**Effondrement de la postérieure** Premièrement, il est bien connu que l'auto-encodage variationnel est difficile dans le cadre du texte, avec effondrement de la distribution postérieure (*posterior collapse*) souvent constaté, car l'aspect auto-régressif des décodeurs leur permet d'atteindre des performances satisfaisantes en se passant de l'encodage. Diverses techniques ont été proposées dans la littérature pour répondre à ce problème, à savoir notamment par des techniques de réduction des capacités du décodeur en apprentissage, via par exemple le *word dropout* (i.e., annulation aléatoire de représentations de certains mots des textes à décoder) [Bowman et al., 2015] ou via la définition d'architectures de décodage moins puissantes [Yang et al., 2017]. D'autres techniques proposent de relâcher des contraintes sur l'espace d'encodage, tel le *KL annealing* [Bowman et al., 2015], qui consiste à augmenter progressivement le poids du terme de KL (en partant de 0 jusqu'à 1) afin de permettre à l'encodeur de se spécialiser sur les codes produits avant d'appliquer de trop fortes contraintes sur l'encodage<sup>2</sup>, ou le *KL Thresholding* (aussi nommé *FreeBits*) [Kingma et al., 2016], qui consiste à contraindre l'encodage seulement jusqu'à un certain seuil de KL que l'on considère acceptable (éventuellement séparément sur les différentes dimensions). Le travail de [Li et al., 2019] a montré qu'une initialisation de l'encodeur par auto-encodage sans contrainte d'initialisation, suivi d'un seuillage de KL lors de l'apprentissage, permettait d'obtenir des résultats compétitifs sur divers jeux de données textuelles. Dans une autre veine, [He et al., 2019] propose de donner un poids plus agressif à l'apprentissage de l'encodeur qu'à celui du décodeur, constatant que lorsque les problèmes d'effondrement de la distribution postérieure surviennent, cela est souvent dû à un retard d'entraînement de l'encodeur par rapport au décodeur. L'approche dans [Wu et al., 2020] propose de coupler leur auto-encodeur variationnel à un auto-encodeur déterministe pour éviter l'effondrement de la postérieure. Enfin, plutôt que de travailler avec des méthodes variationnelles classiques, divers travaux proposent des méthodes génératives adverses, type GAN, avec apprentissage d'encodeur [Makhzani et al., 2015]. Ces auto-encodeurs adverses (AAE), maintenant un fort couplage encodeur-décodeur, ont généralement moins tendance à ignorer le code donné au générateur. Pour l'encodage textuel, il est par ailleurs possible d'intégrer des perturbations dans l'espace d'encodage des AAE pour en améliorer la structuration [Shen et al., 2019].

2. Une version avec poids cycliques a récemment été proposée dans la littérature [Fu et al., 2019].

**Contraintes d'encodage** Deuxièmement, la charge demandée à l'encodeur est très importante : il doit extraire un code utile pour le décodage, en faisant en sorte que l'espérance des codes générés pour les textes d'un auteur à un temps donné soit proche de l'état  $z_{a,t}$ , lui-même mobile au cours de l'apprentissage. D'autre part, pour contenir de l'information utile, les codes tendent à être inférés loin de  $z_{a,t}$ , ce qui induit une variance élevée due à la contrainte de KL, et ne permet pas de stabiliser l'apprentissage du fait d'échantillons trop variables à chaque étape. Une manière de relâcher cette pression serait d'amener l'encodeur à produire des codes de prior  $\mathcal{N}(0, 1)$ , venant s'ajouter à  $z_{a,t}$  pour former  $z_i$  en fonction de  $d^{(i)}$  (plutôt que de lui demander d'inférer la totalité de  $z_i$ , avec contraintes de vraisemblance selon  $z_{a,t}$ ). De cette manière, on encourage l'encodeur à ne se concentrer que sur les invariants à l'auteur et au temps du texte d'entrée (on peut aussi éventuellement passer l'auteur en entrée de l'encodeur pour ne gommer que l'aspect temporel). Quelles que soient l'auteur et le temps, les différentes thématiques sont alors toujours représentées de la même manière autour des états dynamiques, ce qui structure l'espace, tout en considérant une KL par rapport à un prior fixe (plutôt que dépendant de  $z_{a,t}$ ). Une autre manière de faire serait de ne contraindre qu'une partie du code issu de l'encodeur à être centré sur  $z_{a,t}$ , tout en maximisant un terme d'information mutuelle entre cette partie du code et le texte décodé, comme proposé dans le cadre des GANs dans [Chen et al., 2016b].

**Aspect multi-thématique** Troisièmement, l'aspect multi-thématique des textes peut être difficile à modéliser par de simples gaussiennes multivariées dans l'espace de représentation. Suivant des travaux comme [Xiao et al., 2018] ou [Wang et al., 2019], il serait possible de définir des encodeurs produisant des mixtures de gaussiennes, chacune centrée sur une partie du vecteur d'état dynamique  $z_{a,t}$ , ainsi que les paramètres d'une multinomiale permettant d'inférer la composante à considérer. Cet aspect multi-thématique, au delà de l'apport de flexibilité qu'il induit sur l'espace d'encodage, pourrait également permettre de mettre en lumière des évolutions relationnelles entre thèmes de la communauté. On pourrait en outre imaginer des évolutions de priors de Dirichlet à travers le temps pour modéliser les tendances thématiques de la communauté. Le travail dans [Zaheer et al., 2017], qui définit un réseau LSTM avec allocation latente de Dirichlet pour la prise en compte de thématiques dans les modèles de langue d'une communauté d'auteurs, est une autre source d'inspiration importante pour cet aspect.

**Vacance de code** Enfin, le modèle se heurte à des problèmes de vacance dans l'espace de représentation. Il existe de nombreuses paires (auteur, temps) ne correspondant à aucun texte disponible en apprentissage. Ces paires  $(a, t)$  correspondent alors à des états  $z_{a,t}$  jamais vus par le décodeur si les zones sont relativement éloignées les unes des autres. Les auteurs de [Xu et al., 2019] mettent en évidence que, dans le cas des méthodes variationnelles d'auto-encodage textuel classique, la distribution postérieure agrégée  $q(z)$  possède de nombreuses zones de très faible densité, où le décodeur peut avoir du mal à généraliser, qui peuvent néanmoins être atteintes probablement en test. C'est d'autant plus vrai dans notre cas avec priors dynamiques. Pour répondre à ce problème, il serait possible de s'inspirer de [Xu et al., 2019] qui propose de contraindre la postérieure d'encodage gaussienne à être issue d'une projection d'une distribution du simplexe, dans laquelle il est possible de contraindre le remplissage de l'espace utile. Cela a permis aux auteurs d'obtenir de bons résultats pour la manipulation de textes à partir des codes dans le simplexe. Ce genre d'approche pourra dans notre cas permettre de contrôler les trajectoires d'états, de manière à ce qu'elles restent dans des zones connues de l'espace de représentation.

#### 4.4.1.2 Biais d'exposition en prédiction

Les tâches de prédiction envisagées dans ce chapitre visent à définir des états futurs en déroulant la fonction de transition récursivement jusqu'à atteindre le temps souhaité. Un risque important est d'atteindre des zones inconnues du décodeur, menant alors à la génération de textes incohérents. Par ailleurs, il est probable que les erreurs de la fonction de transition s'accumulent et fassent diverger les trajectoires très loin de la zone connue à assez court terme. Une possibilité pour limiter l'accumulation d'erreurs est de se tourner vers des architectures d'auto-encodage de type *TD-VAE* [Gregor et al., 2019], qui considèrent des transitions multiples à chaque étape. En définissant des réseaux d'inférence basés sur des variables latentes séparées de plusieurs pas de temps, ce genre de modèle permet de limiter les biais d'exposition, déjà discutés en section 3.4.2.3. Cependant, cela ne traite pas la possibilité de se trouver loin de la zone connue en prédiction hors des temps d'apprentissage. Le travail de [Xu et al., 2019] discuté à la section précédente pourrait constituer une réponse possible à ce problème. Dans ce cas, les états dynamiques pourraient vivre dans un espace contraint (e.g., une boule), dans lequel des déplacements réguliers pourraient être appris et extrapolés, puis projetés dans un simplexe évitant les problèmes d'effondrement de postérieure et de vacance de codes, avant échantillonnage selon la gaussienne correspondante et décodage. Cependant cela peut paraître quelque peu contre-productif dans notre cas : contenir les trajectoires futures à l'intérieur du simplexe, du moins d'un simplexe rempli des observations d'apprentissage comme c'est le cas dans [Xu et al., 2019], risque de s'avérer limité en terme d'évolution des distributions de textes générés. L'objectif est d'extrapoler les évolutions observées sur les périodes d'apprentissage, ce qui ne peut se faire dans un espace fortement contraint, à moins de travailler sur des données cycliques. Une autre possibilité pourrait être d'émettre des hypothèses sur les propriétés que doit respecter un langage sur une communauté d'auteurs. Une piste, très exploratoire, serait par exemple de se baser sur la cohérence de groupe (i.e., les auteurs d'un même groupe doivent conserver un langage commun dans le temps), ce qui reviendrait à générer des textes à des temps extrapolés selon la fonction de transition et retro-propager un signal de ces textes générés correspondant à une mesure de la cohérence des différents langages engendrés sur les auteurs de la communauté (tout en évitant l'effondrement sur un codage commun à trop court terme). On pourrait aussi s'intéresser à l'invariance de la distribution intemporelle des différents auteurs : les textes générés sur un ensemble de pas de temps échantillonnés pour un auteur dans et hors de la période d'apprentissage doivent posséder une distribution postérieure agrégée proche de celle des textes d'apprentissage.

#### 4.4.1.3 Inférence de relations

Le modèle à dynamiques individuelles permet d'apprendre des trajectoires individuelles pour les différents auteurs, avec leurs propres évolutions. Cependant, puisqu'elles vivent dans un même espace et partagent une fonction de transition commune, de fortes corrélations peuvent exister entre certaines trajectoires, conformément à ce que l'on observe en figure 4.4. Pour autant, en l'état ces corrélations ne nous ont pas permis d'extraire de réelles relations d'influence, ni d'identifier des leaders d'influence dans les expérimentations menées sur nos jeux de données. Une investigation plus en profondeur de ce point devrait être menée, éventuellement sur données artificielles que l'on peut contrôler, pour mieux comprendre les capacités du modèle et identifier les types d'influence que l'on peut retrouver par examen de l'espace d'états appris.

Néanmoins, une possibilité pour renforcer ce point, et amener les trajectoires à s'organiser les unes par rapport aux autres selon les jeux d'influence de la communauté, pourrait être de réfléchir à

une fonction de transition s'appuyant explicitement sur les valeurs d'états de l'ensemble des auteurs au pas de temps précédent, selon par exemple un mécanisme d'attention, pondérant les importances des états de tous les auteurs dans la construction des transitions individuelles. Une autre possibilité serait de s'inspirer de [Delasalles et al., 2019d], qui décrit un modèle à états avec apprentissage de dépendances entre séries temporelles. En plus d'apprendre des états et un décodeur, le modèle considère un coût de régularisation de chaque état selon une fonction de l'ensemble des états au temps précédent, avec matrice de poids apprise, correspondant aux dépendances temporelles entre séries. Dans notre cas, ce coût pourrait être appliqué sur les séquences d'états après échantillonnage, ou entre distributions variationnelles apprises. La matrice de poids et/ou d'attentions apprise peut être statique (i.e., constante sur l'ensemble des pas de temps) ou dynamique, avec dans ce cas un prior de régularité pour stabiliser les influences de chaque auteur dans le temps.

Notons que pour renforcer les capacités d'extraction de relations d'influence du modèle, il faut peut-être accepter de perdre en qualité de modélisation, en régularisant plus fortement le décodeur pour donner plus de poids aux états de conditionnement, et en retirant le conditionnement statique  $z_a$  du décodeur pour forcer deux états proches à coder pour les mêmes distributions textuelles. Des priors, par exemple de Dirichlet, sur la matrice de poids d'influence pourraient également être considérés pour inciter le modèle à découvrir des relations fortes dans les données, en évitant un poids trop fort sur la diagonale de la matrice (auto-influence) et pénalisant les distributions de poids uniformes sur les autres variables. Ces priors peuvent par ailleurs dépendre d'indicateurs annexes, tels que des relations d'affinité découvertes dans les données textuelles [Tshimula et al., 2019], afin de guider le processus d'apprentissage.

#### 4.4.2 Prédiction stochastique inductive

Les modèles présentés dans ce chapitre s'intéressent à l'évolution de la langue d'une manière globale. Que ce soit pour le modèle général ou pour le modèle individuel qui s'intéresse à des dynamiques à l'échelle des auteurs, on ne considère pour chaque entité (i.e., la communauté pour le modèle général ou chacun des auteurs dans le cas du modèle individuel) qu'une seule trajectoire, qui correspond à l'évolution de son modèle de langue au fil du temps. Modéliser l'évolution de fils de discussion pourrait correspondre à une extension intéressante du travail réalisé, avec divers objectifs de prédiction.

Dans cette section, on considère que l'on dispose, en plus des données textuelles, d'une appartenance à un fil de discussion  $\tau$  pour chaque texte  $d \in X$ . Un fil de discussion  $\tau = (\tau_1, \dots, \tau_{|\tau|})$  est une séquence de textes (i.e.,  $\forall i \in \{1 \dots |\tau|\} : \tau_i \in X$ ), ordonnée selon la date de publication des textes. Être capable de prédire le futur d'une discussion à partir des premiers textes qui la composent peut avoir diverses applications : recommandation de fils de discussion, prévision d'évènements, modération de forums, extraction de dynamiques conversationnelles. Cela peut également être utile pour des méthodes d'apprentissage par renforcement basé sur des modèles [Gregor et al., 2019], par exemple pour l'apprentissage de *chatbots* [Serban et al., 2017b, Barlier et al., 2018]. L'objectif est de définir un modèle qui maximise la vraisemblance des textes de tout fil de discussion  $\tau$ , publiés après une date  $T$  donnée (avec  $T$  relatif au début de la discussion) :

$$\max_{\theta} \sum_{\tau} \sum_{\substack{\tau_i \in \tau, \\ t_{\tau}(\tau_i) \geq T}} \log P_{\theta}(\tau_i | (\tau_j)_{t_{\tau}(\tau_j) < T}, t_{\tau}(\tau_i), a(\tau_i)) \quad (4.13)$$

avec  $\theta$  les paramètres du modèle  $P_{\theta}$ ,  $a(\tau_i)$  l'auteur du texte  $\tau_i$  et  $t_{\tau}(\tau_i) = t(\tau_i) - t(\tau_1)$  le délai de publication de  $\tau_i$  après le premier texte de la discussion. Ainsi, le modèle est conditionné non seulement par les textes antérieurs à  $T$  dans  $\tau$ , mais également par la date de publication et l'auteur de chaque

texte. Restant dans le cadre de modèles basés sur des états latents, l'idée est d'extraire les dynamiques de trajectoires dans l'espace de représentation, afin d'être à même de conditionner des modèles de langue appliqués à l'état extrapolé pour un instant futur de la discussion. À l'instar des modèles présentés dans ce chapitre, nous nous appuyerons sur des modèles neuronaux résiduels dans l'espace de représentation. Cependant, puisque la trajectoire latente est conditionnée aux premiers textes de la discussion, il s'agira d'employer un encodeur textuel pour déterminer les états successifs de la discussion considérée.

Nous nous appuyerons sur une approche récemment développée dans le cadre des thèses de Jean-Yves Franceschi et Edouard Delasalles, pour la prédiction de vidéos stochastiques [Franceschi et al., 2020]. Dans cette contribution, l'objectif était de prédire la suite d'une vidéo connaissant son début. À l'instar des fils de discussion, les vidéos correspondent à des données souvent fortement stochastiques [Denton and Fergus, 2018], puisque correspondant à des représentations dynamiques du monde réel, qui n'est par nature pas déterministe. Ces données séquentielles sont stochastiques dans le sens où plusieurs futurs  $\tau_{t+1:T}$  différents peuvent être associés à un même début de  $t$  composantes  $\tau_{1:t}$  (images ou textes). Pour capturer l'incertitude du modèle et produire des distributions de probabilités sur les différents futurs possibles, plutôt qu'uniquement les modes de ces distributions, nous employons là encore un modèle bayésien. Le modèle génératif considéré est donné dans la figure 4.6a. Avec des états latents  $z_t$  déterministes en fonction d'un état initial gaussien  $z_1$  et de composantes stochastiques gaussiennes  $\epsilon_t$  à chaque étape  $t$  (i.e.,  $z_{t+1} = z_t + f_\theta(z_t, \epsilon_{t+1})$  avec  $f_\theta$  la fonction de transition de paramètres  $\theta$ ), le modèle est très similaire au modèle proposé en section 4.3.2 pour la modélisation dynamique stochastique de langue, à ceci près que les états latents sont différents pour chaque séquence considérée. Les composantes  $\epsilon_t$  doivent alors être inférés par un encodeur probabiliste en fonction des éléments précédant  $t$ , alors qu'en section 4.3.2 l'inférence variationnelle se faisait sans avoir recours à un encodeur. Il était en effet possible de stocker des paramètres variationnels pour chaque utilisateur, ce qui n'est plus le cas ici car on veut pouvoir généraliser à toute nouvelle séquence dont on observe le début (toute nouvelle vidéo dans [Franceschi et al., 2020] ou tout nouveau fil de discussion dans la perspective d'application envisagée). Le modèle d'inférence est schématisé en figure 4.6b. Il indique l'extraction de l'état initial  $z_1$  à partir des frames de conditionnement (ici 2), ainsi que les distributions des composantes stochastiques  $\epsilon_t$  en fonction des frames antérieures (i.e.,  $\epsilon_t \sim \mathcal{N}(\mu_\phi((\tau_{1:t}), \sigma_\phi((\tau_{1:t})\mathbb{I}))$ , avec  $\mu_\phi$  et  $\sigma_\phi$  respectivement la moyenne et la variance diagonale d'une gaussienne, obtenues par un réseau de neurones récurrent appliqué sur les éléments du temps 1 au temps  $t$  dans la séquence  $\tau$ ).

La prédiction de futurs de fils de discussion s'apparente donc à une transposition de ce genre de modèle pour la modélisation de langue. Il s'agira d'utiliser des modèles d'encodage et décodage adaptés au texte, là où l'on a utilisé des réseaux convolutionnels VGG [Simonyan and Zisserman, 2014] ou DCGAN [Radford et al., 2016] pour l'encodage ou le décodage d'images dans [Franceschi et al., 2020]. Une difficulté supplémentaire, déjà discutée en section 4.4.1.1, réside dans l'aspect auto-régressif des textes, qui amène souvent les modèles de décodage à ignorer les états latents inférés. Des métriques de réalisme des séquences prédites pourront également être prises en compte, par l'apprentissage de discriminateurs adverses et le renforcement des séquences trompant le discriminateur (techniques déjà abordées en section 3.4.2.3 dans le contexte de la diffusion d'information).

Une autre tâche de prédiction, plus proche des tâches considérées en diffusion d'information au chapitre précédent, serait de chercher à prédire les utilisateurs qui vont participer à une discussion donnée, connaissant son début et un modèle de dynamiques sur la communauté. Plutôt que de chercher à prédire des publications connaissant l'auteur, on pourrait viser à prédire quels seront les auteurs qui publieront (et éventuellement quand). Divers modèles pourront être considérés, éven-

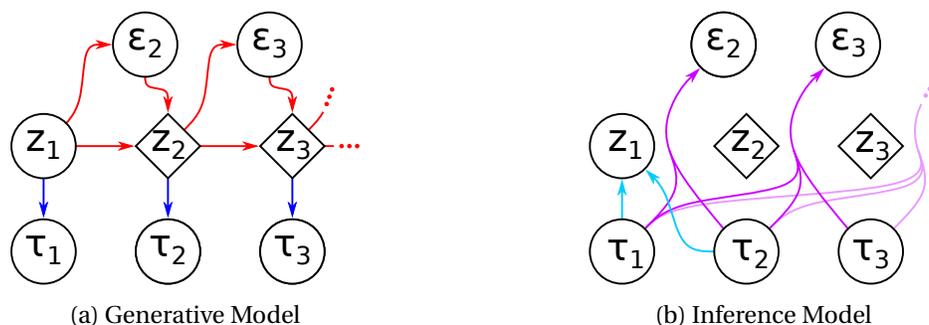


FIGURE 4.6 – Modèle génératif (a) et modèle d’inférence (b) définis pour la prédiction de vidéos stochastiques dans notre contribution [Franceschi et al., 2020], et envisagés pour la prédiction de futurs de fils de discussion.

tuellement basés sur des relations d’influence découvertes selon les approches envisagées à la section 4.4.1.3. L’extraction des fils de discussion dans les données textuelles selon les modèles en place constitue également une piste de recherche intéressante, à l’instar de l’extraction d’épisodes de diffusion énoncée en perspectives du chapitre précédent (section 3.4.1.3).

### 4.4.3 Modèles en ligne

Une application en ligne des modèles proposés est bien sûr également à envisager. La modélisation de la langue en ligne, à partir de données textuelles arrivant en flux a été explorée depuis de nombreuses années, avec diverses approches visant à allier complexité constante (ou quasi) et adaptation des modèles, souvent via des mixtures de modèles statiques et dynamiques. Par exemple, le travail dans [Lin et al., 2011] compare diverses méthodes de lissage (e.g., lissage de Jelinek-Mercer, lissage bayésien de Dirichlet, etc.), pour combiner un modèle de langue statique, appris sur un corpus hors-ligne, et un modèle dynamique, dont les distributions sont issues de comptages dans un historique de textes récents. Différentes stratégies de rétention d’historique sont également considérées, dont une liste FIFO, telle qu’on l’a envisagé dans la section 2.2.3.1, et une technique proposée dans [Goyal et al., 2009], qui “oublie” périodiquement les n-grammes de plus faible fréquence pour maintenir une complexité de croissance logarithmique. Les expérimentations montrent qu’il est possible de relativement bien prédire l’appartenance thématique des textes arrivant en flux, en utilisant un historique en file d’attente et une stratégie dite de *stupid Backoff* normalisée [Brants et al., 2007], qui consiste à utiliser le modèle dynamique uniquement pour les n-grammes dont le nombre d’occurrences dans l’historique dépasse un certain seuil (probabilité selon le modèle de lissage statique sinon). Dans la même veine, [Petrović et al., 2010] propose un algorithme de hachage local (LSH) qui vise à classer les documents arrivant en flux selon des types d’évènements avec une complexité constante. Un autre exemple dans [Levenberg and Osborne, 2009] utilise des filtres de Bloom dynamiques pour établir des modèles de langue en ligne, se mettant à jour continuellement, pour une tâche de traduction en ligne. Enfin, [Hoffman et al., 2010] propose une version en ligne du modèle LDA, pour la représentation de modèles de langue à base de mixtures de thématiques, reposant sur un processus d’inférence variationnelle stochastique. Néanmoins dans tous ces travaux, les approches reposent sur des modèles de langue très simples à base de distributions multinomiales n-grammes.

Bien sûr ces approches de modélisation en ligne ont été étendues plus récemment aux modèles de langue neuronaux. Par exemple, le travail dans [Grave et al., 2016] propose un modèle de langue récurrent s’inspirant du modèle à cache de [Kuhn, 1988], proposé dans le domaine de la reconnais-

sance de la parole. L'idée de [Grave et al., 2016] est de s'appuyer sur un cache continu des états cachés d'un réseau de neurones récurrent, afin de définir la probabilité conditionnelle d'un mot  $w$  selon son contexte  $h_{1:t}$  par une combinaison de 1) la probabilité de ce mot selon le réseau de neurones récurrent statique appris hors ligne et 2) une somme de mesures dépendant de proximités entre l'état caché du RNN encodant le contexte  $h_{1:t}$  et des états du même RNN lorsque le mot à décoder était également  $w$  dans un historique récent. Cela permet de prendre en compte les spécificités de la langue à un instant donné, en plus d'un modèle de langue général constant, et ainsi par exemple éviter des problèmes de mots hors vocabulaire si ceux-ci ont été rencontrés récemment. Divers autres modèles de langue neuronaux à base de mémoire et réseaux de pointeurs ont été proposés dans ce cadre [Merity et al., 2016, Sukhbaatar et al., 2015, Kumar et al., 2016], mais tous ces modèles restent adossés à un modèle de langue statique, sans mise à jour en ligne de ses paramètres.

Globalement, assez peu de travaux ont été proposés pour l'apprentissage continu en ligne de modèles de langue neuronaux, selon des données arrivant en flux de documents, particulièrement prenant en compte la temporalité des documents publiés. On note l'approche dans [Djuric et al., 2015], qui propose un modèle hiérarchique traitant de flux de documents, où les probabilités d'émission des mots dépendent des distributions des documents proches dans le flux, supposant ainsi certaines tendances dynamiques dans la langue modélisée. Cependant, le modèle suppose une segmentation des flux connue, repose sur des heuristiques basées sur des représentations en sacs de mots simplistes, n'a pas de mécanisme d'apprentissage des évolutions de la langue avec capacités d'anticipation et ne capture pas l'incertitude des modèles. Une autre possibilité serait de simplement mettre à jour un modèle de langue type LSTM standard par simple montée de gradient stochastique pour chaque nouvelle donnée entrante. Cependant, le nombre et la magnitude des pas de gradient à effectuer sont très délicats à régler efficacement, car ils ont un impact direct sur la sur ou la sous-adaptation du modèle aux nouveaux textes observés à chaque époque. Travailler par montée de gradient stochastique sur les données d'une fenêtre glissante d'historique est une solution pour limiter le problème, mais le risque associé est de pas être à jour sur les tendances des tous derniers pas de temps de la fenêtre.

L'application en ligne des approches présentées dans ce chapitre pourrait venir combler ces limites. Supposant que la majeure partie des besoins d'adaptation du modèle aux nouvelles tendances du langage sont capturés par l'état dynamique du système, l'optimisation nécessaire du modèle de langue à chaque étape se limite alors à l'adaptation éventuelle à de nouveaux états de conditionnement. Afin de ne pas considérer à chaque étape un échantillonnage de la trajectoire depuis le tout début du processus (et éviter la très coûteuse retro-propagation dans le temps associée), une possibilité est de considérer en début de fenêtre des priors sur les états dépendant de distributions variationnelles postérieures selon les données aux temps précédents, à la manière de ce qui a été réalisé à la section 2.2.3.1, dans le cadre des bandits contextuels variationnels. L'utilité de l'application de ces modèles dynamiques en ligne serait double : disposer à tout temps  $t$  d'un modèle efficace pour la modélisation à  $t$ , malgré d'éventuels faibles volumes de données à chaque période, et être capable d'anticiper les tendances futures de la communauté, par exemple pour des objectifs d'apprentissage par renforcement basé sur des modèles. Dans le cadre des bandits discutés en chapitre 2, on pourrait par exemple être intéressés par maximiser l'adéquation avec un certain style de texte dans les pas de temps futurs, et choisir les utilisateurs à écouter selon ces prévisions futures à moyen terme. Une caractérisation bayésienne de l'incertitude du modèle pourra en outre être envisagée pour garantir une exploration efficace.

Notons enfin qu'un mécanisme de génération de mots nouveaux pourrait être inclus dans nos modèles pour s'adapter au potentiel nouveau vocabulaire rencontré lors des processus en ligne, soit par combinaison de parties de mots (comme dans le modèle BERT [Devlin et al., 2019] par exemple),

soit par élargissement du vocabulaire en fonction des mots nouveaux rencontrés (envisageable assez simplement dans notre architecture neuronale avec *weight tying*, puisqu'il suffit alors d'apprendre une représentation distribuée pour chaque nouveau mot à ajouter).

# Chapitre 5

## Conclusion

La littérature sur l'apprentissage de modèles à partir de données distingue souvent deux types d'incertitude : l'incertitude aléatoire et l'incertitude épistémique [Sallak et al., 2013, Chua et al., 2018]. Alors que la première est inhérente à la nature même des données considérées, la seconde est liée au manque de données d'entraînement du modèle. Si dans le cadre des algorithmes de bandit au chapitre 2, l'incertitude des modèles était plutôt de nature épistémique (capturée pour garantir une exploration efficace de l'espace des paramètres), les données séquentielles issues des media sociaux sont fortement stochastiques par nature, les comportements des auteurs de contenus pouvant s'avérer très aléatoires. Nous nous sommes employés à en extraire néanmoins des tendances générales, tant pour la collecte de données à partir de flux d'information au chapitre 2, que pour la modélisation des jeux d'influence d'une communauté au chapitre 3, ou encore que pour la modélisation des évolutions temporelles de la langue en chapitre 4. Pour l'ensemble de ces problématiques, nous avons défini divers modèles, dont la plupart s'appuient sur des approches d'inférence bayésienne, notamment variationnelle, pour prendre en compte la nature stochastique des données, et combler divers manques d'informations pour les tâches d'apprentissage considérées. En conclusion de chaque chapitre, nous avons proposé diverses perspectives, visant à couvrir un large spectre de possibles extensions des modèles proposés, pour tenter d'en dépasser les limites identifiées et permettre leur application dans des contextes nouveaux. Beaucoup reste donc à faire pour répondre aux nombreux défis du domaine, mais ce travail d'écriture nous a permis d'organiser les choses et d'en apprécier l'ampleur.

Ce manuscrit, centré sur l'extraction de dynamiques de l'information, ne mentionne pas mes travaux antérieurs à 2010, qui correspondent à mon travail de thèse de doctorat en recherche d'information, et relèvent d'axes de recherche trop éloignés de cette thématique générale. Le document ne détaille pas non plus mes travaux pour la retro-ingénierie logicielle à partir de traces [Lamprier et al., 2013], [Lamprier et al., 2014] et [Lamprier et al., 2015a], ni les travaux engagés récemment en *Fair Machine Learning*, tels que [Grari et al., 2019], [Grari et al., 2020b] ou [Grari et al., 2020a], dans le cadre de la thèse de Vincent Grari. C'est le cas également de divers autres travaux en cours :

- Pour le résumé abstraitif, nous pouvons mentionner les travaux [Scialom et al., 2019], [Scialom et al., 2020b] et [Scialom et al., 2020a], menés dans le cadre de la thèse de Thomas Scialom, pour lesquels nous employons des transformeurs de type BERT [Devlin et al., 2019] pour l'encodage de textes et leur décodage sous forme de résumés. Notamment, notre approche [Scialom et al., 2019] s'appuie sur un générateur automatique de questions-réponses, et apprend par renforcement à produire des résumés contenant les réponses attendues pour des questions générées selon le texte à résumer. Les résultats obtenus sont très prometteurs, car sans requérir de résu-

més exemples supplémentaires sur de nouveaux corpus, cette méthode démontre une bonne capacité d'adaptation à des nouveaux types de textes. Néanmoins, cette méthode reste limitée par la capacité à générer des questions utiles, et un point limitant reste la production relativement fréquente de contre-sens dans le résumé produit. Des travaux sont actuellement en cours pour dépasser ce cadre et réduire encore l'expertise humaine introduite dans la méthode, à base d'apprentissage par renforcement inverse et l'utilisation de discriminateurs adverses pour l'identification automatique de ce qui fait qu'un résumé est bon ou mauvais. Notre approche récente dans [Scialom et al., 2020a] définit une méthodologie efficace pour l'apprentissage de GANs pour le résumé abstraitif, basée sur de l'échantillonnage préférentiel pour encourager une meilleure adaptation du discriminateur dans les zones utiles de l'espace de recherche.

- Pour la recherche d'information, nous participons avec Benjamin Piwowarski et notre docto-rante Agnès Mustar au projet ANR COST, qui vise à la modélisation utilisateur et à l'aide à la navigation dans le cadre de tâches de recherche d'information complexes. Après un premier travail sur la suggestion de requête à base de réseaux *Transformer* [Mustar et al., 2020], nos travaux actuels portent sur le guidage long terme des utilisateurs. L'objectif est de suggérer des requêtes qui permettent aux utilisateurs de s'orienter probablement vers des ensembles de documents pertinents cible, inconnus du modèle a priori. L'utilisation de méthodes d'apprentissage par imitation (type Goal-GAIL [Ding et al., 2019]) est envisagée pour la modélisation utilisateur avec documents cible connus. L'idée ensuite est d'utiliser ce modèle utilisateur pour apprendre par renforcement à aider les utilisateurs à raccourcir leurs sessions de recherche. Le découpage des sessions en sous-sessions est également envisagé, via notamment l'utilisation de méthodes de pre-entraînement hiérarchique du type de [Sukhbaatar et al., 2018], avec la définition de buts intermédiaires à atteindre pour guider la recherche.
- Pour la conduite autonome, je suis impliqué depuis peu, dans le cadre de la thèse de Manon Cé-saire et le projet SystemX EPI, dans un travail d'apprentissage de générateurs de scénarios réa-listes adverses. L'idée est la mise en difficulté d'un pilote autonome pour en accroître la robus-tesse. Deux pistes principales de recherche sont considérées actuellement : apprentissage par renforcement basé sur des modèles (s'appuyant notamment sur des modèles du type de ceux employés pour la prédiction de vidéos stochastiques, mentionnés à la section 4.4.2) et modèles adverses de perturbation long terme du pilote par altérations mineures de l'environnement (en s'inspirant des travaux décrits dans [Ilahi et al., 2020]).

Un point commun à la plupart de ces récents travaux est l'emploi de techniques d'apprentissage par renforcement, qui sont au cœur de la recherche en intelligence artificielle actuelle, puisqu'elles permettent l'émergence, sinon de raisonnements, de stratégies complexes avec prises de décisions à effet sur le long terme. Beaucoup reste à faire dans ce domaine néanmoins, notamment autour de problématiques d'optimisation pour environnements non stationnaires, dans lesquelles je suis impli-qué avec le co-encadrement de la thèse de Pierre-Alexandre Kamienny en CIFRE au laboratoire FAIR. D'autres de mes travaux vont dans ce sens, notamment au travers de l'encadrement du stage de fin d'études d'Hector Roussille, pour la définition de paradigmes de *curriculum learning* pour les sys-tèmes multi-agents. L'objectif est de mettre en place des méthodes d'apprentissage progressif favori-sant l'émergence de stratégies collaboratives ou adverses entre agents indépendants, en s'inspirant de travaux en curriculum learning mono-agent (par exemple le pre-entraînement asymétrique adverse proposé dans [Sukhbaatar et al., 2017]).

Pour l'extraction de dynamiques de l'information dans les réseaux sociaux, outre les applications déjà discutées en perspectives des différents chapitres de ce manuscrit (notamment pour limiter les

biais d'exposition des modèles de séquence), l'apprentissage par renforcement peut être employé à divers titres. Par exemple, on pourrait chercher à apprendre des politiques de recherche de contenus particuliers à court ou moyen terme dans les flux de données exploités au chapitre 2. À la différence des objectifs de collecte sans connaissance initiale qui nous placent de facto dans le cadre des problèmes de bandits, éventuellement avec fonctions de récompense évolutives (voir section 2.3.3), ce genre de tâche de recherche dynamique peut être traitée selon des buts échantillonnés d'une distribution pré-déterminée, et selon des récompenses obtenues à horizon fixé. L'objectif est d'apprendre sur quels utilisateurs se concentrer pour les différents types de contenu ciblés, et éventuellement exploiter les tendances pour s'orienter dans les flux. Pour ce dernier point, on pourra envisager des méthodes d'apprentissage par renforcement basées sur des modèles, du type de ceux présentés au chapitre 4. D'autres types d'application pourraient par exemple concerner la modélisation de diffusion et/ou la maximisation d'influence par injection de contenu, avec génération des contenus à injecter et choix possiblement séquentiels d'utilisateurs à cibler, plutôt que de dépendre de distributions de contenus pré-établis comme envisagé en section 3.4.3.1. La découverte automatique de fils de discussion en ligne à partir de flux de données issus des réseaux sociaux est un autre exemple d'application envisageable (e.g., avec récompenses dépendant de la cohérence/vraisemblance de la séquence collectée). Enfin, pour le cadre de la collecte de données à horizon infini, on pourrait viser le meta-apprentissage de politiques d'exploitation/exploration dans les flux de données, avec un objectif similaire à [Castro et al., 2012], mais par le biais de stratégies adaptatives, exploitant les connaissances acquises en continu sur le réseau, et anticipant les différents types de collecte à envisager. Un point commun à l'ensemble de ces tâches est le besoin de stratégies à forte plasticité structurelle, présentant de bonnes capacités d'adaptation et de transfert pour de possibles nouvelles conditions environnementales (e.g., nouveau contenu à collecter ou diffuser, communautés ou distributions changeantes, bruits liés à des événements externes, etc.). Les travaux dans [Finn et al., 2016, Nagabandi et al., 2018, Rusu et al., 2016, Chandak et al., 2020] sont autant de sources d'inspiration pour répondre aux nombreux défis de l'apprentissage en continu (ou *lifelong learning*) [Parisi et al., 2018], appliqué à l'extraction de dynamiques de l'information à partir de flux.

# Bibliographie

- [Abbasi-Yadkori et al., 2011] Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24 : 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 2312–2320. 18, 22, 23, 26, 27, 28
- [Agarwal et al., 2014] Agarwal, A., Hsu, D., Kale, S., Langford, J., Li, L., and Schapire, R. E. (2014). Taming the Monster : A Fast and Simple Algorithm for Contextual Bandits. *ArXiv e-prints*. 23
- [Aggarwal et al., 2003] Aggarwal, C. C., Han, J., Wang, J., and Yu, P. S. (2003). A framework for clustering evolving data streams. In *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29, VLDB '03*, pages 81–92. VLDB Endowment. 10
- [Aggarwal et al., 2004] Aggarwal, C. C., Han, J., Wang, J., and Yu, P. S. (2004). On demand classification of data streams. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04*, pages 503–508, New York, NY, USA. ACM. 10
- [Agrawal et al., 2015] Agrawal, S., Devanur, N. R., and Li, L. (2015). Contextual bandits with global constraints and objective. *CoRR*, abs/1506.03374. 23
- [Agrawal and Goyal, 2012] Agrawal, S. and Goyal, N. (2012). Analysis of thompson sampling for the multi-armed bandit problem. In *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, pages 39.1–39.26. 14
- [Agrawal and Goyal, 2013] Agrawal, S. and Goyal, N. (2013). Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 127–135. 14, 23
- [Alla et al., 1998] Alla, J., Lavrenko, V., and Papka, R. (1998). Event tracking. Technical Report CIIR Technical Report IR – 128, Department of Computer Science, University of Massachusetts. 10
- [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. 87
- [Anshelevich et al., 2009] Anshelevich, E., Chakrabarty, D., Hate, A., and Swamy, C. (2009). Approximability of the firefighter problem : Computing cuts over time. In *Proceedings of the 20th International Symposium on Algorithms and Computation*, pages 974–983. 59
- [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. 59, 87
- [Audibert and Bubeck, 2009] Audibert, J. and Bubeck, S. (2009). Minimax policies for adversarial and stochastic bandits. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*. 14, 18

- [Audibert et al., 2014] Audibert, J.-Y., Bubeck, S., and Lugosi, G. (2014). Regret in online combinatorial optimization. *Math. Oper. Res.*, 39(1) :31–45. 15
- [Audibert et al., 2009] Audibert, J.-Y., Munos, R., and Szepesvári, C. (2009). Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theor. Comput. Sci.*, 410(19) :1876–1902. 14, 17
- [Audiffren and Ralaivola, 2015] Audiffren, J. and Ralaivola, L. (2015). Cornering stationary and restless mixing bandits with remix-ucb. In *Advances in Neural Information Processing Systems 28 : Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3339–3347. 43
- [Auer et al., 2002] Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3) :235–256. 14
- [Azar et al., 2017] Azar, M. G., Osband, I., and Munos, R. (2017). Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 263–272. JMLR. org. 55
- [Bai et al., 2018] Bai, S., Kolter, J. Z., and Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271. 104, 106
- [Balali et al., 2013] Balali, A., Faili, H., Asadpour, M., and Dehghani, M. (2013). A supervised approach for reconstructing thread structure in comments on blogs and online news agencies. *Computación y Sistemas*, 17(2) :207–217. 90
- [Bamler and Mandt, 2017] Bamler, R. and Mandt, S. (2017). Dynamic word embeddings. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 380–389. 106, 110
- [Bao et al., 2015] Bao, Y., Huang, W., Yi, C., Jiang, J., Xue, Y., and Dong, Y. (2015). Effective deployment of monitoring points on social networks. In *International Conference on Computing, Networking and Communications, ICNC 2015, Garden Grove, CA, USA, February 16-19, 2015*, pages 62–66. 12
- [Barabási and Albert, 1999] Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *science*, 286(5439) :509–512. 83
- [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. 71, 90
- [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. 60
- [Barkan and Koenigstein, 2016] Barkan, O. and Koenigstein, N. (2016). Item2vec : neural item embedding for collaborative filtering. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE. 69
- [Barlier et al., 2018] Barlier, M., Laroche, R., and Pietquin, O. (2018). Training dialogue systems with human advice. In *Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 999–1007. 122
- [Barthelmé, 2015] Barthelmé, S. (2015). Expectation propagation : why use it, when to use it. 50
- [Bass, 1969] Bass, F. M. (1969). A new product growth for model consumer durables. *Management Science*, 15 :215–227. 3, 58

- [Bauer and Wortzel, 1966] Bauer, R. A. and Wortzel, L. H. (1966). Doctor's choice : The physician and his sources of information about drugs. *Journal of Marketing Research*, 3(1) :40–47. 3
- [Beal, 2003] Beal, M. J. (2003). *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London. 47
- [Bechini et al., 2016] Bechini, A., Gazzè, D., Marchetti, A., and Tesconi, M. (2016). Towards a general architecture for social media data capture from a multi-domain perspective. In *30th IEEE International Conference on Advanced Information Networking and Applications, AINA 2016, Crans-Montana, Switzerland, 23-25 March, 2016*, pages 1093–1100. 9
- [Belkin and Niyogi, 2003] Belkin, M. and Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6) :1373–1396. 10
- [Bengio et al., 2015] Bengio, S., Vinyals, O., Jaitly, N., and Shazeer, N. (2015). Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179. 94
- [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. 69
- [Bengio et al., 2003] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb) :1137–1155. 69, 104
- [Bengio et al., 1994] Bengio, Y., Simard, P. Y., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, 5(2) :157–166. 76
- [Benhamou et al., 2018] Benhamou, E., Atif, J., Laraki, R., and Auger, A. (2018). A new approach to learning in dynamic bayesian networks (dbns). *CoRR*, abs/1812.09027. 86
- [Bifet and Frank, 2010] Bifet, A. and Frank, E. (2010). Sentiment knowledge discovery in twitter streaming data. In *Discovery Science - 13th International Conference, DS 2010, Canberra, Australia, October 6-8, 2010. Proceedings*, pages 1–15. 11
- [Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. 36
- [Bishop et al., 1997] Bishop, C. M., Hinton, G. E., and Strachan, I. G. D. (1997). GTM through time. In *Fifth International Conference on Artificial Neural Networks (Conf. Publ. No. 440)*, pages 111–116. 105
- [Blei and Lafferty, 2006] Blei, D. M. and Lafferty, J. D. (2006). Dynamic topic models. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, pages 113–120. 105
- [Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3 :993–1022. 105
- [Blondel et al., 2017] Blondel, G., Arias, M., and Gavaldà, R. (2017). Identifiability and transportability in dynamic causal networks. *International journal of data science and analytics*, 3(2) :131–147. 88
- [Blundell et al., 2015] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural networks. *ArXiv*, abs/1505.05424. 52
- [Bnaya et al., 2013] Bnaya, Z., Puzis, R., Stern, R., and Felner, A. (2013). Bandit algorithms for social network queries. In *2013 International Conference on Social Computing*, pages 148–153. 15

- [Boanjak et al., 2012] Boanjak, M., Oliveira, E., Martins, J., Mendes Rodrigues, E., and Sarmento, L. (2012). Twitterecho : A distributed focused crawler to support open research with twitter data. In *Proceedings of the 21st International Conference Companion on World Wide Web, WWW '12 Companion*, pages 1233–1240, New York, NY, USA. ACM. 9
- [Bóta et al., 2013] Bóta, A., Krész, M., and Pluhár, A. (2013). Approximations of the generalized cascade model. *Acta Cybern.*, 21(1) :37–51. 61
- [Bourigault et al., 2014a] Bourigault, S., Lagnier, C., Lamprier, S., Denoyer, L., and Gallinari, P. (2014a). Apprentissage de représentation pour la diffusion d’information dans les réseaux sociaux. In *CORIA 2014 - Conférence en Recherche d’Informations et Applications- 11th French Information Retrieval Conference. CIFED 2014 Colloque International Francophone sur l’Ecrit et le Document, Nancy, France, March 19-23, 2014*, pages 155–170. 70, 93
- [Bourigault et al., 2014b] Bourigault, S., Lagnier, C., Lamprier, S., Denoyer, L., and Gallinari, P. (2014b). Learning social network embeddings for predicting information diffusion. In *Seventh ACM International Conference on Web Search and Data Mining, WSDM 2014, New York, NY, USA, February 24-28, 2014*, pages 393–402. 69, 70, 92
- [Bourigault et al., 2016a] Bourigault, S., Lamprier, S., and Gallinari, P. (2016a). Apprentissage de représentations pour la modélisation de processus de diffusion dans les réseaux sociaux. *Document Numérique*, 19(2-3) :31–52.
- [Bourigault et al., 2016b] Bourigault, S., Lamprier, S., and Gallinari, P. (2016b). Apprentissage de représentations probabilistes pour la prédiction de diffusions d’informations sur les réseaux sociaux. In *CORIA 2016 - Conférence en Recherche d’Informations et Applications - 13th French Information Retrieval Conference. CIFED 2016 Colloque International Francophone sur l’Ecrit et le Document, Toulouse, France, March 9-11, 2016*, pages 89–104.
- [Bourigault et al., 2016c] Bourigault, S., Lamprier, S., and Gallinari, P. (2016c). Learning distributed representations of users for source detection in online social networks. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II*, pages 265–281. 74, 85
- [Bourigault et al., 2016d] Bourigault, S., Lamprier, S., and Gallinari, P. (2016d). 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, San Francisco, CA, USA, February 22-25, 2016*, pages 573–582. 71, 72, 73, 79
- [Bourigault et al., 2017] Bourigault, S., Lamprier, S., and Gallinari, P. (2017). Apprentissage de représentation pour la détection de source dans les réseaux sociaux. In *Conférence en Recherche d’Informations et Applications - CORIA 2017, 14th French Information Retrieval Conference, Marseille, France, March 29-31, 2017. Proceedings*, pages 235–250.
- [Bourigault et al., 2018] Bourigault, S., Lamprier, S., and Gallinari, P. (2018). Détection de sources de diffusion par apprentissage de représentations distribuées. *Document Numérique*, 21(3) :11–31.
- [Bowman et al., 2015] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. *arXiv preprint arXiv :1511.06349*. 119
- [Bowman et al., 2016] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Józefowicz, R., and Bengio, S. (2016). Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 10–21. 109

- [Brants et al., 2007] Brants, T., Popat, A., Xu, P., Och, F. J., and Dean, J. (2007). Large language models in machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 858–867. 124
- [Bubeck and Cesa-Bianchi, 2012] Bubeck, S. and Cesa-Bianchi, N. (2012). Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends® in Machine Learning*, 5(1) :1–122. 14
- [Bubeck et al., 2008] Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. (2008). Online optimization in  $x$ -armed bandits. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 201–208. 51
- [Buccapatnam et al., 2014] Buccapatnam, S., Eryilmaz, A., and Shroff, N. B. (2014). Stochastic bandits with side observations on networks. In *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '14, Austin, TX, USA - June 16 - 20, 2014*, pages 289–300. 43
- [Cappé et al., 2013] Cappé, O., Garivier, A., Maillard, O.-A., Munos, R., and Stoltz, G. (Jun. 2013). Kullback-leibler upper confidence bounds for optimal sequential allocation. *Annals of Statistics*, 41(3) :1516–1541. 14
- [Carpentier and Valko, 2016] Carpentier, A. and Valko, M. (2016). Revealing graph bandits for maximizing local influence. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, Cadiz, Spain, May 9-11, 2016*, pages 10–18. 15, 43, 44, 97, 101
- [Casella and Robert, 1996] Casella, G. and Robert, C. P. (1996). Rao-blackwellisation of sampling schemes. *Biometrika*, 83(1) :81–94. 91
- [Castronovo et al., 2012] Castronovo, M., Maes, F., Fonteneau, R., and Ernst, D. (2012). Learning exploration/exploitation strategies for single trajectory reinforcement learning. In *Proceedings of the 10th European Workshop on Reinforcement Learning (EWRL 2012)*, pages 1–9. 129
- [Cataldi et al., 2010] Cataldi, M., Di Caro, L., and Schifanella, C. (2010). Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining, MDMKDD '10*, pages 4 :1–4 :10, New York, NY, USA. ACM. 11
- [Catanese et al., 2011] Catanese, S., Meo, P. D., Ferrara, E., Fiumara, G., and Provetti, A. (2011). Crawling facebook for social network analysis purposes. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics, WIMS 2011, Sogndal, Norway, May 25 - 27, 2011*, page 52. 9
- [Cella and Cesa-Bianchi, 2019] Cella, L. and Cesa-Bianchi, N. (2019). Stochastic bandits with delay-dependent payoffs. *arXiv preprint arXiv :1910.02757*. 55
- [Centola and Macy, 2007] Centola, D. and Macy, M. (2007). Complex contagions and the weakness of long ties. *American journal of Sociology*, 113(3) :702–734. 60
- [Cesa-Bianchi et al., 2013] Cesa-Bianchi, N., Gentile, C., and Zappella, G. (2013). A gang of bandits. In *Advances in Neural Information Processing Systems 26 : 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 737–745. 15, 43

- [Chakrabarti et al., 1999] Chakrabarti, S., van den Berg, M., and Dom, B. (1999). Focused crawling : A new approach to topic-specific web resource discovery. In *Proceedings of the Eighth International Conference on World Wide Web, WWW '99*, pages 1623–1640, New York, NY, USA. Elsevier North-Holland, Inc. 9
- [Chandak et al., 2020] Chandak, Y., Theodorou, G., Shankar, S., Mahadevan, S., White, M., and Thomas, P. S. (2020). Optimizing for the future in non-stationary mdps. *arXiv preprint arXiv :2005.08158*. 129
- [Chapelle and Li, 2011] Chapelle, O. and Li, L. (2011). An empirical evaluation of thompson sampling. In *Advances in Neural Information Processing Systems 24 : 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 2249–2257. 14, 23
- [Che et al., 2017] Che, T., Li, Y., Zhang, R., Hjelm, R. D., Li, W., Song, Y., and Bengio, Y. (2017). Maximum-likelihood augmented discrete generative adversarial networks. *arXiv preprint arXiv :1702.07983*. 95
- [Chen et al., 2018] Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018). Neural ordinary differential equations. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 6571–6583. Curran Associates, Inc. 118
- [Chen et al., 2015] Chen, S., Fan, J., Li, G., Feng, J., Tan, K.-l., and Tang, J. (2015). Online topic-aware influence maximization. *Proceedings of the VLDB Endowment*, 8(6) :666–677. 99
- [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. 70
- [Chen et al., 2010a] Chen, W., Wang, C., and Wang, Y. (2010a). 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. 60, 61
- [Chen et al., 2013] Chen, W., Wang, Y., and Yuan, Y. (2013). Combinatorial multi-armed bandit : General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 151–159. 15
- [Chen et al., 2016a] Chen, W., Wang, Y., Yuan, Y., and Wang, Q. (2016a). Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *The Journal of Machine Learning Research*, 17(1) :1746–1778. 97
- [Chen et al., 2010b] Chen, W., Yuan, Y., and Zhang, L. (2010b). Scalable influence maximization in social networks under the linear threshold model. In *2010 IEEE international conference on data mining*, pages 88–97. IEEE. 61
- [Chen et al., 2016b] Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016b). Infogan : Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180. 69, 120
- [Chen et al., 2020] Chen, Y.-C., Lu, P.-E., and Chang, C.-S. (2020). A time-dependent sir model for covid-19. *arXiv preprint arXiv :2003.00122*. 58
- [Cheng et al., 2010] Cheng, J., Sun, A. R., Hu, D., and Zeng, D. D. (2010). An information diffusion based recommendation framework for micro-blogging. *Available at SSRN 1713486*. 60

- [Chikhaoui et al., 2015] Chikhaoui, B., Chiazzaro, M., and Wang, S. (2015). A new granger causal model for influence evolution in dynamic social networks : The case of dblp. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 59
- [Chiu et al., 2018] Chiu, C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R. J., Rao, K., Gonina, E., Jaitly, N., Li, B., Chorowski, J., and Bacchiani, M. (2018). State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 4774–4778. 104
- [Cho et al., 2014] Cho, K., van Merriënboer, B., Gulcehre, C., 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)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics. 76
- [Chu et al., 2011] Chu, W., Li, L., Reyzin, L., and Schapire, R. E. (2011). Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, pages 208–214. 14, 21, 22
- [Chua et al., 2018] Chua, K., Calandra, R., McAllister, R., and Levine, S. (2018). Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, pages 4754–4765. 127
- [Chung et al., 2015] Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., and Bengio, Y. (2015). A recurrent latent variable model for sequential data. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 2980–2988. Curran Associates, Inc. 115
- [Clark et al., 2018] Clark, K., Luong, M.-T., Manning, C. D., and Le, Q. V. (2018). Semi-supervised sequence modeling with cross-view training. *arXiv preprint arXiv :1809.08370*. 69
- [Colbaugh and Glass, 2011] Colbaugh, R. and Glass, K. (2011). Emerging topic detection for business intelligence via predictive analysis of ‘meme’ dynamics. In *AI for Business Agility, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-03, Stanford, California, USA, March 21-23, 2011*. 12
- [Combes et al., 2015] Combes, R., Talebi, M. S., Proutière, A., and Lelarge, M. (2015). Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems 28 : Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2116–2124. 15
- [Cortes et al., 2017] Cortes, C., DeSalvo, G., Kuznetsov, V., Mohri, M., and Yang, S. (2017). Discrepancy-based algorithms for non-stationary rested bandits. *arXiv preprint arXiv :1710.10657*. 54
- [Daley et al., 2001] Daley, D. J., Gani, J., and Gani, J. M. (2001). *Epidemic modelling : an introduction*, volume 15. Cambridge University Press. 58
- [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*. 64
- [Datar et al., 2002] Datar, M., Gionis, A., Indyk, P., and Motwani, R. (2002). Maintaining stream statistics over sliding windows : (extended abstract). In *Proceedings of the Thirteenth Annual ACM-SIAM*

- Symposium on Discrete Algorithms, SODA '02*, pages 635–644, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics. 10
- [Dawkins, 1976] Dawkins, R. (1976). *The Selfish Gene*. Oxford University Press, Oxford, UK. 3
- [De Choudhury et al., 2010] De Choudhury, M., Lin, Y.-R., Sundaram, H., Candan, K. S., Xie, L., and Kelliher, A. (2010). How Does the Data Sampling Strategy Impact the Discovery of Information Diffusion in Social Media? In *Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*. 12
- [de la Peña et al., 2009] de la Peña, V. H., Lai, T. L., and Shao, Q. M. (2009). *Self-Normalized Processes : Limit Theory and Statistical Applications*. Springer Series in Probability and its Applications. Springer. 22, 27
- [de Masson d’Autume et al., 2019] de Masson d’Autume, C., Mohamed, S., Rosca, M., and Rae, J. (2019). Training language gans from scratch. In *Advances in Neural Information Processing Systems*, pages 4302–4313. 95
- [Dehaene and Barthelmé, 2018] Dehaene, G. and Barthelmé, S. (2018). Expectation propagation in the large data limit. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 80(1) :199–217. 50
- [Dehaene and Barthelmé, 2015] Dehaene, G. P. and Barthelmé, S. (2015). Bounding errors of expectation-propagation. In *Advances in Neural Information Processing Systems*, pages 244–252. 50
- [Delasalles et al., 2018] Delasalles, E., Lamprier, S., and Denoyer, L. (2018). Apprentissage de l’évolution langagière dans des communautés d’auteurs. In *Conférence en Recherche d’Informations et Applications - CORIA 2018, 15th French Information Retrieval Conference, Rennes, France, May 16-18, 2018. Proceedings*.
- [Delasalles et al., 2019a] Delasalles, E., Lamprier, S., and Denoyer, L. (2019a). Dynamic neural language models. In *Neural Information Processing - 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12-15, 2019, Proceedings, Part III*, pages 282–294. 106, 107, 109, 111
- [Delasalles et al., 2019b] Delasalles, E., Lamprier, S., and Denoyer, L. (2019b). Learning dynamic author representations with temporal language models. In *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*, pages 120–129. 112, 113
- [Delasalles et al., 2020] Delasalles, E., Lamprier, S., and Denoyer, L. (2020). Deep dynamic neural networks for temporal language modeling in author communities. *Knowledge and Information Systems*, to appear.
- [Delasalles et al., 2019c] Delasalles, E., Ziat, A., Denoyer, L., and Gallinari, P. (2019c). Spatio-temporal neural networks for space-time data modeling and relation discovery. *Knowl. Inf. Syst.*, 61(3) :1241–1267. 10
- [Delasalles et al., 2019d] Delasalles, E., Ziat, A., Denoyer, L., and Gallinari, P. (2019d). Spatio-temporal neural networks for space-time data modeling and relation discovery. *Knowledge and Information Systems*, 61(3) :1241–1267. 122
- [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. 63, 73

- [Deng et al., 2020] Deng, R., Chang, B., Brubaker, M. A., Mori, G., and Lehmman, A. (2020). Modeling continuous stochastic processes with dynamic normalizing flows. *arXiv preprint arXiv :2002.10516*. 118
- [Denton and Fergus, 2018] Denton, E. and Fergus, R. (2018). Stochastic video generation with a learned prior. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1174–1183, Stockholmsmässan, Stockholm, Sweden. PMLR. 123
- [Devlin et al., 2019] Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT : pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. 105, 106, 125, 127
- [Ding et al., 2017] Ding, S. H., Fung, B. C., Iqbal, F., and Cheung, W. K. (2017). Learning stylometric representations for authorship analysis. *IEEE Transactions on Cybernetics*. 111
- [Ding et al., 2019] Ding, Y., Florensa, C., Phielipp, M., and Abbeel, P. (2019). Goal-conditioned imitation learning. *Advances in Neural Information Processing Systems*. 128
- [Djuric et al., 2015] Djuric, N., Wu, H., Radosavljevic, V., Grbovic, M., and Bhamidipati, N. (2015). Hierarchical neural language models for joint representation of streaming documents and their content. In *Proceedings of the 24th international conference on world wide web*, pages 248–255. 125
- [Domingos and Hulten, 2000] Domingos, P. M. and Hulten, G. (2000). Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, MA, USA, August 20-23, 2000*, pages 71–80. 10
- [Dos Santos et al., 2018] Dos Santos, L., Piwowarski, B., Denoyer, L., and Gallinari, P. (2018). Representation Learning for Classification in Heterogeneous Graphs with Application to Social Networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(5) :1 – 33. 69
- [Du et al., 2016] Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., and Song, L. (2016). Recurrent marked temporal point processes : Embedding event history to vector. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*, pages 1555–1564, New York, NY, USA. ACM. 76, 83, 93
- [Du et al., 2014] Du, N., Liang, Y., Balcan, M., and Song, L. (2014). Influence function learning in information diffusion networks. In *International Conference on Machine Learning*, pages 2016–2024. 85
- [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. 64, 89
- [Eger and Mehler, 2016] Eger, S. and Mehler, A. (2016). On the linearity of semantic change : Investigating meaning variation via dynamic graph models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2 : Short Papers*. 105
- [El-Sayyad, 1973] El-Sayyad (1973). Bayesian and classical analysis of poisson regression. *J. of the Royal Statistical Society. Series B*. 51
- [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)*. 74, 82
- [Fedus et al., 2018] Fedus, W., Goodfellow, I. J., and Dai, A. M. (2018). Maskgan : Better text generation via filling in the \_\_\_\_\_. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 95, 104
- [Filippi et al., 2010] Filippi, S., Cappé, O., Garivier, A., and Szepesvári, C. (2010). Parametric bandits : The generalized linear case. In *Advances in Neural Information Processing Systems 23 : 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 586–594. 24, 51
- [Finn et al., 2016] Finn, C., Goodfellow, I., and Levine, S. (2016). Unsupervised learning for physical interaction through video prediction. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 64–72. Curran Associates, Inc. 129
- [Fraccaro et al., 2016] Fraccaro, M., Sønderby, S. K., Paquet, U., and Winther, O. (2016). Sequential neural models with stochastic layers. In *NeurIPS*. 52, 115
- [Franceschi et al., 2020] Franceschi, J., Delasalles, E., Chen, M., Lamprier, S., and Gallinari, P. (2020). Stochastic latent residual video prediction. In *Proceedings of the 37th International Conference on Machine Learning, ICML*. 123, 124
- [Freedman, 1975] Freedman, D. A. (1975). On tail probabilities for martingales. *the Annals of Probability*, pages 100–118. 28
- [Frermann and Lapata, 2016] Frermann, L. and Lapata, M. (2016). A bayesian model of diachronic meaning change. *TACL*, 4 :31–45. 105
- [Fu et al., 2019] Fu, H., Li, C., Liu, X., Gao, J., Celikyilmaz, A., and Carin, L. (2019). Cyclical annealing schedule : A simple approach to mitigating kl vanishing. *arXiv preprint arXiv :1903.10145*. 119
- [Furcy and Koenig, 2005] Furcy, D. and Koenig, S. (2005). Limited discrepancy beam search. In *IJCAI*. 94
- [Gal and Ghahramani, 2016] Gal, Y. and Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems 29 : Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 1019–1027. 110
- [Garivier and Moulines, 2011] Garivier, A. and Moulines, E. (2011). On upper-confidence bound policies for switching bandit problems. In *Algorithmic Learning Theory - 22nd International Conference, ALT 2011, Espoo, Finland, October 5-7, 2011. Proceedings*, pages 174–188. 43, 48
- [Gentile et al., 2014] Gentile, C., Li, S., and Zappella, G. (2014). Online clustering of bandits. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 757–765. 15, 24, 43
- [Gerrish and Blei, 2010] Gerrish, S. and Blei, D. M. (2010). A language-based approach to measuring scholarly impact. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 375–382. 105
- [Ghalebi et al., 2018] Ghalebi, E., Mirzasoleiman, B., Grosu, R., and Leskovec, J. (2018). Dynamic network model from partial observations. In *Advances in Neural Information Processing Systems*, pages 9862–9872. 99, 100

- [Gisselbrecht, 2017] Gisselbrecht, T. (2017). *Algorithmes de bandits pour la collecte d'informations en temps réel dans les réseaux sociaux. (Bandit Algorithms for Data Extraction on Social Media)*. PhD thesis, Pierre and Marie Curie University, Paris, France. 18, 48
- [Gisselbrecht et al., 2015a] Gisselbrecht, T., Denoyer, L., Gallinari, P., and Lamprier, S. (2015a). Apprentissage en temps réel pour la collecte d'information dans les réseaux sociaux. *Document Numérique*, 18(2-3) :39–58.
- [Gisselbrecht et al., 2015b] Gisselbrecht, T., Denoyer, L., Gallinari, P., and Lamprier, S. (2015b). Apprentissage en temps réel pour la collecte d'information dans les réseaux sociaux. In *CORIA 2015 - Conférence en Recherche d'Informations et Applications - 12th French Information Retrieval Conference, Paris, France, March 18-20, 2015.*, pages 7–22.
- [Gisselbrecht et al., 2015c] Gisselbrecht, T., Denoyer, L., Gallinari, P., and Lamprier, S. (2015c). Whichstreams : A dynamic approach for focused data capture from large social media. In *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26-29, 2015*, pages 130–139. 9, 15, 17, 101
- [Gisselbrecht et al., 2015d] Gisselbrecht, T., Lamprier, S., and Gallinari, P. (2015d). Policies for contextual bandit problems with count payoffs. In *27th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2015, Vietri sul Mare, Italy, November 9-11, 2015*, pages 542–549. 51
- [Gisselbrecht et al., 2016a] Gisselbrecht, T., Lamprier, S., and Gallinari, P. (2016a). Bandit contextuel pour la capture de données temps réel sur les médias sociaux. In *CORIA 2016 - Conférence en Recherche d'Informations et Applications- 13th French Information Retrieval Conference. CIFED 2016 Colloque International Francophone sur l'Ecrit et le Document, Toulouse, France, March 9-11, 2016, Toulouse, France, March 9-11, 2016.*, pages 57–72.
- [Gisselbrecht et al., 2016b] Gisselbrecht, T., Lamprier, S., and Gallinari, P. (2016b). Collecte ciblée à partir de flux de données en ligne dans les médias sociaux : une approche de bandit contextuel. *Document Numérique*, 19(2-3) :11–30.
- [Gisselbrecht et al., 2016c] Gisselbrecht, T., Lamprier, S., and Gallinari, P. (2016c). Dynamic data capture from social media streams : A contextual bandit approach. In *Proceedings of the Tenth International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016.*, pages 131–140.
- [Gisselbrecht et al., 2016d] Gisselbrecht, T., Lamprier, S., and Gallinari, P. (2016d). Linear bandits in unknown environments. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part II*, pages 282–298. 24
- [Gjoka et al., 2010] Gjoka, M., Kurant, M., Butts, C. T., and Markopoulou, A. (2010). Walking in facebook : A case study of unbiased sampling of osns. In *INFOCOM 2010. 29th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 15-19 March 2010, San Diego, CA, USA*, pages 2498–2506. 9
- [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. 59
- [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. 59, 60, 62, 64, 93

- [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. 59, 64
- [Gomez-Rodriguez et al., 2013a] Gomez-Rodriguez, M., Leskovec, J., and Schölkopf, B. (2013a). Modeling information propagation with survival theory. In *Proceedings of the 30th International Conference on Machine Learning (ICML-10)*. 64
- [Gomez-Rodriguez et al., 2013b] Gomez-Rodriguez, M., Leskovec, J., and Schölkopf, B. (2013b). 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. 64
- [Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc. 94
- [Goyal et al., 2009] Goyal, A., Daumé III, H., and Venkatasubramanian, S. (2009). Streaming for large scale nlp : Language modeling. In *Proceedings of Human Language Technologies : The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 512–520. 124
- [Granovetter, 1978] Granovetter, M. S. (1978). Threshold Models of Collective Behavior. *American Journal of Sociology*, 83(6) :1420–1143. 59
- [Grari et al., 2020a] Grari, V., Lamprier, S., and Detyniecki, M. (2020a). Adversarial learning for counterfactual fairness. *CoRR*, abs/2008.13122. 88, 127
- [Grari et al., 2019] Grari, V., Ruf, B., Lamprier, S., and Detyniecki, M. (2019). Fair adversarial gradient tree boosting. In *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*, pages 1060–1065. 127
- [Grari et al., 2020b] Grari, V., Ruf, B., Lamprier, S., and Detyniecki, M. (2020b). Fairness-aware neural rényi minimization for continuous features. In *IJCAI'20*. 127
- [Grave et al., 2016] Grave, E., Joulin, A., and Usunier, N. (2016). Improving neural language models with a continuous cache. *arXiv preprint arXiv :1612.04426*. 124, 125
- [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. 76
- [Graves and Schmidhuber, 2005] Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6) :602–610. 76
- [Greensmith et al., 2004] Greensmith, E., Bartlett, P. L., and Baxter, J. (2004). Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov) :1471–1530. 91
- [Gregor et al., 2019] Gregor, K., Papamakarios, G., Besse, F., Buesing, L., and Weber, T. (2019). Temporal difference variational auto-encoder. In *International Conference on Learning Representations*. 121, 122
- [Griffiths and Kalish, 2007] Griffiths, T. L. and Kalish, M. L. (2007). Language evolution by iterated learning with bayesian agents. *Cognitive science*, 31(3) :441–480. 105

- [Grover and Leskovec, 2016] Grover, A. and Leskovec, J. (2016). node2vec : Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. 69
- [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. 58, 62, 86
- [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. 60
- [Guille et al., 2013] Guille, A., Hacid, H., Favre, C., and Zighed, D. A. (2013). Information diffusion in online social networks : A survey. *ACM Sigmod Record*, 42(2) :17–28. 60
- [Guillou et al., 2016] Guillou, F., Gaudel, R., and Preux, P. (2016). Scalable explore-exploit collaborative filtering. In *Pacific Asia Conference On Information Systems (PACIS)*. Association For Information System. 15
- [Gupta et al., 2013] Gupta, P., Goel, A., Lin, J. J., Sharma, A., Wang, D., and Zadeh, R. (2013). WTF : the who to follow service at twitter. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 505–514. 9, 13
- [Guralnik and Srivastava, 1999] Guralnik, V. and Srivastava, J. (1999). Event detection from time series data. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '99*, pages 33–42, New York, NY, USA. ACM. 10
- [Gutowski et al., 2018] Gutowski, N., Amghar, T., Camp, O., and Chhel, F. (2018). Context enhancement for linear contextual multi-armed bandits. In Tsoukalas, L. H., Grégoire, É., and Alamaniotis, M., editors, *IEEE 30th International Conference on Tools with Artificial Intelligence, ICTAI 2018, 5-7 November 2018, Volos, Greece*, pages 1048–1055. IEEE. 24
- [Hall et al., 2008] Hall, D., Jurafsky, D., and Manning, C. D. (2008). Studying the history of ideas using topic models. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIG-DAT, a Special Interest Group of the ACL*, pages 363–371. 105
- [Hannon et al., 2010] Hannon, J., Bennett, M., and Smyth, B. (2010). Recommending twitter users to follow using content and collaborative filtering approaches. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, pages 199–206. 9, 13
- [Hawkes, 1971] Hawkes, A. G. (1971). Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1) :83–90. 93
- [He et al., 2019] He, J., Spokoyny, D., Neubig, G., and Berg-Kirkpatrick, T. (2019). Lagging inference networks and posterior collapse in variational autoencoders. *arXiv preprint arXiv :1901.05534*. 119
- [He et al., 2016] He, X., Xu, K., Kempe, D., and Liu, Y. (2016). Learning influence functions from incomplete observations. In *Advances in Neural Information Processing Systems*, pages 2073–2081. 89
- [Hernández-Lobato et al., 2016] Hernández-Lobato, J. M., Li, Y., Rowland, M., Bui, T. D., Hernández-Lobato, D., and Turner, R. E. (2016). Black-box alpha divergence minimization. In *ICML*. 52

- [Hesabi et al., 2015] Hesabi, Z. R., Sellis, T., and Zhang, X. (2015). Anytime concurrent clustering of multiple streams with an indexing tree. In *Proceedings of the 4th International Conference on Big Data, Streams and Heterogeneous Source Mining : Algorithms, Systems, Programming Models and Applications - Volume 41*, BIGMINE'15, pages 19–32. JMLR.org. 10
- [Ho and Ermon, 2016] Ho, J. and Ermon, S. (2016). Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573. 96
- [Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8) :1735–1780. 76, 106
- [Hoffman et al., 2010] Hoffman, M., Bach, F. R., and Blei, D. M. (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pages 856–864. 124
- [Hoffman et al., 2012] Hoffman, M., Blei, D. M., Wang, C., and Paisley, J. (2012). Stochastic Variational Inference. *ArXiv e-prints*. 39
- [Hong and Davison, 2010] Hong, L. and Davison, B. D. (2010). Empirical study of topic modeling in twitter. In *ECIR*. 31
- [Howard and Ruder, 2018] Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1 : Long Papers*, pages 328–339. 105
- [Huberman et al., 2008] Huberman, B., Romero, D., and Wu, F. (2008). Social networks that matter : Twitter under the microscope. *First Monday*, 14(1). 4
- [Hulten et al., 2001] Hulten, G., Spencer, L., and Domingos, P. (2001). Mining time-changing data streams. In *ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 97–106. ACM Press. 10
- [Ilahi et al., 2020] Ilahi, I., Usama, M., Qadir, J., Janjua, M. U., Al-Fuqaha, A., Hoang, D. T., and Niyato, D. (2020). Challenges and countermeasures for adversarial attacks on deep reinforcement learning. *arXiv preprint arXiv :2001.09684*. 128
- [Inan et al., 2017] Inan, H., Khosravi, K., and Socher, R. (2017). Tying word vectors and word classifiers : A loss framework for language modeling. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 110
- [Indyk and Motwani, 1998] Indyk, P. and Motwani, R. (1998). Approximate nearest neighbors : Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, STOC '98*, pages 604–613, New York, NY, USA. ACM. 10
- [Islam et al., 2018] Islam, M. R., Muthiah, S., Adhikari, B., Prakash, B. A., and Ramakrishnan, N. (2018). Deepdiffuse : Predicting the'who'and'when'in cascades. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1055–1060. IEEE. 77
- [Iwata et al., 2012] Iwata, T., Yamada, T., Sakurai, Y., and Ueda, N. (2012). Sequential modeling of topic dynamics with multiple timescales. *TKDD*, 5(4) :19 :1–19 :27. 105
- [Jaksch et al., 2010] Jaksch, T., Ortner, R., and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr) :1563–1600. 55
- [Joshi and Boyd, 2009] Joshi, S. and Boyd, S. (2009). Sensor selection via convex optimization. *Trans. Sig. Proc.*, 57(2) :451–462. 12
- [Jouxtel, 2005] Jouxtel, P. (2005). Le Pommier, Paris. 3

- [Kabán and Girolami, 2002] Kabán, A. and Girolami, M. A. (2002). A dynamic probabilistic model to visualise topic evolution in text streams. *J. Intell. Inf. Syst.*, 18(2-3) :107–125. 105
- [Kalman, 1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D) :35–45. 47
- [Kargupta et al., 2004] Kargupta, H., Bhargava, R., Liu, K., Powers, M., Blair, P., Bushra, S., Dull, J., Sarkar, K., Klein, M., Vasa, M., and Handy, D. (2004). VEDAS : A mobile and distributed data stream mining system for real-time vehicle monitoring. In *Proceedings of the Fourth SIAM International Conference on Data Mining, Lake Buena Vista, Florida, USA, April 22-24, 2004*, pages 300–311. 10
- [Karlsson et al., 2013] Karlsson, S. et al. (2013). Forecasting with bayesian vector autoregressions. *Handbook of Economic Forecasting*, 2(Part B) :791–897. 44
- [Kaufmann et al., 2012] Kaufmann, E., Korda, N., and Munos, R. (2012). Thompson sampling : An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory - 23rd International Conference, ALT 2012, Lyon, France, October 29-31, 2012. Proceedings*, pages 199–213. 14
- [Kazerouni et al., 2016] Kazerouni, A., Ghavamzadeh, M., Abbasi-Yadkori, Y., and Van Roy, B. (2016). Conservative Contextual Linear Bandits. *ArXiv e-prints*. 23
- [Kefato et al., 2018] Kefato, Z. T., Sheikh, N., Bahri, L., Soliman, A., Montresor, A., and Girdzijauskas, S. (2018). Cas2vec : Network-agnostic cascade prediction in online social networks. In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 72–79. 69
- [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. 59, 60, 61, 97
- [Kenter et al., 2015] Kenter, T., Wevers, M., Huijnen, P., and de Rijke, M. (2015). Ad hoc monitoring of vocabulary shifts over time. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 1191–1200. 105
- [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. 58
- [Kimura and Saito, 2006] Kimura, M. and Saito, K. (2006). Tractable models for information diffusion in social networks. In *European conference on principles of data mining and knowledge discovery*, pages 259–271. Springer. 60
- [Kingma and Ba, 2014] Kingma, D. P. and Ba, J. (2014). Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*. 52, 83
- [Kingma et al., 2016] Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751. 119
- [Kingma and Welling, 2013] Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *CoRR*, abs/1312.6114. 52, 82, 109, 119
- [Kirkby, 2007] Kirkby, R. B. (2007). *Improving hoeffding trees*. PhD thesis, The University of Waikato. 10
- [Kleinberg et al., 2008] Kleinberg, R. D., Niculescu-mizil, A., and Sharma, Y. (2008). Regret bounds for sleeping experts and bandits. In *In 21st COLT*, pages 425–436. 16

- [Kohli et al., 2013] Kohli, P., Salek, M., and Stoddard, G. (2013). A fast bandit algorithm for recommendation to users with heterogenous tastes. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA*. 15
- [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. 61
- [Koren et al., 2009] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8) :30–37. 88
- [Kossinets, 2006] Kossinets, G. (2006). Effects of missing data in social networks. *Social networks*, 28(3) :247–268. 88
- [Krishnamurthy and Wahlberg, 2009] Krishnamurthy, V. and Wahlberg, B. (2009). Partially observed markov decision process multiarmed bandits—structural results. *Mathematics of Operations Research*, 34(2) :287–302. 55
- [Krishnan et al., 2017] Krishnan, R. G., Shalit, U., and Sontag, D. (2017). Structured inference networks for nonlinear state space models. In *AAAI*. 52, 91, 109, 115
- [Kruskal, 1964] Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a non-metric hypothesis. *Psychometrika*, 29(1) :1–27. 69
- [Kuhn, 1988] Kuhn, R. (1988). Speech recognition and the frequency of recently used words : A modified markov model for natural language. In *Proceedings of the 12th conference on Computational linguistics-Volume 1*, pages 348–350. Association for Computational Linguistics. 124
- [Kulis and Grauman, 2009] Kulis, B. and Grauman, K. (2009). Kernelized locality-sensitive hashing for scalable image search. In *IEEE International Conference on Computer Vision (ICCV)*. 10
- [Kumar et al., 2016] Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., Gulrajani, I., Zhong, V., Paulus, R., and Socher, R. (2016). Ask me anything : Dynamic memory networks for natural language processing. In *International conference on machine learning*, pages 1378–1387. 125
- [Kusner et al., 2017] Kusner, M. J., Loftus, J., Russell, C., and Silva, R. (2017). Counterfactual fairness. In *Advances in Neural Information Processing Systems*, pages 4066–4076. 88
- [Kveton et al., 2015] Kveton, B., Wen, Z., Ashkan, A., and Szepesvari, C. (2015). Combinatorial cascading bandits. In *Advances in Neural Information Processing Systems*, pages 1450–1458. 96
- [Lage et al., 2013] Lage, R., Denoyer, L., Gallinari, P., and Dolog, P. (2013). Choosing which message to publish on social networks : a contextual bandit approach. In *Advances in Social Networks Analysis and Mining 2013, ASONAM '13, Niagara, ON, Canada - August 25 - 29, 2013*, pages 620–627. 15
- [Lagnier et al., 2014] Lagnier, C., Bourigault, S., Lamprier, S., Denoyer, L., and Gallinari, P. (2014). Learning information spread in content networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.
- [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. 60
- [Lagnier et al., 2018] Lagnier, C., Gaussier, E., and Kawala, F. (2018). User-centered probabilistic models for content diffusion in the blogosphere. *Online Social Networks and Media*, 5 :61–75. 60
- [Lagrée et al., 2017] Lagrée, P., Cappé, O., Cautis, B., and Maniu, S. (2017). Effective large-scale online influence maximization. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 937–942. IEEE. 97

- [Lai and Robbins, 1985] Lai, T. and Robbins, H. (1985). Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1) :4 – 22. 14
- [Lamb et al., 2016] Lamb, A. M., Goyal, A. G. A. P., Zhang, Y., Zhang, S., Courville, A. C., and Bengio, Y. (2016). Professor forcing : A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pages 4601–4609. 94
- [Lample et al., 2018] Lample, G., Ott, M., Conneau, A., Denoyer, L., and Ranzato, M. (2018). Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5039–5049. 104
- [Lamprier, 2019] Lamprier, S. (2019). A recurrent neural cascade-based model for continuous-time diffusion. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 3632–3641. 78, 79, 81, 82, 83, 84, 91
- [Lamprier et al., 2013] Lamprier, S., Baskiotis, N., Ziadi, T., and Hillah, L. (2013). CARE : A platform for reliable comparison and analysis of reverse-engineering techniques. In *2013 18th International Conference on Engineering of Complex Computer Systems, Singapore, July 17-19, 2013*, pages 252–255. 127
- [Lamprier et al., 2015a] Lamprier, S., Baskiotis, N., Ziadi, T., and Hillah, L. (2015a). The CARE platform for the analysis of behavior model inference techniques. *Information & Software Technology*, 60 :32–50. 127
- [Lamprier et al., 2015b] Lamprier, S., Bourigault, S., and Gallinari, P. (2015b). 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, ASONAM 2015, Paris, France, August 25 - 28, 2015*, pages 178–185.
- [Lamprier et al., 2016] Lamprier, S., Bourigault, S., and Gallinari, P. (2016). Influence learning for cascade diffusion models : focus on partial orders of infections. *Social Netw. Analys. Mining*, 6(1) :93 :1–93 :16. 60, 64, 66, 67, 68
- [Lamprier et al., 2017] Lamprier, S., Gisselbrecht, T., and Gallinari, P. (2017). Variational thompson sampling for relational recurrent bandits. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part II*, pages 405–421. 45, 47
- [Lamprier et al., 2018] Lamprier, S., Gisselbrecht, T., and Gallinari, P. (2018). Profile-based bandit with unknown profiles. *Journal of Machine Learning Research*, 19 :53 :1–53 :40. 25, 29, 31
- [Lamprier et al., 2019] Lamprier, S., Gisselbrecht, T., and Gallinari, P. (2019). Contextual bandits with hidden contexts : a focused data capture from social media streams. *Data Min. Knowl. Discov.*, 33(6) :1853–1893. 37, 40
- [Lamprier et al., 2014] Lamprier, S., Ziadi, T., Baskiotis, N., and Hillah, L. (2014). Exact and efficient temporal steering of software behavioral model inference. In *2014 19th International Conference on Engineering of Complex Computer Systems, Tianjin, China, August 4-7, 2014*, pages 166–175. 127
- [Le and Mikolov, 2014] Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196. 69, 107
- [Lee, 2012] Lee, Y. (2012). Spherical hashing. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR '12*, pages 2957–2964, Washington, DC, USA. IEEE Computer Society. 10

- [Lei et al., 2015] Lei, S., Maniu, S., Mo, L., Cheng, R., and Senellart, P. (2015). Online influence maximization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 645–654. 97
- [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*, pages 497–506. 90
- [Levenberg and Osborne, 2009] Levenberg, A. and Osborne, M. (2009). Stream-based randomised language models for smt. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 756–764. 124
- [Levine et al., 2017] Levine, N., Crammer, K., and Mannor, S. (2017). Rotting bandits. 54
- [Li et al., 2019] Li, B., He, J., Neubig, G., Berg-Kirkpatrick, T., and Yang, Y. (2019). A surprisingly effective fix for deep latent variable modeling of text. *arXiv preprint arXiv :1909.00868*. 119
- [Li et al., 2016] Li, C., Ma, J., Guo, X., and Mei, Q. (2016). Deepcas : an end-to-end predictor of information cascades. *CoRR*, abs/1611.05373. 78
- [Li et al., 2017] Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., and Jurafsky, D. (2017). Adversarial learning for neural dialogue generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2157–2169. 95
- [Li et al., 2011] Li, L., Chu, W., Langford, J., and Wang, X. (2011). Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 297–306. 20, 21
- [Li et al., 2017] Li, L., Lu, Y., and Zhou, D. (2017). Provably Optimal Algorithms for Generalized Linear Contextual Bandits. *ArXiv e-prints*. 23
- [Li et al., 2013] Li, R., Wang, S., and Chang, K. C. (2013). Towards social data platform : Automatic topic-focused monitor for twitter stream. *PVLDB*, 6(14) :1966–1977. 11
- [Li et al., 2018] Li, S., Xiao, S., Zhu, S., Du, N., Xie, Y., and Song, L. (2018). Learning temporal point processes via reinforcement learning. In *Advances in neural information processing systems*, pages 10781–10791. 96
- [Li et al., 2015] Li, Y., Hernández-Lobato, J. M., and Turner, R. E. (2015). Stochastic expectation propagation. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 2323–2331. Curran Associates, Inc. 50, 52
- [Lin et al., 2011] Lin, J., Snow, R., and Morgan, W. (2011). Smoothing techniques for adaptive online language models : topic tracking in tweet streams. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 422–429. 124
- [Liu et al., 2016] Liu, H., Morstatter, F., Tang, J., and Zafarani, R. (2016). The good, the bad, and the ugly : uncovering novel research opportunities in social media mining. *International Journal of Data Science and Analytics*, 1(3-4) :137–143. 4
- [Lokhov, 2016] Lokhov, A. (2016). Reconstructing parameters of spreading models from partial observations. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 3467–3475. Curran Associates, Inc. 89
- [Luo et al., 2018] Luo, Y., Cai, X., Zhang, Y., Xu, J., et al. (2018). Multivariate time series imputation with generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 1596–1607. 95

- [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*. 59
- [Ma et al., 2018] Ma, H., King, I., and Lyu, M. R. (2018). Social recommendation in dynamic networks. In Alhajj, R. and Rokne, J. G., editors, *Encyclopedia of Social Network Analysis and Mining, 2nd Edition*. Springer. 69
- [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. 59, 61
- [Maddison et al., 2016] Maddison, C. J., Mnih, A., and Teh, Y. W. (2016). The concrete distribution : A continuous relaxation of discrete random variables. *CoRR*, abs/1611.00712. 82
- [Makhzani et al., 2015] Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*. 119
- [Mehmood et al., 2013] Mehmood, Y., Barbieri, N., Bonchi, F., and Ukkonen, A. (2013). Csi : Community-level social influence analysis. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 48–63. Springer. 59
- [Melis et al., 2018] Melis, G., Dyer, C., and Blunsom, P. (2018). On the state-of-the-art of evaluation in neural language models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 104, 106
- [Merity et al., 2018a] Merity, S., Keskar, N. S., and Socher, R. (2018a). An analysis of neural language modeling at multiple scales. *CoRR*, abs/1803.08240. 104
- [Merity et al., 2018b] Merity, S., Keskar, N. S., and Socher, R. (2018b). Regularizing and optimizing LSTM language models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 104, 106, 110
- [Merity et al., 2016] Merity, S., Xiong, C., Bradbury, J., and Socher, R. (2016). Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*. 125
- [Micarelli and Gasparetti, 2007] Micarelli, A. and Gasparetti, F. (2007). Adaptive focused crawling. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 231–262. Springer Berlin Heidelberg. 9
- [Mikolov et al., 2013] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781. 69, 104, 105
- [Mikolov et al., 2010] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048. 76, 106
- [Mikolov and Zweig, 2012] Mikolov, T. and Zweig, G. (2012). Context dependent recurrent neural network language model. In *2012 IEEE Spoken Language Technology Workshop (SLT), Miami, FL, USA, December 2-5, 2012*, pages 234–239. 76
- [Milli et al., 2018] Milli, L., Rossetti, G., Pedreschi, D., and Giannotti, F. (2018). Active and passive diffusion processes in complex networks. *Applied network science*, 3(1) :42. 90
- [Minka, 2001] Minka, T. P. (2001). *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Cambridge, MA, USA. AAI0803033. 49, 50

- [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. 71
- [Murphy and Russell, 2002] Murphy, K. P. and Russell, S. (2002). Dynamic bayesian networks : representation, inference and learning. 86
- [Mustar et al., 2020] Mustar, A., Lamprier, S., and Piwowarski, B. (2020). On the study of transformers for query suggestion. In *soumis à TOIS 2020*. 128
- [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 on Data Mining (ICDM)*, pages 539–548. IEEE. 58
- [Nachum et al., 2018] Nachum, O., Gu, S., Lee, H., and Levine, S. (2018). Near-optimal representation learning for hierarchical reinforcement learning. *arXiv preprint arXiv :1810.01257*. 69
- [Nagabandi et al., 2018] Nagabandi, A., Finn, C., and Levine, S. (2018). Deep online learning via meta-learning : Continual adaptation for model-based rl. *arXiv preprint arXiv :1812.07671*. 129
- [Najar et al., 2012] Najar, A., Denoyer, L., and Gallinari, P. (2012). Predicting information diffusion on social networks with partial knowledge. In Mille, A., Gandon, F. L., Misselis, J., Rabinovich, M., and Staab, S., editors, *Proceedings of the 21st World Wide Web Conference, WWW 2012, Lyon, France, April 16-20, 2012 (Companion Volume)*, pages 1197–1204. ACM. 59, 62, 85, 88
- [Narasimhan et al., 2015] Narasimhan, H., Parkes, D. C., and Singer, Y. (2015). Learnability of influence in networks. In *Advances in Neural Information Processing Systems*, pages 3186–3194. 85, 89, 99
- [Neal, 2012] Neal, R. M. (2012). *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media. 52
- [Nemhauser et al., 1978] Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. (1978). An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1) :265–294. 97
- [Netrapalli and Sanghavi, 2012] Netrapalli, P. and Sanghavi, S. (2012). Learning the graph of epidemic cascades. *ACM SIGMETRICS Performance Evaluation Review*, 40(1) :211–222. 73, 98
- [Nguyen et al., 2018] Nguyen, G. H., Lee, J. B., Rossi, R. A., Ahmed, N. K., Koh, E., and Kim, S. (2018). Continuous-time dynamic network embeddings. In *Companion Proceedings of the The Web Conference 2018, WWW '18*, pages 969–976, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee. 78
- [Nodelman et al., 2012] Nodelman, U., Koller, D., and Shelton, C. R. (2012). Expectation propagation for continuous time bayesian networks. *CoRR*, abs/1207.1401. 86
- [Nodelman et al., 2002] Nodelman, U., Shelton, C. R., and Koller, D. (2002). Continuous time bayesian networks. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 378–387. 86
- [Nuara et al., 2018] Nuara, A., Trovo, F., Gatti, N., and Restelli, M. (2018). A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns. In *Thirty-Second AAAI Conference on Artificial Intelligence*. 15
- [Oppen and Winther, 1998] Oppen, M. and Winther, O. (1998). A bayesian approach to on-line learning. *On-line learning in neural networks*, pages 363–378. 50

- [Opper and Winther, 2000] Opper, M. and Winther, O. (2000). Gaussian processes for classification : Mean-field algorithms. *Neural computation*, 12(11) :2655–2684. 49
- [Ortner et al., 2014] Ortner, R., Ryabko, D., Auer, P., and Munos, R. (2014). Regret bounds for restless markov bandits. *Theor. Comput. Sci.*, 558 :62–76. 43
- [Osband et al., 2013] Osband, I., Russo, D., and Van Roy, B. (2013). (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pages 3003–3011. 55
- [Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking : Bringing order to the web. Technical report, Stanford InfoLab. 56
- [Papadimitriou et al., 2003] Papadimitriou, S., Brockwell, A., and Faloutsos, C. (2003). Adaptive, hands-off stream mining. In *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29, VLDB '03*, pages 560–571. VLDB Endowment. 10
- [Parisi et al., 2018] Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2018). Continual lifelong learning with neural networks : A review. *CoRR*, abs/1802.07569. 129
- [Pascanu et al., 2013] Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 1310–1318. 76
- [Paulus et al., 2017] Paulus, R., Xiong, C., and Socher, R. (2017). A deep reinforced model for abstractive summarization. *arXiv preprint arXiv :1705.04304*. 95
- [Pearl, 1986] Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial intelligence*, 29(3) :241–288. 50
- [Pearl, 2009] Pearl, J. (2009). *Causality*. Cambridge university press. 88
- [Pearl, 2012] Pearl, J. (2012). The do-calculus revisited. *arXiv preprint arXiv :1210.4852*. 87
- [Perozzi et al., 2014] Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk : Online learning of social representations. *CoRR*, abs/1403.6652. 78
- [Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. 105
- [Petrović et al., 2010] Petrović, S., Osborne, M., and Lavrenko, V. (2010). Streaming first story detection with application to twitter. In *Human language technologies : The 2010 annual conference of the north american chapter of the association for computational linguistics*, pages 181–189. 124
- [Pike-Burke and Grunewalder, 2019] Pike-Burke, C. and Grunewalder, S. (2019). Recovering bandits. In *Advances in Neural Information Processing Systems*, pages 14122–14131. 54, 55
- [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. 74
- [Qin et al., 2014] Qin, L., Chen, S., and Zhu, X. (2014). Contextual combinatorial bandit and its application on diversified online recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining, Philadelphia, Pennsylvania, USA, April 24-26, 2014*, pages 461–469. 23

- [Radford et al., 2016] Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*. 95, 123
- [Radford et al., 2019] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. 106
- [Ranganath et al., 2014] Ranganath, R., Gerrish, S., and Blei, D. M. (2014). Black Box Variational Inference. *ArXiv e-prints*. 82
- [Ranzato et al., 2015] Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. (2015). Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*. 94, 95
- [Rasmussen, 2003] Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer. 52
- [Ratliff et al., 2018] Ratliff, L. J., Sekar, S., Zheng, L., and Fiez, T. (2018). Incentives in the dark : multi-armed bandits for evolving users with unknown type. *arXiv preprint arXiv:1803.04008*. 55
- [Reichenbach, 1991] Reichenbach, H. (1991). *The direction of time*, volume 65. Univ of California Press. 87
- [Ren et al., 2020] Ren, Y., Guo, S., Labeau, M., Cohen, S. B., and Kirby, S. (2020). Compositional languages emerge in a neural iterated learning model. *arXiv preprint arXiv:2002.01365*. 105
- [Ren et al., 2019] Ren, Z., Dong, K., Zhou, Y., Liu, Q., and Peng, J. (2019). Exploration via hindsight goal generation. In *Advances in Neural Information Processing Systems*, pages 13464–13474. 55
- [Rezende et al., 2014] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *ICML*. 52, 109
- [Riquelme et al., 2018] Riquelme, C., Tucker, G., and Snoek, J. (2018). Deep bayesian bandits showdown : An empirical comparison of bayesian deep networks for thompson sampling. In *International Conference on Learning Representations*. 49, 52
- [Rosenbaum and Rubin, 1983] Rosenbaum, P. R. and Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1) :41–55. 87
- [Rosenfeld and Erk, 2018] Rosenfeld, A. and Erk, K. (2018). Deep neural models of semantic shift. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 474–484. 106, 110
- [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*. 61
- [Rudolph and Blei, 2017] Rudolph, M. R. and Blei, D. M. (2017). Dynamic bernoulli embeddings for language evolution. *CoRR*, abs/1703.08052. 106
- [Rudolph et al., 2017] Rudolph, M. R., Ruiz, F. J. R., Athey, S., and Blei, D. M. (2017). Structured embedding models for grouped data. In *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 251–261. 106
- [Rusu et al., 2016] Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671*. 129

- [Ryan and Gross, 1950] Ryan, B. and Gross, N. (1950). Acceptance and diffusion of hybrid corn seed in two iowa communities. *Iowa Agriculture and Home Economics Experiment Station Research Bulletin*, 29(372) :1. 3
- [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. 60, 63, 79, 80, 93
- [Saito et al., 2010] Saito, K., Kimura, M., Ohara, K., and Motoda, H. (2010). Generative models of information diffusion with asynchronous timedelay. *Journal of Machine Learning Research - Proceedings Track*, 13 :193–208. 60
- [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. 62, 63, 64, 65, 72
- [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. 60
- [Salas et al., 2018] Salas, A., Kessler, S., Zohren, S., and Roberts, S. (2018). Practical bayesian learning of neural networks via adaptive subgradient methods. *arXiv preprint arXiv :1811.03679*. 52
- [Sallak et al., 2013] Sallak, M., Aguirre, E., and Schon, W. (2013). Incertitudes aléatoires et épistémiques, comment les distinguer et les manipuler dans les études de fiabilité? In *QUALITA2013*, Compiègne, France. 127
- [Schölkopf, 2019] Schölkopf, B. (2019). Causality for machine learning. 86, 87
- [Scialom et al., 2020a] Scialom, T., Dray, P., Lamprier, S., Piwowarski, B., and Staiano, J. (2020a). Cold-gans : Taming language gans with cautious sampling strategies. In Laroche, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems 33 : Annual Conference on Neural Information Processing Systems 2020, NeurIPS*. 96, 127, 128
- [Scialom et al., 2020b] Scialom, T., Dray, P., Lamprier, S., Piwowarski, B., and Staiano, J. (2020b). Discriminative adversarial search for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML*. 95, 127
- [Scialom et al., 2019] Scialom, T., Lamprier, S., Piwowarski, B., and Staiano, J. (2019). Answers unite! unsupervised metrics for reinforced summarization models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3244–3254. 95, 127
- [Semeniuta et al., 2018] Semeniuta, S., Severyn, A., and Gelly, S. (2018). On accurate evaluation of gans for language generation. *arXiv preprint arXiv :1806.04936*. 95
- [Serban et al., 2017a] Serban, I. V., II, A. G. O., Pineau, J., and Courville, A. C. (2017a). Piecewise latent variables for neural variational text processing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 422–432. 107

- [Serban et al., 2017b] Serban, I. V., Sankar, C., Germain, M., Zhang, S., Lin, Z., Subramanian, S., Kim, T., Pieper, M., Chandar, S., Ke, N. R., Mudumba, S., de Brébisson, A., Sotelo, J., Suhubdy, D., Michalski, V., Nguyen, A., Pineau, J., and Bengio, Y. (2017b). A deep reinforcement learning chatbot. *CoRR*, abs/1709.02349. 122
- [Shaghaghian and Coates, 2016] Shaghaghian, S. and Coates, M. (2016). Bayesian inference of diffusion networks with unknown infection times. In *2016 IEEE Statistical Signal Processing Workshop (SSP)*, pages 1–5. IEEE. 89
- [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. 59
- [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. 74
- [Shen et al., 2019] Shen, T., Mueller, J., Barzilay, R., and Jaakkola, T. (2019). Educating text autoencoders : Latent representation guidance via denoising. *arXiv preprint arXiv :1905.12777*. 119
- [Shi et al., 2015] Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-k., and Woo, W.-c. (2015). Convolutional LSTM network : A machine learning approach for precipitation nowcasting. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 802–810. Curran Associates, Inc. 76
- [Shi et al., 2018] Shi, Y., Lei, M., Zhang, P., and Niu, L. (2018). Diffusion based network embedding. *CoRR*, abs/1805.03504. 78
- [Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*. 123
- [Slivkins, 2009] Slivkins, A. (2009). Contextual bandits with similarity information. *CoRR*, abs/0907.3986. 23
- [Slivkins and Upfal, 2008] Slivkins, A. and Upfal, E. (2008). Adapting to a changing environment : the brownian restless bandits. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, pages 343–354. 43
- [Snoek et al., 2015] Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., and Adams, R. (2015). Scalable bayesian optimization using deep neural networks. In *International conference on machine learning*, pages 2171–2180. 51
- [Song and Croft, 1999] Song, F. and Croft, W. B. (1999). A general language model for information retrieval. In *Proceedings of the 1999 ACM CIKM International Conference on Information and Knowledge Management, Kansas City, Missouri, USA, November 2-6, 1999*, pages 316–321. 104
- [Spaan and Lima, 2009] Spaan, M. T. J. and Lima, P. U. (2009). A decision-theoretic approach to dynamic sensor selection in camera networks. In *Int. Conf. on Automated Planning and Scheduling*, pages 279–304. 12
- [Su and Khoshgoftaar, 2009] Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009. 69
- [Sukhbaatar et al., 2018] Sukhbaatar, S., Denton, E., Szlam, A., and Fergus, R. (2018). Learning goal embeddings via self-play for hierarchical reinforcement learning. *CoRR*, abs/1811.09083. 128
- [Sukhbaatar et al., 2017] Sukhbaatar, S., Kostrikov, I., Szlam, A., and Fergus, R. (2017). Intrinsic motivation and automatic curricula via asymmetric self-play. *CoRR*, abs/1703.05407. 128

- [Sukhbaatar et al., 2015] Sukhbaatar, S., Weston, J., Fergus, R., et al. (2015). End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448. 125
- [Sutskever et al., 2011] Sutskever, I., Martens, J., and Hinton, G. E. (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pages 1017–1024. 76
- [Tai et al., 2015] Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *CoRR*, abs/1503.00075. 77
- [Tan and Lee, 2015] Tan, C. and Lee, L. (2015). All who wander : On the prevalence and characteristics of multi-community engagement. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 1056–1066. 110
- [Tarde, 1890] Tarde, G. (1890). *Les lois de l'imitation : étude sociologique*. Félix Alcan, Paris. 3
- [Taxidou, 2013] Taxidou, I. (2013). Realtime analysis of information diffusion in social media. *PVLDB*, 6(12) :1416–1421. 12
- [Taxidou and Fischer, 2014a] Taxidou, I. and Fischer, P. M. (2014a). Online analysis of information diffusion in twitter. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 1313–1318. 100
- [Taxidou and Fischer, 2014b] Taxidou, I. and Fischer, P. M. (2014b). Rapid : A system for real-time analysis of information diffusion in twitter. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*, pages 2060–2062. 12
- [Tekin and Liu, 2012] Tekin, C. and Liu, M. (2012). Online learning of rested and restless bandits. *IEEE Trans. Information Theory*, 58(8) :5588–5611. 43, 54
- [Thompson, 1933] Thompson, W. (1933). On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Bulletin of the American Mathematics Society*, 25 :285–294. 14
- [Tshimula et al., 2019] Tshimula, J. M., Chikhaoui, B., and Wang, S. (2019). Har-search : a method to discover hidden affinity relationships in online communities. In *2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 176–183. IEEE. 122
- [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. 85
- [Tzen and Raginsky, 2019] Tzen, B. and Raginsky, M. (2019). Neural stochastic differential equations : Deep latent gaussian models in the diffusion limit. *CoRR*, abs/1905.09883. 118
- [Ullah and Lee, 2016] Ullah, F. and Lee, S. (2016). Social content recommendation based on spatial-temporal aware diffusion modeling in social networks. *Symmetry*, 8(9) :89. 60
- [Valko et al., 2013] Valko, M., Korda, N., Munos, R., Flaounas, I. N., and Cristianini, N. (2013). Finite-time analysis of kernelised contextual bandits. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, UAI 2013, Bellevue, WA, USA, August 11-15, 2013*. 51
- [Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008. 77, 97, 104, 106

- [Vaswani and Lakshmanan, 2015] Vaswani, S. and Lakshmanan, L. V. S. (2015). Influence maximization with bandits. *CoRR*, abs/1503.00024. 15, 98
- [Venkatraman et al., 2015] Venkatraman, A., Hebert, M., and Bagnell, J. A. (2015). Improving multi-step prediction of learned time series models. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*. 94
- [Ver Steeg and Galstyan, 2013] Ver Steeg, G. and Galstyan, A. (2013). Information-theoretic measures of influence based on content dynamics. In *Proceedings of the sixth ACM international conference on Web search and data mining*, WSDM '13, pages 3–12, New York, NY, USA. ACM. 62
- [Vinyals et al., 2017] Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2017). Show and tell : Lessons learned from the 2015 MSCOCO image captioning challenge. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4) :652–663. 104
- [Wang et al., 2012a] Wang, C., Blei, D. M., and Heckerman, D. (2012a). Continuous time dynamic topic models. *CoRR*, abs/1206.3298. 105
- [Wang et al., 2011] Wang, E., Silva, J., Willett, R., and Carin, L. (2011). Dynamic relational topic model for social network analysis with noisy links. In *2011 IEEE Statistical Signal Processing Workshop (SSP)*, pages 497–500. 105
- [Wang et al., 2012b] Wang, H., Can, D., Kazemzadeh, A., Bar, F., and Narayanan, S. (2012b). A system for real-time twitter sentiment analysis of 2012 U.S. presidential election cycle. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea*, pages 115–120. 11
- [Wang et al., 2016] Wang, H., Wu, Q., and Wang, H. (2016). Learning hidden features for contextual bandits. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 1633–1642, New York, NY, USA. ACM. 24, 51
- [Wang et al., 2017a] Wang, J., Zheng, V. W., Liu, Z., and Chang, K. C. (2017a). Topological recurrent neural network for diffusion prediction. *CoRR*, abs/1711.10162. 77, 92
- [Wang et al., 2012c] Wang, L., Ermon, S., and Hopcroft, J. E. (2012c). Feature-enhanced probabilistic models for diffusion network inference. In *Proceedings of the 2012 European conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*, ECML PKDD'12, pages 499–514. Springer-Verlag. 60
- [Wang et al., 2019] Wang, W., Gan, Z., Xu, H., Zhang, R., Wang, G., Shen, D., Chen, C., and Carin, L. (2019). Topic-guided variational auto-encoder for text generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 166–177, Minneapolis, Minnesota. Association for Computational Linguistics. 120
- [Wang and McCallum, 2006] Wang, X. and McCallum, A. (2006). Topics over time : a non-markov continuous-time model of topical trends. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 424–433. 105
- [Wang et al., 2017b] Wang, Y., Long, M., Wang, J., Gao, Z., and Yu, P. S. (2017b). Predrnn : Recurrent neural networks for predictive learning using spatiotemporal lstms. In *Advances in Neural Information Processing Systems 30 : Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 879–888. 76

- [Wang et al., 2017c] Wang, Y., Shen, H., Liu, S., Gao, J., and Cheng, X. (2017c). Cascade dynamics modeling with attention-based recurrent neural network. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2985–2991. 77, 83
- [Wang et al., 2018] Wang, Z., Chen, C., and Li, W. (2018). Attention network for information diffusion prediction. In *Companion Proceedings of the The Web Conference 2018, WWW '18*, pages 65–66, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee. 77, 84
- [Weiss et al., 2012] Weiss, Y., Fergus, R., and Torralba, A. (2012). Multidimensional spectral hashing. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part V, ECCV'12*, pages 340–353, Berlin, Heidelberg. Springer-Verlag. 10
- [Wen et al., 2017] Wen, Z., Kveton, B., Valko, M., and Vaswani, S. (2017). Online influence maximization under independent cascade model with semi-bandit feedback. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 3022–3032. Curran Associates, Inc. 97
- [Whittle, 1988] Whittle, P. (1988). Restless bandits : Activity allocation in a changing world. *Journal of Applied Probability*, 25 :287–298. 43
- [Wierstra et al., 2010] Wierstra, D., Förster, A., Peters, J., and Schmidhuber, J. (2010). Recurrent policy gradients. *Logic Journal of the IGPL*, 18(5) :620–634. 76
- [Williams and Zipser, 1989] Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2) :270–280. 94
- [Williamson, 2016] Williamson, S. A. (2016). Nonparametric network models for link prediction. *Journal of Machine Learning Research*, 17(202) :1–21. 99, 100
- [Winn and Bishop, 2005] Winn, J. M. and Bishop, C. M. (2005). Variational message passing. *Journal of Machine Learning Research*, 6 :661–694. 36
- [Wiseman and Rush, 2016] Wiseman, S. and Rush, A. M. (2016). Sequence-to-sequence learning as beam-search optimization. *arXiv preprint arXiv :1606.02960*. 94
- [Wu et al., 2020] Wu, C., Wang, P. Z., and Wang, W. Y. (2020). On the encoder-decoder incompatibility in variational text modeling and beyond. *arXiv preprint arXiv :2004.09189*. 119
- [Wu et al., 2013] Wu, X., Kumar, A., Sheldon, D., and Zilberstein, S. (2013). Parameter learning for latent network diffusion. In *Twenty-Third International Joint Conference on Artificial Intelligence*. 88, 89, 90
- [Xiao et al., 2018] Xiao, Y., Zhao, T., and Wang, W. Y. (2018). Dirichlet variational autoencoder for text modeling. *arXiv preprint arXiv :1811.00135*. 120
- [Xu et al., 2019] Xu, P., Cheung, J. C. K., and Cao, Y. (2019). On variational learning of controllable representations for text without supervision. *arXiv preprint arXiv :1905.11975*. 120, 121
- [Yang et al., 2019] Yang, C., Tang, J., Sun, M., Cui, G., and Liu, Z. (2019). Multi-scale information diffusion prediction with reinforced recurrent networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4033–4039. AAAI Press. 77, 95
- [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. 62, 85

- [Yang et al., 2014] Yang, J., McAuley, J., and Leskovec, J. (2014). Detecting cohesive and 2-mode communities in directed 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. 71
- [Yang et al., 2017] Yang, Z., Hu, Z., Salakhutdinov, R., and Berg-Kirkpatrick, T. (2017). Improved variational autoencoders for text modeling using dilated convolutions. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3881–3890. JMLR. org. 119
- [Yao et al., 2018] Yao, Z., Sun, Y., Ding, W., Rao, N., and Xiong, H. (2018). Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 673–681. 105, 106, 110
- [Zaheer et al., 2017] Zaheer, M., Ahmed, A., and Smola, A. J. (2017). Latent lstm allocation joint clustering and non-linear dynamic modeling of sequential data. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pages 3967–3976. JMLR.org. 107, 120
- [Zellers et al., 2019] Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., and Choi, Y. (2019). Defending against neural fake news. In *Advances in Neural Information Processing Systems*, pages 9051–9062. 95
- [Zhang et al., 2019] Zhang, W., Feng, Y., Meng, F., You, D., and Liu, Q. (2019). Bridging the gap between training and inference for neural machine translation. *arXiv preprint arXiv:1906.02448*. 94
- [Zhang et al., 2018] Zhang, Y., Lyu, T., and Zhang, Y. (2018). Cosine : Community-preserving social network embedding from information diffusion cascades. In *AAAI*. 79
- [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. 60
- [Zhong et al., 2013] Zhong, W., Raahemi, B., and Liu, J. (2013). Classifying peer-to-peer applications using imbalanced concept-adapting very fast decision tree on IP data stream. *Peer-to-Peer Networking and Applications*, 6(3) :233–246. 10
- [Zhou et al., 2020] Zhou, W., Ge, T., Xu, K., Wei, F., and Zhou, M. (2020). Self-adversarial learning with comparative discrimination for text generation. In *International Conference on Learning Representations*. 95
- [Zhuo et al., 2019] Zhuo, W., Zhao, Y., Zhan, Q., and Liu, Y. (2019). DiffusionGAN : Network embedding for information diffusion prediction with generative adversarial nets. In *2019 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPAA/BDCloud/SocialCom/SustainCom)*, pages 808–816. 96
- [Zinkevich, 2003] Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936. 98
- [Zong et al., 2012] Zong, B., Wu, Y., Singh, A. K., and Yan, X. (2012). Inferring the underlying structure of information cascades. In *2012 IEEE 12th International Conference on Data Mining*, pages 1218–1223. IEEE. 90