A Recurrent Neural Cascade-based Model for Continuous-Time Diffusion

Sylvain Lamprier [1]

Abstract

Many works have been proposed in the literature to capture the dynamics of diffusion in networks. While some of them define graphical Markovian models to extract temporal relationships between node infections in networks, others consider diffusion episodes as sequences of infections via recurrent neural models. In this paper we propose a model at the crossroads of these two extremes, which embeds the history of diffusion in infected nodes as hidden continuous states. Depending on the trajectory followed by the content before reaching a given node, the distribution of influence probabilities may vary. However, content trajectories are usually hidden in the data, which induces challenging learning problems. We propose a topological recurrent neural model which exhibits good experimental performances for diffusion modelling and prediction.

1. Introduction

The recent development of online social networks enabled researchers to suggest methods to explain and predict observations of diffusion across networks. A diffusion process refers to the temporal spread of related events on a network. These events can correspond to the publication of a given video, the listening of a specific music album, the adoption of a given behavior or language, or even the infection by a given disease. Classical cascade models, which are at the heart of the research literature on diffusion, regard the phenomenon of diffusion as an iterative process in which a content transits from nodes to others in the network (Saito et al., 2008; Gomez-Rodriguez et al., 2011), by a so-called word-of-mouth phenomenon. In this setting, diffusion modeling corresponds to learning probability distributions of content transmission. Various cascade models have been proposed in the literature, each inducing its own learning

process to extract the main dynamics of the network from observed diffusion episodes.

However, most of these models rely on a strong Markovian assumption for which the probabilities of next infections[1] only depend on who is infected at the current step, not on the past trajectories of the diffused content. We claim that the history of spread contains much valuable information that should be taken into account by the models. In many applications the object of the diffusion is either unknown (due to privacy or technical limitations for instances) or its available features are not exploitable by the models (low correlation with the diffusion, high noise, high variability, etc.). In such cases, which is the setting of our work, past trajectories can give useful insights about the nature of the diffusion. Also, the content may be changed during the diffusion, with different transformations depending on which nodes re-transmit the information.

Some recent approaches rely on representation learning and recurrent neural networks (RNN) to predict the future spread of diffusion given the past. A naive possibility would be to consider diffusion episodes as sequences of infections and propose temporal point process approaches to model the dynamics. Using the Recurrent Marked Temporal Point Process model (Du et al., 2016), the current hidden state of the RNN would embed the history of the whole diffusion sequence, which would be used to output the next infected node and its time of infection. However, diffusion episodes are not sequences but trees, naive recurrent methods usually fail in capturing the true dynamics of the networks. Embedding the whole past in the state of a given node rather than restraining it to its specific ancestor branch leads to consider many independent and noisy events for future predictions. A model that would consider the true diffusion paths would be more effective, by focusing on the useful past. If the true diffusion paths were known, it would be possible to adapt works on recurrent neural models for tree structures such as successfully proposed in (Tai et al., 2015) for NLP tasks. Unfortunately, in most of applications the topology of diffusion is unknown while learning. In the task considered in this paper, the only observations available are the timestamps of the infected nodes.

---

[1]Sorbonne Universités, LIP6, F-75005, Paris, France. Correspondence to: Sylvain Lamprier <Sylvain.Lamprier@lip6.fr>.

---

[1]Throughout this paper, we refer to infection for denoting the participation of a node of the network in the diffusion.

To cope with it, (Wang et al., 2017a) proposed `Topo-LSTM`, a Long-Short Term Memory network that considers a known graph of relationships between nodes to compute hidden states and cells of infected nodes. The hidden state and cell of a given node at t depend on those from each of its predecessors that have been infected before t. Since nodes may have multiple predecessors that are infected at time t, the classical `LSTM` cannot be applied directly. Instead, (Wang et al., 2017a) proposed a cell function that aggregates infector candidate states via mean pooling. This allows to take the topology of the possible diffusion into account, but not the past trajectory of the content (it averages all possible paths). To overcome this, (Wang et al., 2017b) proposed a cascade attention-based RNN, which defines a neural attention mechanism to assign weights to infector candidates before summing their contribution in the new hidden state at t. The attention network is supposed to learn to identify from whom comes the next infection based on past states. However, such an approach is likely to converge to most of attention weight vectors in the center of the simplex, since diffusion is a stochastic process with mostly very weak influence probabilities. The deterministic inference process of the approach limits its ability to produce relevant states by mixing from multiple candidates rather than sampling the past trajectories from their posterior probabilities. Note the similar approach in (Wang et al., 2018), which does not use a RNN but defines a composition module of the past via attention networks to embed the episode in a representation space from which the probability of the next infected node can be deduced. Beyond the limits discussed above w.r.t. deterministic mixing of diffusion trajectories, no delay of infection is considered in this work, which makes it impossible to use for diffusion modeling purposes.

Recently, many works in representation learning used random walks on graphs to sample trajectories that can be used to learn a network embedding which respects some topological constraints. While `DeepWalk` (Perozzi et al., 2014) only uses structural information, models proposed in (Nguyen et al., 2018) or (Shi et al., 2018) include temporal constraints in random walks to sample feasible trajectories w.r.t. observed diffusion episodes. The approach `DeepCas` from (Li et al., 2016) applies this idea for the prediction of diffusion cascades. However, such approaches require a graph of diffusion relations as input, which is not always available (and not always representative of the true diffusion channels of the considered network as reported in (Ver Steeg & Galstyan, 2013)). In our work, we consider that no graph is available beforehand. Moreover, no inference process is introduced in `DeepCas` to sample trajectories from their posterior probabilities given the observed diffusion sequences. The sampling of trajectories is performed in an initialization step, before learning the parameters of the diffusion model.

In this paper, we propose the first bayesian topological recurrent neural network for sequences with tree dependencies, which we apply for diffusion cascades modelling. Rather than building on a preliminary random walk process, the idea is to consider trajectory inference during learning, in order to converge to better representations of the infected nodes. Following the stochastic nature of diffusion, the model infers trajectory distributions from observations of infections, which are in turn used for the inference of infection probabilities in an iterative learning process. Our probabilistic model, based on the famous continuous-time independent cascade model (`CTIC`) (Saito et al., 2009) is able to extract full paths of diffusion from sequential observations of infections via black-box inference, which has various applications in the field. Our experiments validate the approach for both modeling and prediction purposes.

Section 2 presents some background and notations of the approach. Section 3 presents the proposed model. Section 4 reports experimental results of the approach.

2. Background

2.1. Information Diffusion

Information diffusion is observed as a set of diffusion episodes $\mathcal{D}$. Classically, episodes considered in this paper only contain the first infection event of each node (the earliest time a content reached the node). Let $\mathcal{U} = \{u_0, u_1, ..., u_{N-1}\}$ be a set of N nodes, u_0 standing for the world node, used to model influences from external factors or spontaneous infections (as done in (Gruhl et al., 2004) for instance). A diffusion episode $D = (U^D, T^D)$ reports the diffusion of a given content in the network as a sequence of infected nodes $U^D = \left(U_0^D, ..., U_{|D|-1}^D\right)$ and a set $T^D = \{t_u^D \in \mathbb{R}^+ + \infty | u \in \mathcal{U}\}$ of infection time-stamps for all $u \in \mathcal{U}$. U^D is ordered chronologically w.r.t. the infection time-stamps T^D. Thus, $U_i^D \in \mathcal{U}$ corresponds to the i-th infected node in D for all $i \in \{0, ..., |D| - 1\}$, with $|D|$ the number of infected nodes in the diffusion. Every episode in $\mathcal{D}$ starts by the world node u_0 (i.e., $U_0^D = u_0$ for all episodes D). We note t_u^D the infection time-stamp in D for any node in $\mathcal{U}$, ∞ for nodes not infected in D. Time-stamps are relative w.r.t. to $t_{U_1^D}^D$, arbitrarily set to 1 in the data. Note that we also set $t_{U_0^D}^D = 0$ for every episode D. In the following, $D_i = (U_i^D, t_{U_i^D}^D)$ denotes the $i - th$ infected node in U with its associated time-stamp.

Cascades are richer structures than diffusion episodes, since they describe how a given diffusion happened. The cascade structure stores the first transmission event $(u \rightarrow v)$ that succeeded from any node u to each infected node v. Thus, a cascade $C^D = (U^D, T^D, I^D)$ corresponds to a transmission tree rooted in u_0 and reaching nodes in U^D

during the diffusion, according to a sequence I^D of infector indices in U^D: for any $j \in \{1, ..., |D| - 1\}$ and any $i \in \{0, ..., |D| - 2\}$, I_j^D equals i iff U_j^D was infected by U_i^D in the diffusion D (i.e., U_i^D is the infector of U_j^D). We arbitrarily set $I_0^D = 0$ (no infector for the world node). Note that I_1^D is always equal to 0, since there is no other candidate for being the infector of U_1^D than the world u_0. For convenience, we note $I^D(i) \in U^D$ the infector of U_i^D for $i \in \{0, ..., |D| - 1\}$. Note that the cascade structures respect that $t_{I^D(j)}^D < t_{U_j^D}^D$ for every D and all $j \in \{1, ..., |D| - 1\}$ (the infector of a node v is mandatory a node that was infected before v). Several different cascade structures are possible for a given diffusion episode. Cascade models usually make assumptions on these latent diffusion structures to learn the diffusion parameters.

2.2. Cascade Models

Unlike the Linear Threshold (LT) methods (Granovetter, 1978), which consider the social pressure exerted on nodes to trigger new infections, cascade models focus on the individual influence of the infected nodes on their neighborhood. By considering the spread of diffusion as cascades of infections over the network, these cascade models tend to reproduce more realistic temporal diffusion dynamics on social networks (Guille et al., 2013). The Independent Cascade model (IC) (Goldenberg et al., 2001) follows an iterative process in which, at each iteration, every newly infected node u gets a unique chance to infect any other node v of the network with some probability $\theta_{u,v}$. The process iterates while new infections occur. Since the EM algorithm proposed by (Saito et al., 2008) to learn its parameters, IC has been at the heart of diffusion works.

However, in real life, diffusion occurs in continuous time, not discrete as assumed in IC. (Lamprier et al., 2016) proposed DAIC, a delay-agnostic version of IC, where diffusion between nodes is assumed to follow uniform delay distributions rather than occurring in successive discrete time-steps. A neural version of DAIC has then been proposed in (Bourigault et al., 2016), to improve robustness and compactness, via representation learning of diffusion probabilities. Beyond uniform time delay distributions, two main works deal with continuous-time diffusion. NetRate (Gomez-Rodriguez et al., 2011) learns parametric time-dependent distributions to fit with observed infection time-stamps. As NetRate, CTIC (Saito et al., 2009) uses exponential distributions to model delays of diffusion between nodes, but rather than a single parameter for each possible relationship, delays and influence factors are considered as separated parameters, which leads to more freedom w.r.t. to observed diffusion tendencies. Delays and influence parameters are learned conjointly by an EM-like algorithm. Note the continuous-time cascade model extension in (Zhang

et al., 2018), which embeds users in a representation space so that their relative positions both respect some structural community properties and can be used to explain infection time-stamps of users. Several extensions of IC have also concerned the use of context information about the modeled diffusion (see for instances (Saito et al., 2011), (Wang et al., 2012), (Guille & Hacid, 2012) or (Lagnier et al., 2013)).

2.3. The Continuous-Time Independent Cascade model

In our model we consider that diffusion probabilities from any infected node v depend on a latent state associated to v, which embeds the past trajectory of the diffused content. This state depends on the state of the node u who first transmitted the content to v. Therefore, we need to rely on a continuous-time model such as CTIC (Saito et al., 2009), which serves as a basis for our work. In CTIC, two parameters are defined for each pair (u, v) of nodes in the network: $k_{u,v} \in [0; 1]$, which corresponds to the probability that node u succeeds in infecting v, and $r_{u,v} > 0$, which corresponds to a time-delay parameter used in an exponential distribution when u infects v. If u succeeds in infecting v in an episode D, v is infected at time $t_v^D = t_u^D + \delta$, where $\delta \sim Exp(r_{u,v})$. These parameters are learned via maximizing the following likelihood on a set of episodes $\mathcal{D}$:

$$p(\mathcal{D}) = \prod_{D \in \mathcal{D}} p(D) = \prod_{D \in \mathcal{D}} \prod_{v \in U^D} h_v^D \prod_{v \notin U^D} g_v^D \quad (1)$$

where h_v^D stands for the probability that v is infected at t_v^D by previous infected nodes in D and g_v^D is the probability that v is not infected by any infected node in D.

We build on this in the following, but rather than considering a pair of parameters $k_{u,v}$ and $r_{u,v}$ for each pair of nodes (u, v) (which implies $2 \times |\mathcal{U}| \times (|\mathcal{U}| - 1)$ parameters to store), we propose to consider neural functions which output the corresponding parameters according to the hidden state of the emitter u, depending on its ancestor branch in the cascade, and a continuous embedding of the receiver v.

3. Recurrent Neural Diffusion Model

This section first presents our recurrent generative model of cascades. Then, it details the proposed learning algorithm.

3.1. Recurrent Diffusion Model

As discussed above, we consider that each infected node v in an episode D owns a state $z_v^D \in \mathbb{R}^d$ depending on the path the content followed to reach v in D, with d the dimension of the representation space. Knowing the state z_u^D of the node u that first infected v, the state z_v^D is computed as:

$$z_v^D = f_\phi(z_u^D, \omega_v^{(f)}) \quad (2)$$

with $f_\phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$ a function, with parameters ϕ, that transforms the state of u according to a static continuous representation $\omega_v^{(f)} \in \mathbb{R}^d$ for the node v. This function can either be an Elman RNN cell, a multi-layer perceptron (MLP) or a Gated Recurrent Unit (GRU). An LSTM could also be used here, but z_v^D should include both the cell and the state of v in that case.

Given a state z for an infected node u in D, the probability that u succeeds in transmitting the diffused content to v is given by:

$$k_{u,v}(z) = \sigma\left(< z, \omega_v^{(k)} > \right) \tag{3}$$

where $\sigma(.)$ stands for the sigmoid function and $\omega_v^{(k)}$ an embedding of size d for any node $v \in \mathcal{U}$ as a content receiver.

Similarly to the CTIC model, if a node u succeeds in infecting another node v, the delay of infection depends on an exponential distribution with parameter $r_{u,v}$. To simplify learning, we assume that the delay of infection does not depend on the history of diffusion, only the probability of infection does. Thus, for a given pair (u, v), the delay parameter is the same for every episode D:

$$r_{u,v} = exp(-| < \omega_u^{(r,1)}, \omega_v^{(r,2)} > |) \tag{4}$$

with $|x|$ the absolute value of a real scalar x and where $\omega_u^{(r,1)}$ and $\omega_v^{(r,2)}$ correspond to two embeddings of size d for any node $u \in \mathcal{U}$, $\omega^{(r,1)}$ for the source of the transition and $\omega^{(r,2)}$ for its destination in order to enable asymmetric behavior.

We set $\Theta = \left(\phi, z_0, (\omega_u^{(f)})_{u \in \mathcal{U}}, (\omega_u^{(r,1)})_{u \in \mathcal{U}}, (\omega_u^{(r,2)})_{u \in \mathcal{U}}, (\omega_u^{(k)})_{u \in \mathcal{U}} \right)$ as the parameters of our model, where z_0 corresponds to the state of the initial node (the world node) u_0. The full generative process, similar to the one of CTIC, is given in the supplementary material. The process iterates while there remains some nodes in a set of infectious nodes (initialized with u_0). At each iteration, the process selects the infectious node u with minimal time-stamp of infection, removes it from the set of infectious nodes, records it as infected and attempts to infect each non infected node v according to the probability $k_{u,v}(z_u^D)$. If it succeeds, a time t is sampled for v with an exponential law with parameter $r_{u,v}$. If the new time t for v is lower than its current time t_v^D (initialized with $t_v^D = \infty$), this new time is stored in t_v^D, v is added to the set of infectious nodes and its new state z_v^D is computed according to its new infector u.

3.2. Learning the Model

As in CTIC, we need to define the probability that the node $u \in U^D$ infects the node $v \in U^D$ at time $t_v^D > t_u^D$:

$$a_{u,v}^D(z) = k_{u,v}(z) r_{u,v} \exp^{-r_{u,v}(t_v^D - t_u^D)} \tag{5}$$

Also, the probability that u does not infect v before t_v^D given a state z for u in D is:

$$
\begin{aligned}
b_{u,v}^D(z) &= 1 - k_{u,v}(z) \int_{t_u^D}^{t_v^D} r_{u,v} \exp^{-r_{u,v}(t - t_u^D)} \, dt \\
&= k_{u,v}(z) \exp^{-r_{u,v}(t_v^D - t_u^D)} + 1 - k_{u,v}(z) \tag{6}
\end{aligned}
$$

The probability density that node v is infected at time t_v^D given a set of states $\boldsymbol{z}$ for all nodes infected before v is:

$$
\begin{aligned}
h_v^D(\boldsymbol{z}) &= \sum_{\substack{u \in \mathcal{U}, \\ t_u^D < t_v^D}} a_{u,v}^D(z_u) \prod_{\substack{x \in \mathcal{U} \backslash \{u\}, \\ t_x^D < t_v^D}} b_{x,v}^D(z_x) \\
&= \prod_{\substack{x \in \mathcal{U}, \\ t_x^D < t_v^D}} b_{x,v}^D(z_x) \sum_{\substack{u \in \mathcal{U}, \\ t_u^D < t_v^D}} \frac{a_{u,v}^D(z_u)}{b_{u,v}^D(z_u)} \tag{7}
\end{aligned}
$$

where z_u is the state of node u in $\boldsymbol{z}$. Similarly, the probability density that node v is not infected in D at the end of observation time T given a set of states $\boldsymbol{z}$ for all nodes infected in D is:

$$
\begin{aligned}
g_v^D(\boldsymbol{z}) &= \prod_{u \in U^D} (k_{u,v}(z_u) \exp^{-r_{u,v}(T - t_u^D)} + 1 - k_{u,v}(z_u)) \\
&\approx \prod_{u \in U^D} (1 - k_{u,v}(z_u)) \tag{8}
\end{aligned}
$$

where the approximation is done assuming a sufficiently long observation period.

The learning process of our model is based on a likelihood maximization, similarly to maximizing eq.1 in the classical CTIC model. However, in our case the infection probabilities depend on hidden states $\boldsymbol{z}$ associated to the infected nodes. Since observations only contain infection time-stamps, this requires to marginalize over every possible sequence of ancestors I for every $D \in \mathcal{D}$:

$$\log p(\mathcal{D}) = \sum_{D \in \mathcal{D}} \log p(D) = \sum_{D \in \mathcal{D}} \log \sum_{I \in \mathcal{I}^D} p(D, I) \tag{9}$$

where $\mathcal{I}^D$ is the set $\{v \in \mathbb{N}^{|D|-1} | v_0 = 0 \wedge \forall i > 0, v_i < i\}$ of all possible ancestors sequences for D. $p(D, I)$ corresponds to the joint probability of the episode D and an ancestor sequence $I \in \mathcal{I}^D$. Taking $p(D, I) = p(I)p(D|I)$ would lead to an intractable computation of $p(D|I)$ using our recurrent cascade model, since it would imply to estimate the probability of any infection in D according to the full ancestors sequence. Fortunately, using the bayesian chain rule, the joint probability can be written as:

$$
\begin{aligned}
p(D, I) &= \prod_{i=1}^{|D|-1} p(D_i|D_{<i}, I_{<i}) \, p(I_i|D_{\le i}, I_{<i}) \\
&\times \prod_{v \notin U^D} p(v \notin U^D | D_{\le |D|-1}, I) \tag{10}
\end{aligned}
$$

where $D_{<i} = (D_j)_{j \in \{0,...,i-1\}}$ corresponds to the sequence of the i first components of D (the i first components in U^D with their associated time-stamps) and $I_{<i} = (I_j)_{j \in \{0,...,i-1\}}$ stands for the vector containing the i first components of I. We have for every $i \in \{1, ..., |D|-1\}$: $p(D_i|D_{<i}, I_{<i}) = h_{U_i^D}^D(z_{<i}^D)$, where $z_{<i}^D$ is a set containing the states of the i first infected nodes in D, which can be deduced from $D_{<i}$ and $I_{<i}$ using the equation 2. We also have: $p(v \notin U^D|D_{\leq|D|-1}, I) = g_v^D(z_{\leq i}^D)$. The probability $p(I_i|D_{\leq i}, I_{<i})$ is the conditional probability that $I(U_i^D)$ was the node who first infected U_i^D, given all the previous infection events and the fact that U_i^D was infected at $t_{U_i^D}^D$ by one of the previously infected nodes in D. It can be obtained, according to formula 7, via:

$$p(I_i|D_{\leq i}, I_{<i}) =$$
$$\frac{a_{I(U_i^D),U_i^D}^D\left(z_{I(U_i^D)}^D\right) / b_{I(U_i^D),U_i^D}^D\left(z_{I(U_i^D)}^D\right)}{\sum\limits_{u \in \mathcal{U}, t_u^D < t_{U_i^D}^D} a_{u,U_i^D}^D\left(z_u^D\right) / b_{u,U_i^D}^D\left(z_u^D\right)} \quad (11)$$

with $I(U_i^D) \in U^D$ the infector of U_i^D stored in I.

Unfortunately the log-likelihood from formula 9 is still particularly difficult to optimize directly since it requires to consider every possible vector $I \in \mathcal{I}^D$ for each training episode D at each optimization iteration. Moreover, the probability products in formula 10 would lead to zero gradients because of floating point representation limits. Therefore, we need to define an approach where the optimization can be done via trajectory sampling. Different choices would be possible. First, MCMC approaches such as the Gibbs Sampling EM could be used, but they require to sample from the posteriors of the full trajectories of the cascades, which is very unstable and complex to perform. The full computation of the posterior distributions could be avoided by using simpler propositional distributions (such as done for instance via importance sampling with auxiliary variables in (Farajtabar et al., 2015) for diffusion source detection), but this would face a very high variance in our case. Another possibility is to adopt a variational approach (Kingma & Welling, 2013), where an auxiliary distribution q is learned for the inference of the latent variables. As done in (Krishnan et al., 2016) for the inference in sequences, a smoothing strategy could be developed by relying on a bi-directional RNN that would consider past and future infections for the inference of the ancestors of nodes via $q(I_i|D, I_{<i})$ for every infected node i in an episode D. However, learning the parameters of such a distribution is particularly difficult (episodes of different lengths, cascades considered as sequences, etc.). Also, another possibility for smoothing would be to define an independent distribution q_i^D for every episode $D \in \mathcal{D}$ and every infection $i \in \{1, ..., |D| - 1\}$. However, this induces a huge number of variational parameters, increasing with the size of the training set (linearly

in the number of training episodes and quadratically in the size of the episodes). Thus, we propose to rather rely on the conditional distribution of ancestors given the past for sampling (i.e, $q_i^D(I_i) = p(I_i|D_{\leq i}, I_{<i})$), which corresponds to a filtering inference process.

From the Jensen inequality on concave functions, we get for a given episode D:

$$
\begin{aligned}
\log p(D) &= \log \sum_{I \in \mathcal{I}^D} p(D, I) \\
&\geq \mathbb{E}_{I \sim q^D}\left[\log p(D, I) - \log q^D(I)\right] \\
&= \mathbb{E}_{I \sim q^D}\left[\sum_{i=1}^{|D|-1} \log p(D_i|D_{<i}, I_{<i}) \right. \\
&\qquad\qquad \left. + \sum_{v \notin U^D} \log p(v \notin U^D|D_{\leq|D|-1}, I)\right] \\
&\triangleq \mathcal{L}(D; \Theta) \quad (12)
\end{aligned}
$$

where $q^D = \prod_{i=1}^{|D|-1} p(I_i|D_{\leq i}, I_{<i})$. This leads to a lower-bound of the log-likelihood, which corresponds to an expectation from which it is easy to sample: at each new infection of a node i in a episode D, we can sample from a distribution depending on the past only. Maximizing this lower-bound (also called the ELBO) encourages the process to choose trajectories that explain the best the observed episode. To maximize it via stochastic optimization, we refer to the score function estimator (Ranganath et al., 2014), which leverages the derivative of the log-function ($\nabla_\theta \log p(\mathbf{x}; \theta) = \frac{\nabla_\theta p(\mathbf{x}; \theta)}{p(\mathbf{x}; \theta)}$) to express the gradient as an expectation from which we can sample. Another possibility would have been to rely on the Gumbel-Softmax and the Concrete distribution with reparametrization such as done in (Maddison et al., 2016), but we observed greatly better results using the log-trick. The gradient of the ELBO function for all the episodes is given by:

$$\nabla_\Theta \mathcal{L}(\mathcal{D}; \Theta) = \quad (13)$$

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

where $p^I(D)$ is a shortcut for $\prod_{i=1}^{|D|-1} p(D_i|D_{<i}, I_{<i})$ $\prod_{v \notin U^D} p(v \notin U^D|D_{\leq|D|-1}, I)$ and b is a moving-average baseline of the ELBO per training episode, used to reduce the variance (taken over the 100 previous training epochs in our experiments). This stochastic gradient formulation enables to obtain unbiased steepest ascent directions despite the need to sample the ancestor vectors for the computation of the node states (with the replacement of expectations by averages over K samples for each episode). It contains two

terms: while the first one encourages high conditional probabilities for ancestors that maximize the likelihood of the full episodes, the second one leads to improve the likelihood of the observed infections regarding the past of the sampled diffusion path.

The optimization is done using the ADAM optimizer (Kingma & Ba, 2014) over mini-batches of M episodes ordered by length to avoid padding ($M = 512$ and $K = 1$ in our experiments). Our full efficient algorithm is given in the supplementary material [2].

4. Experiments

4.1. Setup

We perform experiments on one artificial and three real-world datasets:

- Arti: Episodes generated on a scale-free random graph of 100 nodes (obtained following the BarabsiAlbert algorithm). The generation process follows the CTIC model, where each $k_{u,v}$ is a function of the transmitted content and the features of the receiver v. Features of the hub nodes (nodes with a degree greater than 30) are sampled from a Dirichet with $\alpha = 10$ (multi-content nodes), while those of other nodes are sampled from a Dirichet with $\alpha = 0.1$ (content-specific nodes). For each episode, a content $z \in \mathbb{R}^5$ is sampled from a Dirichlet with parameter $\alpha = 0.1$ before each simulation and the sigmoid of the dot product between this content and the node features determines the transmission probabilities. The content remains hidden from the experimented models. 10000 episodes for training, 5000 for validation, 5000 for testing. Mean length of the episodes=6.89 (stdev=7.7).

- Digg: Data collected from the Digg stream API during one month. Infections are the "diggs" posted by users on links published on the media. We kept the 100 most active users from the collected data. 20000 episodes for training, 5000 for validation, 5000 for testing. Mean length of the episodes=4.26 (stdev=9,26).

- Weibo: Retweet cascades extracted from the Weibo microbloging website using the procedure described in (Leskovec et al., 2009). The dataset was collected by (Fu et al., 2013). 4000 nodes, 45000 episodes for training, 5000 for validation, 5000 for testing. Mean length of episodes=4.58 (stdev=2.15).

- Memetracker: The memetracker dataset described in (Leskovec et al., 2009) contains millions of blog posts and news articles. Each website or blog stands as a user,

and we use the phrase clusters extracted by (Leskovec et al., 2009) to trace the flow of information. 500 nodes, 250000 for training, 5000 for validation, 5000 for testing. Mean length of episodes=8.68 (stdev=11.45).

We consider the following temporal baselines:

- CTIC: the Continuous-Time Independent Cascade model in its original version (Saito et al., 2009);

- RNN: the Recurrent Temporal Point Process model from (Du et al., 2016) where episodes are considered as sequences that can be treated with a classical RNN outputting at each step the probability distributions of the next infected node and its time-stamp;

- CYAN: Similar to RMTPP but with an attention mechanism to select previous states (Wang et al., 2017b);

- CYAN-cov: The same as Cyan but with a more sophisticated attention mechanism using an history of attention states, to give more weights to important nodes;

- DAN: the attention model described in (Wang et al., 2018). It is very similar to CYAN but uses a pooling mechanism rather than a recurrent network to consider the past in the predictions. In the version of (Wang et al., 2018), the model only predicts the next infected node at each step, not its time of infection. To enable a comparison with the other models, we extended it by adding a time prediction mechanism similar to the temporal head of CYAN.

- EmbCTIC: a version of our model where the node state z is replaced in eq. 3 by a static embedding for the source (similarly to the formulation of the delay parameter in eq. 4). This corresponds to an embedded version of CTIC, similarly to the embedded version of DAIC from (Bourigault et al., 2016).

Note that to adapt baselines based on RNN for diffusion modeling and render them comparable to cascade-based ones, we add a "end node" at the end of each episode before training and testing them. In such a way, these models are able to model the end of the episodes via this end node as the next event (no time-delay prediction for this node however).

Our model and the baselines were tuned by a grid search process on a validation set for each dataset (although the best hyper-parameters obtained for Arti1 remained near optimal for the other ones). For every model with an embedding space (i.e., all except CTIC), we set its dimension to $d = 50$ (larger dimensions induce a more difficult convergence of the learning without significant gain in accuracy). The reported results for our model use a GRU module as the recurrent state transformation function f_ϕ.

We evaluate our models on three distinct tasks:

---

[2]The full code in python is available at: https://github.com/lampriers/recCTIC

Diffusion modelling The performances of the methods are reported in term of negative log-likelihood of the test episodes (i.e., $NLL = -(1/|\mathcal{D}_{test}|) \sum_{D \in \mathcal{D}_{test}} \log p(D)$). Lower values denote models that are less surprised by test episodes than others, rendering their generalization ability. The NLL measure depends on the model, but for each one it renders the probability of an episode to be observed according to the model, both on which nodes are eventually infected and at what time. For our model which has to sample trajectories, the NLL is approximated via importance sampling by considering $p(D) \approx \frac{1}{S} \sum_{s=1}^{S} \frac{p(D, I^{(s)})}{q^D(I^{(s)})}$ computed on S infector vectors sampled from q^D. We used $S = 100$ in our experiments;

Diffusion generation The models are compared on their ability to simulate realistic cascades. The aim is to predict the marginal probabilities of nodes to be eventually infected. The results are reported in term of Cross-Entropy (CE) taken over the whole set of nodes for each episode: $CE = \frac{1}{|\mathcal{D}_{test}| \times |\mathcal{U}|} \sum_{D \in \mathcal{D}_{test}} \sum_{u \in \mathcal{U}} \log p(u \in U^D)^{\mathbb{I}(u \in U^D)} + \log p(u \notin U^D)^{\mathbb{I}(u \notin U^D)}$, where $\mathbb{I}(.)$ stands for the indicator function returning 1 if its argument is true, 0 else. $p(u \in U^D)$ is estimated via Monte-Carlo simulations (following the generation process of the models and counting the rate of simulations in which u is infected). 1000 simulations are performed for each test episode in our experiments.

Diffusion Path prediction The models are assessed on their ability to choose the true infectors in observed diffusion episodes. This is only considered on the artificial dataset for which we have the ground truth on who infected whom. The INF measure corresponds to the expectation of the rate of true infectors chosen by the models: $INF = \frac{1}{\sum_{D \in \mathcal{D}_{test}}(|D|-1)} \sum_{D \in \mathcal{D}_{test}} \sum_{i \in \{1,...,|D|-1\}} p(I_i = inf(i, D)|D_{\leq i})$, with $inf(i, D)$ the true infector of the i-th infected node in the episode D. In the RNN baseline, their is no infector selection mechanism, it is thus excluded from the results for this measure. For models with attention (CYAN and DAN), we consider the attention weights as selection probabilities. For cascade based models which explicitly model this, we directly use the corresponding probability $p(I_i = inf(i, D)|D_{\leq i})$. In our model, this corresponds to an expectation over previous infectors in the cascade (i.e., $\frac{1}{S} \sum_{s=1}^{S} p(I_i = inf(i, D)|D_{\leq i}, I_{<i}^{(s)})$), with $I^{(s)}$ the s-th sampled vector from q^D.

For each task, we report results with different amounts of initial observations from test episodes: infections occurred before a given delay τ from the start of the episode are given as input to the models, from with they infer internal representations, evaluation measures are computed on the remaining of the episode. In tables 1 to 4, τ_0 means that

NLL	τ_0	τ_1	τ_2	τ_3
rnn	19,72	14,55	11,78	7,65
cyan	18,78	13,62	11,04	7,09
cyan-cov	19,55	14,35	11,56	7,41
dan	18,71	13,49	10,86	6,84
ctic	20,08	14,26	10,18	5,99
embCTIC	19,90	14,13	10,11	5,97
recCTIC	**17,39**	**11,59**	**8,31**	**4,97**

CE	τ_0	τ_1	τ_2	τ_3
rnn	79,0	21,0	14,2	9,3
cyan	86,9	19,6	14,4	9,4
cyan-cov	91,7	27,8	19,7	12,4
dan	97,9	98,6	75,7	43,8
ctic	**63,1**	23,2	24,7	21,8
embCTIC	65,8	22,6	17,2	12,1
recCTIC	68,7	**15,9**	**11,2**	**8,3**

INF	τ_0	τ_1	τ_2	τ_3
cyan	0,30	0,15	0,11	0,11
cyan-cov	0,29	0,14	0,10	0,10
dan	0,42	0,31	0,23	0,20
ctic	0,73	0,67	0,61	0,58
embCTIC	0,73	0,67	0,61	0,58
recCTIC	**0,90**	**0,88**	**0,86**	**0,84**

Table 1. Results on the artificial dataset.

nothing was initially observed, the models are not conditioned on the start of the episodes. τ_1 means that infections at the first time stamp are known beforehand, prediction and modeling results thus concern time-stamps greater than 1 (models are conditioned on diffusion sources). τ_2 and τ_3 mean that infections occurred respectively before a delay of $\tau = maxT/20$ and a delay of $\tau = maxT/10$ from the start of the episode are known and used to condition the models. More details are given in the supplementary material.

4.2. Results

Results on the artificial dataset are given in Table 1. In this well-fitted dataset, embedding the history for predicting the future of diffusion looks of very great importance: our approach shows very significant improvements over all other models on the three measures (except for CE with weak conditioning). In this dataset there exists some hub nodes through which most of the diffusion episodes transit, whatever the nature of the diffusion. This corresponds to a usual setting in most of real-world (scale-free) networks. In that case, the path of the diffusion contains very useful information that can be leveraged to predict infections after the hub node: the infection of the hub node is a necessary condition for the infection of its successors, but not sufficient since this node is triggered in various kinds of situations. Depending on who transmitted the content to it, different successors are infected then. Markovian cascade models such as CTIC or embCTIC cannot model this since infection probabilities only depend on disjoint past events of infection, not on paths taken by the propagated content. RNN-based models are better armed for this, but their performances are undermined by

NLL	τ_0	τ_1	τ_2	τ_3
rnn	31,03	26,18	24,15	23,02
cyan	16,82	11,56	9,32	8,03
cyan-cov	21,36	16,83	14,50	13,31
dan	18,47	13,52	11,43	10,29
ctic	27,70	22,07	19,20	17,74
embCTIC	15,98	10,31	7,92	6,75
recCTIC	**15,67**	**10,30**	**7,86**	**6,74**
CE	τ_0	τ_1	τ_2	τ_3
rnn	0,46	0,28	0,24	0,22
cyan	0,43	0,25	0,21	0,19
cyan-cov	0,43	**0,23**	0,20	0,17
dan	0,44	0,25	0,22	0,19
ctic	0,45	0,31	0,20	0,16
embCTIC	0,43	0,31	0,21	0,17
recCTIC	**0,43**	0,27	**0,17**	**0,14**

Table 2. NLL and CE results on Digg

NLL	τ_0	τ_1	τ_2	τ_3
rnn	27,58	28,98	18,62	17,15
cyan	29,59	30,04	30,04	18,79
cyan-cov	27,50	29,12	29,12	18,55
dan	32,35	25,02	21,97	20,39
ctic	23,92	17,88	13,31	12,28
embCTIC	24,71	18,02	13,58	12,39
recCTIC	**21,72**	**14,08**	**11,29**	**10,34**
CE	τ_0	τ_1	τ_2	τ_3
rnn	0,59	0,37	0,30	0,28
cyan	0,59	0,37	0,31	0,29
cyan-cov	0,59	0,36	0,30	0,28
dan	**0,58**	0,26	0,26	0,24
ctic	**0,58**	0,25	0,31	0,28
embCTIC	0,59	0,26	0,30	0,28
recCTIC	0,59	**0,24**	**0,21**	**0,20**

Table 3. NLL and CE results on Weibo

NLL	τ_0	τ_1	τ_2	τ_3
rnn	112,3	118,2	110,4	103,6
cyan	115,2	113,1	109,2	102,1
cyan-cov	95,58	95,05	93,64	90,20
dan	91,70	89,97	86,19	78,91
ctic	52,70	55,54	48,48	44,33
embCTIC	54,18	52,29	49,68	45,15
recCTIC	**50,11**	**49,34**	**48,35**	**42,20**
CE	τ_0	τ_1	τ_2	τ_3
rnn	1,68	1,66	1,59	1,51
cyan	1,66	1,64	1,59	1,49
cyan-cov	1,61	1,59	1,52	1,39
dan	1,59	**1,58**	1,58	1,44
ctic	**1,33**	1,68	1,60	1,46
embCTIC	1,59	1,66	1,57	1,39
recCTIC	1,51	1,60	**1,49**	**1,36**

Table 4. NLL and CE results on Memetracker

their way of aggregating past information. Attention mechanisms of CYAN and DAN attempt to overcome this, but look quite unstable with errors accumulating through time. Our approach appears as an effective compromise between both worlds, by embedding past as RNN approaches, while maintaining the bayesian cascade structures of the data. Its results on the INF measure are very great compared to the other approaches, which highlights the very good ability for uncovering the dynamics of diffusion, even for such complex problems with strong entanglement between diffusion episodes of different natures.

The good behavior of the approach is not only observed on the artificial dataset, which has been generated by the cascade model of CTIC on a graph of relationships, but also on real-world datasets. Tables 2 to 4 report results on the three real-world datasets. In these tables, we observe that RNN based approaches have more difficulties to model test episodes than cascade based ones. The attention mechanism of the CYAN and DAN approaches allow them to get sometimes closer to the cascade based results (especially on Digg), but their performances are very variable from a dataset to another. These methods are good for the task they were initially designed - predicting the directly next infection (this had been observed in our experimentations)-, but not for modeling or long term prediction purposes. This is a strong limitation since the directly next infection at a given point does not help much to understand the dynamics of a network. Our approach obtains better results than all other methods in most of settings, especially for the dynamics modelling task (NLL), though infection prediction results (CE) are also usually good compared with its competitors. Interestingly, while embCTIC usually beats CTIC, recCTIC often obtains even better results. This validates that the history of nodes in the diffusion has a great importance for capturing the main dynamics of the network. Thanks to the black-box inference process and the recurrent mechanism of our proposal, the propositional distribution

q^D is encouraged to resemble the conditionnal probability of the full ancestors vector. Regarding the results, the inference process looks to have actually converged toward useful trajectories. This enables the model to adapt distributions regarding the diffusion trajectory during learning. This also allows the model to simulate more consistent cascades regarding sources of diffusion.

5. Conclusion

We proposed a recurrent cascade-based diffusion modeling approach, which is at the crossroads of cascade-based and RNN approaches. It leverages the best of both worlds with an ability to embed the history of diffusion for prediction while still capturing the tree dependencies underlying the diffusion in network. Results validate the approach both for modeling and prediction tasks. In this work, we based the sampling of trajectories on a filtering approach where only the past observations are considered for the inference of the infector of a node. Outgoing works concern the development of an inductive variational distribution that rely on whole observed episodes for inference.

Acknowledgements

References

Bourigault, S., Lamprier, S., and Gallinari, P. 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*, pp. 573–582, 2016. doi: 10.1145/2835776.2835817. URL http://doi.acm.org/10.1145/2835776.2835817.

Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., and Song, L. 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, pp. 1555–1564, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939875. URL http://doi.acm.org/10.1145/2939672.2939875.

Farajtabar, M., Gomez-Rodriguez, M., Du, N., Zamani, M., Zha, H., and Song, L. Back to the past: Source identification in diffusion networks from partially observed cascades. *CoRR*, abs/1501.06582, 2015. URL http://arxiv.org/abs/1501.06582.

Fu, K., hong Chan, C., and Chau, M. Assessing censorship on microblogs in china: Discriminatory keyword analysis and the real-name registration policy. *Internet Computing, IEEE*, 17(3):42–50, May 2013. ISSN 1089-7801. doi: 10.1109/MIC.2013.28.

Goldenberg, J., Libai, B., and Muller, E. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.

Gomez-Rodriguez, M., Balduzzi, D., and Schölkopf, B. Uncovering the temporal dynamics of diffusion networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ICML '11, pp. 561–568. ACM, 2011.

Granovetter, M. Threshold Models of Collective Behavior. *American Journal of Sociology*, 83(6):1420–1143, 1978.

Gruhl, D., Guha, R., Liben-Nowell, D., and Tomkins, A. Information diffusion through blogspace. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pp. 491–501, New York, NY, USA, 2004. ACM. ISBN 1-58113-844-X.

Guille, A. and Hacid, H. 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, 2012.

Guille, A., Hacid, H., Favre, C., and Zighed, D. A. Information diffusion in online social networks: A survey. *SIGMOD Rec.*, 42(2):17–28, July 2013. ISSN 0163-5808. doi: 10.1145/2503792.2503797. URL http://doi.acm.org/10.1145/2503792.2503797.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Krishnan, R. G., Shalit, U., and Sontag, D. Structured Inference Networks for Nonlinear State Space Models. *ArXiv e-prints*, September 2016.

Lagnier, C., Denoyer, L., Gaussier, E., and Gallinari, P. Predicting information diffusion in social networks using content and user's profiles. In *European Conference on Information Retrieval*, ECIR '13, 2013.

Lamprier, S., Bourigault, S., and Gallinari, P. Influence learning for cascade diffusion models: focus on partial orders of infections. *Social Netw. Analys. Mining*, 6(1):93:1–93:16, 2016. doi: 10.1007/s13278-016-0406-1. URL https://doi.org/10.1007/s13278-016-0406-1.

Leskovec, J., Backstrom, L., and Kleinberg, J. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pp. 497–506, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. doi: 10.1145/1557019.1557077.

Li, C., Ma, J., Guo, X., and Mei, Q. Deepcas: an end-to-end predictor of information cascades. *CoRR*, abs/1611.05373, 2016. URL http://arxiv.org/abs/1611.05373.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *CoRR*, abs/1611.00712, 2016. URL http://arxiv.org/abs/1611.00712.

Nguyen, G. H., Lee, J. B., Rossi, R. A., Ahmed, N. K., Koh, E., and Kim, S. Continuous-time dynamic network embeddings. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pp. 969–976, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-5640-4. doi: 10.1145/

3184558.3191526. URL https://doi.org/10.1145/3184558.3191526.

Perozzi, B., Al-Rfou, R., and Skiena, S. Deepwalk: Online learning of social representations. *CoRR*, abs/1403.6652, 2014. URL http://arxiv.org/abs/1403.6652.

Ranganath, R., Gerrish, S., and Blei, D. M. Black Box Variational Inference. *ArXiv e-prints*, December 2014.

Saito, K., Nakano, R., and Kimura, M. 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, pp. 67–75. Springer-Verlag, 2008.

Saito, K., Kimura, M., Ohara, K., and Motoda, H. 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, pp. 322–337, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-05223-1.

Saito, K., Ohara, K., Yamagishi, Y., Kimura, M., and Motoda, H. Learning diffusion probability based on node attributes in social networks. In Kryszkiewicz, M., Rybinski, H., Skowron, A., and Ras, Z. W. (eds.), *ISMIS*, volume 6804 of *Lecture Notes in Computer Science*, pp. 153–162. Springer, 2011. ISBN 978-3-642-21915-3. URL http://dblp.uni-trier.de/db/conf/ismis/ismis2011.html#SaitoOYKM11.

Shi, Y., Lei, M., Zhang, P., and Niu, L. Diffusion based network embedding. *CoRR*, abs/1805.03504, 2018. URL http://arxiv.org/abs/1805.03504.

Tai, K. S., Socher, R., and Manning, C. D. Improved semantic representations from tree-structured long short-term memory networks. *CoRR*, abs/1503.00075, 2015. URL http://arxiv.org/abs/1503.00075.

Ver Steeg, G. and Galstyan, A. 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, pp. 3–12, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1869-3. doi: 10.1145/2433396.2433400.

Wang, J., Zheng, V. W., Liu, Z., and Chang, K. C. Topological recurrent neural network for diffusion prediction. *CoRR*, abs/1711.10162, 2017a. URL http://arxiv.org/abs/1711.10162.

Wang, L., Ermon, S., and Hopcroft, J. E. 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, pp. 499–514. Springer-Verlag, 2012.

Wang, Y., Shen, H., Liu, S., Gao, J., and Cheng, X. Cascade dynamics modeling with attention-based recurrent neural network. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 2985–2991, 2017b. doi: 10.24963/ijcai.2017/416. URL https://doi.org/10.24963/ijcai.2017/416.

Wang, Z., Chen, C., and Li, W. Attention network for information diffusion prediction. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, pp. 65–66, Republic and Canton of Geneva, Switzerland, 2018. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-5640-4. doi: 10.1145/3184558.3186931. URL https://doi.org/10.1145/3184558.3186931.

Zhang, Y., Lyu, T., and Zhang, Y. Cosine: Community-preserving social network embedding from information diffusion cascades. In *AAAI*, 2018.