Journal of Machine Learning Research 18 (2017) 1-49 Submitted 3/16; Revised 1/17; Published 5/17

COEVOLVE: A Joint Point Process Model for Information Diffusion and Network Evolution*◇

Mehrdad Farajtabar MEHRDAD@GATECH.EDU
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA

Yichen Wang YICHEN.WANG@GATECH.EDU
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA

Manuel Gomez-Rodriguez MANUELGR@MPI-SWS.ORG
MPI for Software Systems
Paul-Ehrlich-Strasse, 67663 Kaiserslautern
Germany

Shuang Li SLI370@GATECH.EDU
H. Milton Stewart School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA

Hongyuan Zha ZHA@CC.GATECH.EDU
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA

Le Song LSONG@CC.GATECH.EDU
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA

Editor: Edo Airoldi

Abstract

Information diffusion in online social networks is affected by the underlying network topology, but it also has the power to change it. Online users are constantly creating new links when exposed to new information sources, and in turn these links are alternating the way information spreads. However, these two highly intertwined stochastic processes, information diffusion and network evolution, have been predominantly studied *separately*, ignoring their co-evolutionary dynamics.

We propose a temporal point process model, COEVOLVE, for such joint dynamics, allowing the intensity of one process to be modulated by that of the other. This model allows us to efficiently simulate interleaved diffusion and network events, and generate

. *Preliminary version of this work appeared in (Farajtabar et al., 2015b).
. ◇ Implementation codes are available at https://github.com/farajtabar/Coevolution

traces obeying common diffusion and network patterns observed in real-world networks such as Twitter. Furthermore, we also develop a convex optimization framework to learn the parameters of the model from historical diffusion and network evolution traces. We experimented with both synthetic data and data gathered from Twitter, and show that our model provides a good fit to the data as well as more accurate predictions than alternatives.

Keywords: social networks, information diffusion, network structure, co-evolutionary dynamics, point processes, Hawkes process, survival analysis

1. Introduction

Online social networks, such as Twitter or Weibo, have become large information networks where people share, discuss and search for information of personal interest as well as breaking news (Kwak et al., 2010). In this context, users often forward to their *followers* information they are exposed to via their *followees*, triggering the emergence of information *cascades* that travel through the network (Cheng et al., 2014), and constantly create new links to information sources, triggering changes in the network itself over time. Importantly, recent empirical studies with Twitter data have shown that both information diffusion and network evolution are coupled and network changes are often triggered by information diffusion (Antoniades and Dovrolis, 2015; Weng et al., 2013; Myers and Leskovec, 2014).

While there have been many recent works on modeling information diffusion (Gomez-Rodriguez et al., 2010; Du et al., 2013; Cheng et al., 2014; Farajtabar et al., 2015a) and network evolution (Chakrabarti et al., 2004; Leskovec et al., 2008, 2010), most of them treat these two stochastic processes independently and separately, ignoring the influence one may have on the other over time. More notably, Weng et al. (2013) were the first to show experimental evidence that information diffusion influences network evolution in microblogging sites both at system-wide and individual levels. In particular, they studied *Yahoo! Meme*, a social micro-blogging site similar to Twitter, which was active between 2009 and 2012, and showed that information diffusion causes about 24 % of the new links, and that the likelihood of a new link from a user to another increases with the number of posts by the second user seen by the first one. Also, Antoniades and Dovrolis (2015) studied the temporal characteristics of retweet-driven connections within the Twitter network and realized that the number of retweets is an important factor to infer such connections. They showed that links created due to information diffusion account for 42 % of new links, and are rather infrequent, compared to tweets and retweets, but they are responsible for a large percentage of the new links in Twitter.

However, there are a few limitations in the above-mentioned studies. First, they only characterize the effect that information diffusion has on the network dynamics, or the vice versa, but not the bidirectional influence. Second, previous studies are mostly empirical and only make binary predictions on link creation events without precise timing. For example, the work of (Weng et al., 2013; Antoniades and Dovrolis, 2015) predict whether a new link will be created based on the number of retweets; and, Myers and Leskovec (2014) predict whether a burst of new links will occur based on the number of retweets and users' similarity. Thus, to better understand information diffusion and network evolution, there is an urgent need for joint probabilistic models of the two processes, which are largely inexistent to date.

In this paper, we propose a probabilistic generative model, Coevolve, for the joint dynamics of information diffusion and network evolution on Twitter-like networks. Our

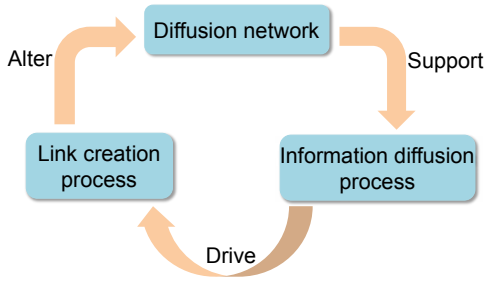


Figure 1: Illustration of how information diffusion and network structure processes interact

model is able to learn parameters from real world data, and predict the precise timing of both diffusion and new link events. The proposed model is based on the framework of temporal point processes, which explicitly characterizes the continuous time interval between events, and it consists of two interwoven and interdependent components, as shown in Figure 1:

I. **Information diffusion process.** We design an "identity revealing" multivariate Hawkes process (Farajtabar et al., 2014) to capture the mutual excitation behavior of retweeting events, where the intensity of such events in a user is boosted by previous events from her time-varying set of followees. Although Hawkes processes have been used for information diffusion before (Farajtabar et al., 2016; Blundell et al., 2012; Iwata et al., 2013; Zhou et al., 2013a; Farajtabar et al., 2014; Linderman and Adams, 2014; Valera and Gomez-Rodriguez, 2015), the key innovation of our approach is to explicitly model the excitation due to a particular source node, hence revealing the identity of the source. Such design reflects the reality that information sources are explicitly acknowledged, and it also allows a particular information source to acquire new links in a rate according to her "informativeness".

II. **Network evolution process.** We model link creation as an "information driven" survival process, and couple the intensity of this process with retweeting events. Although survival processes have been used for link creation before (Hunter et al., 2011; Vu et al., 2011), the key innovation in our model is to incorporate retweeting events as the driving force for such processes. Since our model has captured the source identity of each retweeting event, new links will be targeted toward information sources, with an intensity proportional to their degree of excitation and each source's influence.

Our model is designed in such a way that it allows the two processes, information diffusion and network evolution, unfold simultaneously in the same time scale and exercise bidirectional influence on each other, allowing sophisticated coevolutionary dynamics to be generated, as illustrated in Figure 2.

Importantly, the flexibility of our model does not prevent us from efficiently simulating diffusion and link events from the model and learning its parameters from real world data:

- **Efficient simulation.** We design a scalable sampling procedure that exploits the sparsity of the generated networks. Its complexity is $O(nd \log m)$, where n is the number of events, m is the number of users and d is the maximum number of followees per user.

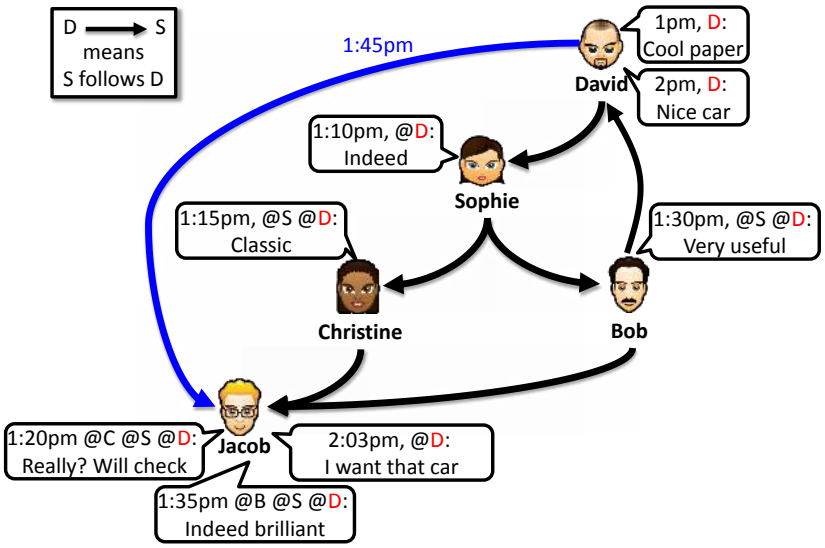


Figure 2: Illustration of information diffusion and network structure co-evolution.
Information diffusion → network structure: David's tweet at 1:00 pm about a paper is retweeted by Sophie and Christine respectively at 1:10 pm and 1:15 pm to reach out to Jacob. Jacob retweets about this paper at 1:20 pm and 1:35 pm and then finds David a good source of information and decides to follow him directly at 1:45 pm.
Information diffusion → network structure: As a result new path of information from David to Jacob (and his downstream followers) is created. Consequently, a subsequent tweet by David about a car at 2:00 pm directly reaches out to Jacob without need to Sophie and Christine retweet.

- **Convex parameters learning.** We show that the model parameters that maximize the joint likelihood of observed diffusion and link creation events can be efficiently found via convex optimization.

Then, we experiment with our model and show that it can produce coevolutionary dynamics of information diffusion and network evolution, and generate retweet and link events that obey common information diffusion patterns (*e.g.*, cascade structure, size and depth), static network patterns (*e.g.*, node degree) and temporal network patterns (*e.g.*, shrinking diameter) described in related literature (Leskovec et al., 2005, 2010; Goel et al., 2012). Finally, we show that, by modeling the coevolutionary dynamics, our model provides significantly more accurate link and diffusion event predictions than alternatives in large scale Twitter data set (Antoniades and Dovrolis, 2015).

The remainder of this article is organized as follows. We first proceed by building sufficient background on the temporal point processes framework in Section 2. Then, we introduce our joint model of information diffusion and network structure co-evolution in Section 3. Sections 4 and 5 are devoted to answer two essential questions: how can we generate data from the model? and how can we efficiently learn the model parameters from historical event data? Any generative model should be able to answer the above questions.

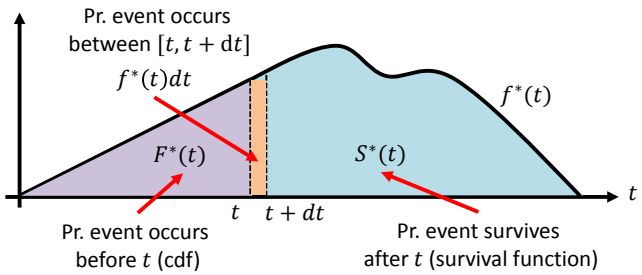


Figure 3: Illustration of three inter-related quantities in point processes framework: conditional density function, conditional cumulative density function, and survival function.

In Sections 6, 7, and 8 we perform empirical investigation of the properties of the model, we evaluate the accuracy of the parameter estimation in synthetic data, and we evaluate the performance of the proposed model in the real-world data set, respectively. Section 9 reviews the related work, and finally, the paper is concluded in Section 10. Proofs, more detailed contents and experimental results, and some extensions are left to the appendices.

2. Background on Temporal Point Processes

A temporal point process is a random process whose realization consists of a list of discrete events localized in time, $\{t_i\}$ with $t_i \in \mathbb{R}^+$ and $i \in \mathbb{Z}^+$. Many different types of data produced in online social networks can be represented as temporal point processes, such as the times of retweets and link creations. A temporal point process can be equivalently represented as a counting process, $N(t)$, which records the number of events before time t. Let the history $\mathcal{H}(t)$ be the list of times of events $\{t_1, t_2, \ldots, t_n\}$ up to but not including time t. Then, the number of observed events in a small time window $[t, t + dt)$ of length dt is

$$dN(t) = \sum_{t_i \in \mathcal{H}(t)} \delta(t - t_i)\, dt, \tag{1}$$

and hence $N(t) = \int_0^t dN(s)$, where $\delta(t)$ is a Dirac delta function. More generally, given a function $f(t)$, we can define the convolution with respect to $dN(t)$ as

$$f(t) \star dN(t) := \int_0^t f(t - \tau)\, dN(\tau) = \sum\nolimits_{t_i \in \mathcal{H}(t)} f(t - t_i). \tag{2}$$

The point process representation of temporal data is fundamentally different from the discrete time representation typically used in social network analysis. It directly models the time interval between events as random variables, avoids the need to pick a time window to aggregate events, and allows temporal events to be modeled in a fine grained fashion. Moreover, it has a remarkably rich theoretical support (Aalen et al., 2008).

An important way to characterize temporal point processes is via the conditional intensity function—a stochastic model for the time of the next event given all the times of previous events. Formally, the conditional intensity function $\lambda^*(t)$ (intensity, for short) is

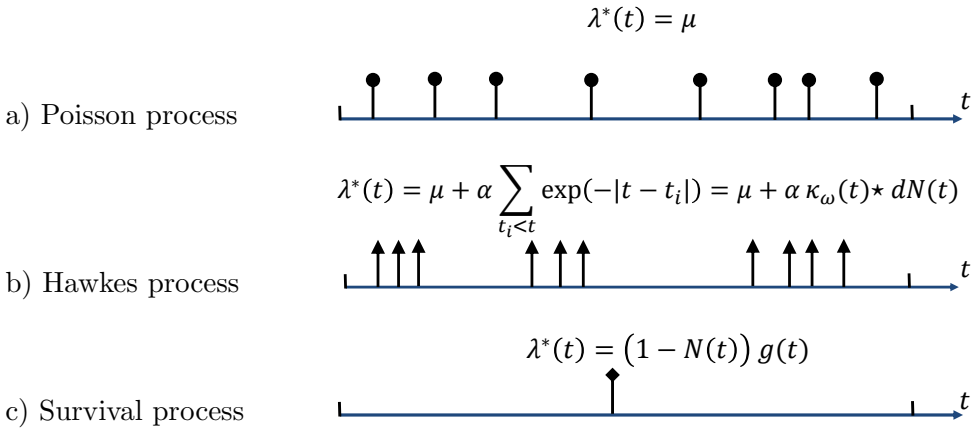


Figure 4: Three types of point processes with a typical realization

the conditional probability of observing an event in a small window $[t, t + dt)$ given the history $\mathcal{H}(t)$, *i.e.*,

$$\lambda^*(t)dt := \mathbb{P}\left\{\text{event in } [t, t + dt) | \mathcal{H}(t)\right\} = \mathbb{E}[dN(t)|\mathcal{H}(t)], \tag{3}$$

where one typically assumes that only one event can happen in a small window of size dt and thus $dN(t) \in \{0, 1\}$. Then, given the observation until time t and a time $t' \geqslant t$, we can also characterize the conditional probability that no event happens until t' as

$$S^*(t') = \exp\left(-\int_t^{t'} \lambda^*(\tau)\, d\tau\right), \tag{4}$$

the (conditional) probability density function that an event occurs at time t' as

$$f^*(t') = \lambda^*(t')\, S^*(t'), \tag{5}$$

and the (conditional) cumulative density function, which accounts for the probability that an event happens before time t':

$$F^*(t') = 1 - S^*(t') = \int_t^{t'} f^*(\tau)\, d\tau. \tag{6}$$

Figure 3 illustrates these quantities. Moreover, we can express the log-likelihood of a list of events $\{t_1, t_2, \ldots, t_n\}$ in an observation window $[0, T)$ as

$$\mathfrak{L} = \sum_{i=1}^{n} \log \lambda^*(t_i) - \int_0^T \lambda^*(\tau)\, d\tau, \quad T \geqslant t_n. \tag{7}$$

This simple log-likelihood will later enable us to learn the parameters of our model from observed data. Finally, the functional form of the intensity $\lambda^*(t)$ is often designed to capture the phenomena of interests. Some useful functional forms we will use are (Aalen et al., 2008):

(i) **Poisson process.** The intensity is assumed to be independent of the history $\mathcal{H}(t)$, but it can be a nonnegative time-varying function, *i.e.*,

$$\lambda^*(t) = g(t) \geqslant 0. \tag{8}$$

(ii) **Hawkes process.** The intensity is history dependent and models a mutual excitation between events, *i.e.*,

$$\lambda^*(t) = \mu + \alpha\kappa_\omega(t) \star dN(t) = \mu + \alpha \sum\nolimits_{t_i \in \mathcal{H}(t)} \kappa_\omega(t - t_i), \tag{9}$$

where,

$$\kappa_\omega(t) := \exp(-\omega t)\mathbb{I}[t \geqslant 0] \tag{10}$$

is an exponential triggering kernel and $\mu \geqslant 0$ is a baseline intensity independent of the history. Here, the occurrence of each historical event increases the intensity by a certain amount determined by the kernel and the weight $\alpha \geqslant 0$, making the intensity history dependent and a stochastic process by itself. In our work, we focus on the exponential kernel, however, other functional forms, such as log-logistic function, are possible, and the general properties of our model do not depend on this particular choice.

(iii) **Survival process.** There is only one event for an instantiation of the process, *i.e.*,

$$\lambda^*(t) = (1 - N(t))g(t), \tag{11}$$

where $g(t) \geqslant 0$ and the term $(1 - N(t))$ makes sure $\lambda^*(t)$ is 0 if an event already happened before t.

Figure 4 illustrates these processes. Interested reader should refer to Aalen et al. (2008) for more details on the framework of temporal point processes.

Point process models have been applied to many tasks in networks such as fake news mitigation (Farajtabar et al., 2017), recommendation systems (Hosseini et al., 2017), outlier detection (Li et al., 2017), activity shaping in social networks (Farajtabar et al., 2014), verifying crowd-generated data (Tabibian et al., 2017), sequence modeling using deep recurrent neural networks (Xiao et al., 2017), and campaigning in networks (Farajtabar et al., 2016).

3. Generative Model of Information Diffusion and Network Evolution

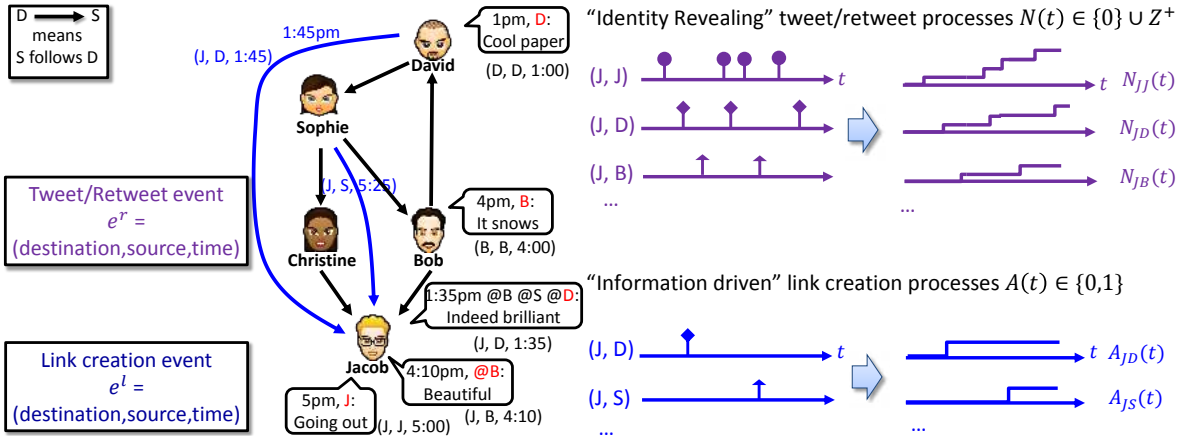
In this section, we use the above background on temporal point processes to formulate Coevolve, our probabilistic model for the joint dynamics of information diffusion and network evolution.

3.1 Event Representation

We model the generation of two types of events: tweet/retweet events, e^r, and link creation events, e^l. Instead of just the time t, we record each event as a triplet, as illustrated in Figure 5(a):

$$e^r \quad \text{or} \quad e^l \quad := \quad (\underset{\substack{\uparrow \\ \text{destination}}}{u}, \quad \overset{\text{source}}{\underset{\downarrow}{s}}, \quad \underset{\substack{\uparrow \\ \text{time}}}{t}). \tag{12}$$

For retweet event, the triplet means that the destination node u retweets at time t a tweet originally posted by source node s. Recording the source node s reflects the real world scenario that information sources are explicitly acknowledged. Note that the occurrence of event e^r does *not* mean that u is directly retweeting from or is connected



(a) Event representation (b) Point and counting processes

Figure 5: Events of point processes and their associated counting processes for link creation and information diffusion; (a) Trace of events generated by a tweet from David followed by new links Jacob creates to follow David and Sophie; (b) The associated points in time and the counting process realizations.

to s. This event can happen when u is retweeting a message by another node u' where the original information source s is acknowledged. Node u will pass on the same source acknowledgement to its followers (*e.g.*, "I agree @a @b @c @s"). Original tweets posted by node u are allowed in this notation. In this case, the event will simply be $e^r = (u, u, t)$. Given a list of retweet events up to but not including time t, the history $\mathcal{H}^r_{us}(t)$ of retweets by u due to source s is

$$\mathcal{H}^r_{us}(t) = \{e^r_i = (u_i, s_i, t_i) | u_i = u \text{ and } s_i = s\}. \tag{13}$$

The entire history of retweet events is denoted as

$$\mathcal{H}^r(t) := \cup_{u,s\in[m]} \mathcal{H}^r_{us}(t). \tag{14}$$

For link creation event, the triplet means that destination node u creates at time t a link to source node s, *i.e.*, from time t on, node u starts following node s. To ease the exposition, we restrict ourselves to the case where links cannot be deleted and thus each (directed) link is created only once. However, our model can be easily augmented to consider multiple link creations and deletions per node pair, as discussed in Section E. We denote the link creation history as $\mathcal{H}^l(t)$.

3.2 Joint Model with Two Interwoven Components

Given m users, we use two sets of counting processes to record the generated events, one for information diffusion and another for network evolution. More specifically,

I. Retweet events are recorded using a matrix $\boldsymbol{N}(t)$ of size $m \times m$ for each fixed time point t. The (u, s)-th entry in the matrix, $N_{us}(t) \in \{0\} \cup \mathbb{Z}^+$, counts the number of retweets of u due to source s up to time t. These counting processes are "identity revealing", since they keep track of the source node that triggers each retweet. The

matrix $\boldsymbol{N}(t)$ is typically less sparse than $\boldsymbol{A}(t)$, since $N_{us}(t)$ can be nonzero even when node u does not directly follow s. We also let $d\boldsymbol{N}(t) := (\ dN_{us}(t)\ )_{u,s\in[m]}$.

II. Link events are recorded using an adjacency matrix $\boldsymbol{A}(t)$ of size $m \times m$ for each fixed time point t. The (u, s)-th entry in the matrix, $A_{us}(t) \in \{0, 1\}$, indicates whether u is directly following s. Therefore, $A_{us}(t) = 1$ means the directed link has been created before t. For simplicity of exposition, we do not allow self-links. The matrix $\boldsymbol{A}(t)$ is typically sparse, but the number of nonzero entries can change over time. We also define $d\boldsymbol{A}(t) := (\ dA_{us}(t)\ )_{u,s\in[m]}$.

Then, the interwoven information diffusion and network evolution processes can be characterized using their respective intensities

$$\mathbb{E}[d\boldsymbol{N}(t) \,|\, \mathcal{H}^r(t) \cup \mathcal{H}^l(t)] = \boldsymbol{\Gamma}^*(t)\, dt \tag{15}$$

$$\mathbb{E}[d\boldsymbol{A}(t) \,|\, \mathcal{H}^r(t) \cup \mathcal{H}^l(t)] = \boldsymbol{\Lambda}^*(t)\, dt, \tag{16}$$

where,

$$\boldsymbol{\Gamma}^*(t) = (\ \gamma^*_{us}(t)\ )_{u,s\in[m]} \tag{17}$$

$$\boldsymbol{\Lambda}^*(t) = (\ \lambda^*_{us}(t)\ )_{u,s\in[m]}. \tag{18}$$

The sign * means that the intensity matrices will depend on the joint history, $\mathcal{H}^r(t) \cup \mathcal{H}^l(t)$, and hence their evolution will be coupled. By this coupling, we make: (i) the counting processes for link creation to be "information driven" and (ii) the evolution of the linking structure to change the information diffusion process. In the next two sections, we will specify the details of these two intensity matrices.

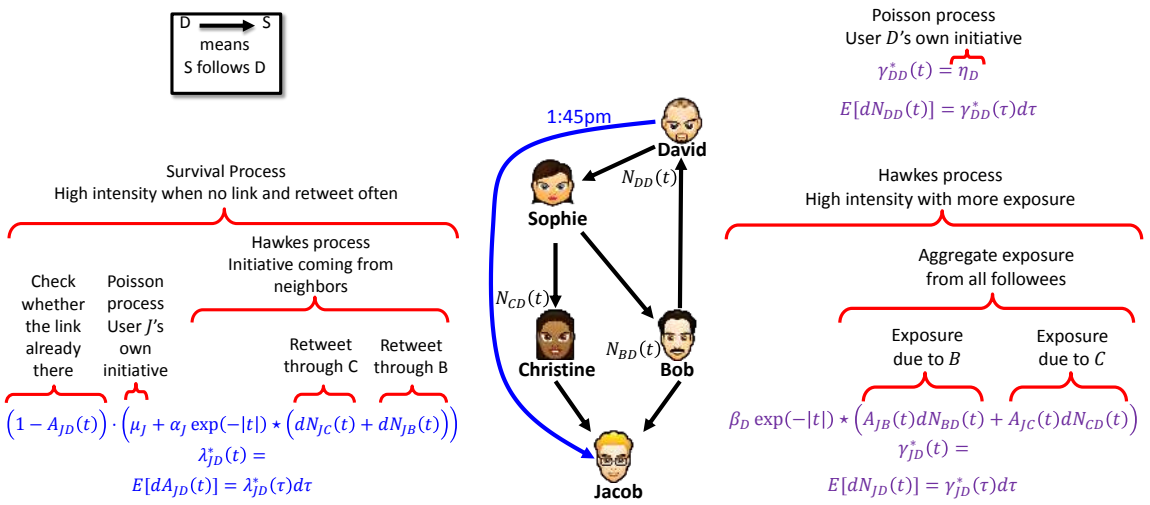
3.3 Information Diffusion Process

We model the intensity, $\boldsymbol{\Gamma}^*(t)$, for retweeting events using multivariate Hawkes process:

$$\gamma^*_{us}(t) = \mathbb{I}[u = s]\, \eta_u + \mathbb{I}[u \neq s]\, \beta_s \sum\nolimits_{v\in\mathcal{F}_u(t)} \kappa_{\omega_1}(t) \star (A_{uv}(t)\ dN_{vs}(t))\,, \tag{19}$$

where $\mathbb{I}[\cdot]$ is the indicator function and $\mathcal{F}_u(t) := \{v \in [m] : A_{uv}(t) = 1\}$ is the current set of followees of u. The term $\eta_u \geqslant 0$ is the intensity of original tweets by a user u on his own initiative, becoming the source of a cascade, and the term $\beta_s \sum_{v\in\mathcal{F}_u(t)} \kappa_{\omega_1}(t) \star (A_{uv}(t)\ dN_{vs}(t))$ models the propagation of peer influence over the network, where the triggering kernel $\kappa_{\omega_1}(t)$ models the decay of peer influence over time.

Note that the retweeting intensity matrix $\boldsymbol{\Gamma}^*(t)$ is by itself a stochastic process that depends on the time-varying network topology, the non-zero entries in $\boldsymbol{A}(t)$, whose growth is controlled by the network evolution process in Section 3.4. Hence the model design captures the influence of the network topology and each source's influence, β_s, on the information diffusion process. More specifically, to compute $\gamma^*_{us}(t)$, one first finds the current set $\mathcal{F}_u(t)$ of followees of u, and then aggregates the retweets of these followees that are due to source s. Note that these followees may or may not *directly* follow source s. Then, the more frequently node u is exposed to retweets of tweets originated from source s via her followees, the more likely she will also retweet a tweet originated from source s. Once node u retweets due to source s, the corresponding $N_{us}(t)$ will be incremented, and this in turn will increase the likelihood of triggering retweets due to source s among the followers of u. Thus, the source does *not* simply broadcast the message to nodes directly following her but her influence



(a) Link creation process (b) Social network (c) Information diffusion process

Figure 6: Our hypothetical social network where the information diffusion paths make Jacob follow David: (a) The breakdown of conditional intensity function for the link creation process of Jacob following David $A_{JD}(t)$; (b) The information paths between Jacob and David; (c) The information diffusion processes for David tweeting on his own initiative $N_{DD}(t)$ and Jacob retweeting posts originated from David $N_{JD}(t)$.

propagates through the network even to those nodes that do not directly follow her. Finally, this information diffusion model allows a node to repeatedly generate events in a cascade, and is very different from the independent cascade or linear threshold models (Kempe et al., 2003) which allow at most one event per node per cascade.

3.4 Network Evolution Process

In our model, each user is exposed to information through a time-varying set of neighbors. By doing so, information diffusion affects network evolution, increasing the practical application of our model to real-world network data sets. The particular definition of exposure (*e.g.*, a retweet's neighbor) depends on the type of historical information that is available. Remarkably, the flexibility of our model allows for different types of diffusion events, which we can broadly classify into two categories.

In the first category, events corresponds to the times when an information cascade hits a person, for example, through a retweet from one of her neighbors, but she does not explicitly like or forward the associated post. Here, we model the intensity, $\mathbf{\Lambda}^*(t)$, for link creation using a combination of survival and Hawkes process:

$$\lambda^*_{us}(t) = (1 - A_{us}(t)) \left(\mu_u + \alpha_u \sum_{v \in \mathcal{F}_u(t)} \kappa_{\omega_2}(t) \star dN_{vs}(t) \right), \tag{20}$$

where the term $1 - A_{us}(t)$ effectively ensures a link is created only once, and after that, the corresponding intensity is set to zero. The term $\mu_u \geqslant 0$ denotes a baseline intensity, which models when a node u decides to follow a source s spontaneously at her own initiative. The term $\alpha_u \kappa_{\omega_2}(t) \star dN_{vs}(t)$ corresponds to the retweets by node v (a followee of node u) which are originated from source s. The triggering kernel $\kappa_{\omega_2}(t)$ models the decay of interests over time.

In the second category, the person decides to explicitly like or forward the associated post and influencing events correspond to the times when she does so. In this case, we model the intensity, $\mathbf{\Lambda}^*(t)$, for link creation as:

$$\lambda^*_{us}(t) = (1 - A_{us}(t))(\mu_u + \alpha_u \, \kappa_{\omega_2}(t) \star dN_{us}(t)), \tag{21}$$

where the terms $1 - A_{us}(t)$, $\mu_u \geqslant 0$, and the decaying kernel $\kappa_{\omega_2}(t)$ play the same role as the corresponding ones in Equation (20). The term $\alpha_u \kappa_{\omega_2}(t) \star dN_{us}(t)$ corresponds to the retweets of node u due to tweets originally published by source s. The higher the corresponding retweet intensity, the more likely u will find information by source s useful and will create a *direct* link to s.

In both cases, the link creation intensity $\mathbf{\Lambda}^*(t)$ is also a stochastic process by itself, which depends on the retweet events, be it the retweets by the neighbors of node u or the retweets by node u herself, respectively. Therefore, it captures the influence of retweets on the link creation, and closes the loop of mutual influence between information diffusion and network topology. Figure 6 illustrates these two interdependent intensities.

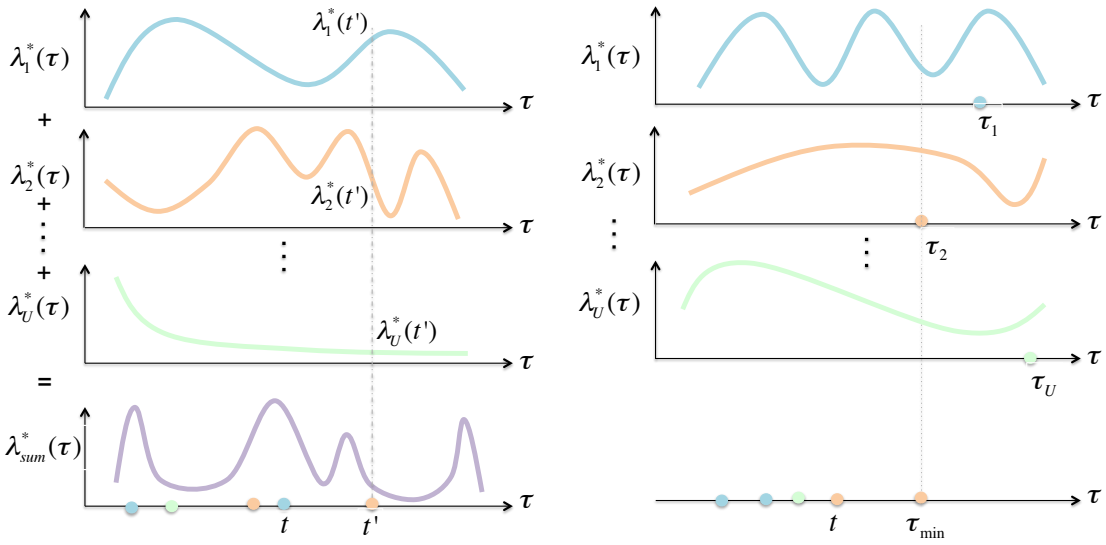
Intuitively, in the latter category, information diffusion events are more prone to trigger new connections, because, they involve the target and source nodes in an explicit interaction, however, they are also less frequent. Therefore, it is mostly suitable to large event data sets, as the ones we generate in our synthetic experiments. In contrast, in the former category, information diffusion events are less likely to inspire new links but found in abundance. Therefore, it is more suitable for smaller data sets, as the ones we use in our real-world experiments. Consequently, in our synthetic experiments we used the latter and in our real-world experiments, we used the former.

More generally, the choice of exposure event should be made based on the type and amount of available historical information. Note that, these are two realizations, among the many others, of the link formation process in the Coevolve framework. Many other extensions can be found in appendix E. In practice, these different forms can be utilized depending on the conditions and constraints of the networks and the data in hand. More importantly, in this section, we make one example on how to tailor the model to fit the application best.

Finally, note that creating a link is more than just adding a path or allowing information sources to take shortcuts during diffusion. The network evolution makes fundamental changes to the diffusion dynamics and stationary distribution of the diffusion process in Section 3.3. As shown in Farajtabar et al. (2014), given a fixed network structure $\boldsymbol{A}$, the expected retweet intensity $\boldsymbol{\mu}_s(t)$ at time t due to source s will depend of the network structure in a nonlinear fashion, *i.e.*,

$$\boldsymbol{\mu}_s(t) := \mathbb{E}[\mathbf{\Gamma}^*_{\cdot s}(t)] = (e^{(\boldsymbol{A} - \omega_1 \boldsymbol{I})t} + \omega_1 (\boldsymbol{A} - \omega_1 \boldsymbol{I})^{-1} (e^{(\boldsymbol{A} - \omega_1 \boldsymbol{I})t} - \boldsymbol{I}))\, \boldsymbol{\eta}_s, \tag{22}$$

where $\boldsymbol{\eta}_s \in \mathbb{R}^m$ has a single nonzero entry with value η_s and $e^{(\boldsymbol{A} - \omega_1 \boldsymbol{I})t}$ is the matrix exponential. When $t \to \infty$, the stationary intensity $\bar{\boldsymbol{\mu}}_s = (\boldsymbol{I} - \boldsymbol{A}/\omega)^{-1} \boldsymbol{\eta}_s$ is also nonlinearly



(a) Ogata's algorithm (b) Proposed algorithm

Figure 7: Ogata's algorithm vs our simulation algorithm in simulating U interdependent point processes characterized by intensity functions $\lambda_1(t), \ldots, \lambda_U(t)$; (a) Illustrating Ogata's algorithm, which first takes a sample from the process with intensity equal to sum of individual intensities and then assigns it to the proper dimension proportionally to its contribution to the sum of intensities; (b) Illustrating our proposed algorithm, which first draws a sample from each dimension independently and then takes the minimum time among them.

related to the network structure. Thus, given two network structures $\boldsymbol{A}(t)$ and $\boldsymbol{A}(t')$ at two points in time, which are different by a few edges, the effect of these edges on the information diffusion is not just an additive relation. Depending on how these newly created edges modify the eigen-structure of the sparse matrix $\boldsymbol{A}(t)$, their effect on the information diffusion dynamics can be very significant.

4. Efficient Simulation of Coevolutionary Dynamics

We could simulate samples (link creations, tweets and retweets) from our model by adapting Ogata's thinning algorithm (Ogata, 1981), originally designed for multidimensional Hawkes processes. However, a naive implementation of Ogata's algorithm would scale poorly, *i.e.*, for each sample, we would need to re-evaluate $\boldsymbol{\Gamma}^*(t)$ and $\boldsymbol{\Lambda}^*(t)$. Thus, to draw n sample events, we would need to perform $O(m^2n^2)$ operations, where m is the number of nodes. Figure 7(a) schematically demonstrates the main steps of Ogata's algorithm. Please refer to Appendix B for further details.

Here, we design a sampling procedure that is especially well-fitted for the structure of our model. The algorithm is based on the following key idea: if we consider each intensity function in $\boldsymbol{\Gamma}^*(t)$ and $\boldsymbol{\Lambda}^*(t)$ as a separate point process and draw a sample from each, the minimum among all these samples is a valid sample for the multidimensional point process.

Algorithm 1 Simulation Algorithm for Coevolve

Initialization:
Initialize the priority queue Q
for $\forall\ u, s \in [m]$ **do**
  Sample next link event e^l_{us} from A_{us} (Algorithm 3)
  $Q.insert(e^l_{us})$
  Sample next retweet event e^r_{us} from N_{us} (Algorithm 3)
  $Q.insert(e^r_{us})$
end for
General Subroutine:
$t \leftarrow 0$
while $t < T$ **do**
  $e \leftarrow Q.extract\_min()$
  if $e = (u, s, t')$ is a retweet event **then**
    Update the history $\mathcal{H}^r_{us}(t') = \mathcal{H}^r_{us}(t) \cup \{e\}$
    for $\forall\ v\ s.t.\ u \rightsquigarrow v$ **do**
      Update event intensity: $\gamma_{vs}(t') = \gamma_{vs}(t'^{-}) + \beta$
      Sample retweet event e^r_{vs} from γ_{vs} (Algorithm 3)
      $Q.update\_key(e^r_{vs})$
      if NOT $s \rightsquigarrow v$ **then**
        Update link intensity: $\lambda^*_{vs}(t') = \lambda^*_{vs}(t'^{-}) + \alpha$
        Sample link event e^l_{vs} from λ_{vs} (Algorithm 3)
        $Q.update\_key(e^l_{vs})$
      end if
    end for
  else
    Update the history $\mathcal{H}^l_{us}(t') = \mathcal{H}^l_{us}(t) \cup \{e\}$
    $\lambda^*_{us}(t) \leftarrow 0\ \ \forall\ t > t'$
  end if
  $t \leftarrow t'$
end while

As the results of this section are general and can be applied to simulate any multi-dimensional point process model we abuse the notation a little bit and represent U (possibly inter-dependent) point processes by U intensity functions $\lambda^*_1, \ldots, \lambda^*_U$. In the specific case of simulating coevolutionary dynamics we have $U = m^2 + m(m-1)$ were the first and second terms are the number information diffusion and link creation processes, respectively. Figure 7 illustrates the way in which both algorithms differ. The new algorithm has the following steps:

1. Initialization: Simulate each dimension separately and find their next sampled event time.

2. Minimization: Take the minimum among all the sampled times and declare it as the next event of the multidimensional process.

Algorithm 2 Efficient Intensity Computation

Global Variabels:
Last time of intensity computation: t
Last value of intensity computation: I
Initialization:
$t \leftarrow 0$
$I \leftarrow \mu$
function $get\_intensity(t')$
  $I' \leftarrow (I - \mu)\exp(-\omega(t' - t)) + \mu$
  $t \leftarrow t'$
  $I \leftarrow I'$
  return I
end function

Algorithm 3 1-D next event sampling

Input: Current time: t
Output: Next event time: s
$s \leftarrow t$
$\hat{\lambda} \leftarrow \lambda^*(s)$ (Algorithm 2)
while $s < T$ **do**
  $g \sim Exponential(\hat{\lambda})$
  $s \leftarrow s + g$
  $\bar{\lambda} \leftarrow \lambda^*(s)$ (Algorithm 2)
  Rejection test:
  $d \sim Uniform(0, 1)$
  if $d \times \hat{\lambda} < \bar{\lambda}$ **then**
    return s
  else
    $\hat{\lambda} = \bar{\lambda}$
  end if
end while
return s

3. Update: Recalculate the intensities of the dimensions that are affected by this approved sample and re-sample only their next event. Then go to step 2.

Note that, events are sampled one by one; After each event, the intensities of affected dimensions are recomputed and fixed. This means that, the history is fixed and the dimensions of the Hawkes process become independent (just until the next event). We take next sample. Then, we make the necessary updates in the intensity functions to capture the influence of dimensions on each other. Therefore, each dimension can be thought independent until next sample is drawn because the intensity function is known.

To prove that the new algorithm generates samples from the same distribution as Ogata's algorithm does we need the following Lemma. It justifies step 2 of the above outline.

Lemma 1 *Assume we have U independent non-homogeneous Poisson processes with intensity $\lambda_1^*(\tau), \ldots, \lambda_U^*(\tau)$. Take random variable τ_u equal to the time of process u's first event after time t. Define $\tau_{min} = \min_{1 \le u \le U} \{\tau_u\}$ and $u_{min} = \operatorname{argmin}_{1 \le u \le U} \{\tau_u\}$. Then,*

(a) τ_{min} is the first event after time t of the Poisson process with intensity $\lambda_{sum}^*(\tau)$. In other words, τ_{min} has the same distribution as the next event (t') in Ogata's algorithm.

(b) u_{min} follows the conditional distribution $\mathbb{P}(u_{min} = u | \tau_{min} = x) = \frac{\lambda_U^*(x)}{\lambda_{sum}^*(x)}$. I.e. the dimension firing the event comes from the same distribution as the one in Ogata's algorithm.

Given the above Lemma, we can now prove that the distribution of the samples generated by the proposed algorithm is identical to the one generated by Ogata's method.

Theorem 2 *The sequence of samples from Ogata's algorithm and our proposed algorithm follow the same distribution.*

It's noteworthy that the dimensions are not independent, but their dependency is considered in the update stage. Until the next update happens they can be considered independent because their intensity function is determined and fixed.

This new algorithm is specially suitable for the structure of our inter-coupled processes. Since social and information networks are typically sparse, every time we sample a new node (or link) event from the model, only a small number of intensity functions in the local neighborhood of the node (or the link), will change. This number is of $O(d)$ where d is the maximum number of followers/followees per node. As a consequence, we can reuse most of the individual samples for the next overall sample. Moreover, we can find which intensity function has the minimum sample time in $O(\log m)$ operations using a heap priority queue. The heap data structure will help maintain the minimum and find it in logarithmic time with respect to the number of elements therein. Therefore, we have reduced an $O(nm)$ factor in the original algorithm to $O(d \log m)$.

Finally, we exploit the properties of the exponential function to update individual intensities for each new sample in $O(1)$. For simplicity consider a Hawkes process with intensity $\lambda^*(t) = \mu + \sum_{t_i \in \mathcal{H}_t} \alpha \exp(-\omega(t - t_i))$. Note that both link creation and information diffusion processes have this structure. Now, let $t_i < t_{i+1}$ be two arbitrary times, we have

$$\lambda^*(t_{i+1}) = (\lambda^*(t_i) - \mu) \exp(-\omega(t_{i+1} - t_i)) + \mu. \tag{23}$$

It can be readily generalized to the multivariate case too. Therefore, we can compute the current intensity without explicitly iterating over all previous events. As a result we can change an $O(n)$ factor in the original algorithm to $O(1)$. Furthermore, the exponential kernel also facilitates finding the upper bound of the intensity since it always lies at the beginning of one of the processes taken into consideration. Algorithm 2 summarizes the procedure to compute intensities with exponential kernels, and Algorithm 3 shows the procedure to sample the next event in each dimension making use of the special property of exponential triggering kernel.

The simulation algorithm is shown in Algorithm 1. By using this algorithm we reduce the complexity from $O(n^2 m^2)$ to $O(nd \log m)$, where d is the maximum number of followees per node. That means, our algorithm scales logarithmically with the number of nodes and linearly with the number of edges at any point in time during the simulation. Moreover, events for new links, tweets and retweets are generated in a temporally intertwined and

Algorithm 4 MM-type parameter learning for Coevolve

Input: Set of retweet events $\mathcal{E} = \{e_i^r\}$ and link creation events $\mathcal{A} = \{e_i^l\}$ observed in time window $[0, T)$
Output: Learned parameters $\{\mu_u\}, \{\alpha_u\}, \{\eta_u\}, \{\beta_s\}$
Initialization:
for $u \leftarrow 1$ to m **do**
  Initialize μ_u and α_u randomly
end for
for $u \leftarrow 1$ to m **do**
  $\eta_u = \frac{\sum_{e_i^r \in \mathcal{E}} \mathbb{I}[u=u_i=s_i]}{T}$
end for
for $s \leftarrow 1$ to m **do**
  $\beta_s = \frac{\sum_{e_i^r \in \mathcal{E}} \mathbb{I}[s=s_i \neq u_i]}{\sum_{u \in [m]} \mathbb{I}[u \neq s] \sum_{v \in \mathcal{F}_u(t)} \int_0^T \kappa_{\omega_1}(t) \star (A_{uv}(t)\ dN_{vs}(t))\, dt}$
end for
while not converged **do**
  for $i \leftarrow 1$ to n_l **do**
    $\nu_{i1} = \frac{\mu_{u_i}}{\mu_{u_i} + \alpha_{u_i} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \left(\kappa_{\omega_2}(t) \star dN_{vs}(t)\right)\Big|_{t=t_i}}$
    $\nu_{i2} = \frac{\alpha_{u_i} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \left(\kappa_{\omega_2}(t) \star dN_{vs}(t)\right)\Big|_{t=t_i}}{\mu_{u_i} + \alpha_{u_i} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \left(\kappa_{\omega_2}(t) \star dN_{vs}(t)\right)\Big|_{t=t_i}}$
  end for
  for $u \leftarrow 1$ to m **do**
    $\mu_u = \frac{\sum_{e_i^l \in \mathcal{A}} \mathbb{I}[u=u_i]\, \nu_{i1}}{\sum_{s \in [m]} \int_0^T (1 - A_{us}(t))\, dt}$
    $\alpha_u = \frac{\sum_{e_i^l \in \mathcal{A}} \mathbb{I}[u=u_i] \nu_{i2}}{\sum_{s \in [m]} \int_0^T (1 - A_{us}(t))(\kappa_{\omega_2}(t) \star dN_{us}(t))\, dt}$
  end for
end while

interleaving fashion, since every new retweet event will modify the intensity for link creation and vice versa.

5. Efficient Parameter Estimation from Coevolutionary Events

In this section, we first show that learning the parameters of our proposed model reduces to solving a convex optimization problem and then develop an efficient, parameter-free Minorization-Maximization algorithm to solve such problem.

5.1 Concave Parameter Learning Problem

Given a collection of retweet events $\mathcal{E} = \{e_i^r\}$ and link creation events $\mathcal{A} = \{e_i^l\}$ recorded within a time window $[0, T)$, we can easily estimate the parameters needed in our model using maximum likelihood estimation. To this aim, we compute the joint log-likelihood $\mathfrak{L}$

of these events using Equation (7), *i.e.*,

$$\mathfrak{L}(\{\mu_u\},\{\alpha_u\},\{\eta_u\},\{\beta_s\}) = \underbrace{\sum_{e_i^r\in\mathcal{E}} \log\left(\gamma^*_{u_i s_i}(t_i)\right) - \sum_{u,s\in[m]} \int_0^T \gamma^*_{us}(\tau)\, d\tau}_{\text{tweet / retweet}} + \underbrace{\sum_{e_i^l\in\mathcal{A}} \log\left(\lambda^*_{u_i s_i}(t_i)\right) - \sum_{u,s\in[m]} \int_0^T \lambda^*_{us}(\tau)\, d\tau}_{\text{links}}. \tag{24}$$

For the terms corresponding to retweets, the log term sums only over the actual observed events while the integral term actually sums over all possible combination of destination and source pairs, even if there is no event between a particular pair of destination and source. For such pairs with no observed events, the corresponding counting processes have essentially survived the observation window $[0, T)$, and the term $-\int_0^T \gamma^*_{us}(\tau)d\tau$ simply corresponds to the log survival probability. The terms corresponding to links have a similar structure.

Once we have an expression for the joint log-likelihood of the retweet and link creation events, the parameter learning problem can be then formulated as follows:

$$\begin{array}{ll} \text{minimize}_{\{\mu_u\},\{\alpha_u\},\{\eta_u\},\{\beta_s\}} & -\mathfrak{L}(\{\mu_u\},\{\alpha_u\},\{\eta_u\},\{\beta_s\}) \\ \text{subject to} & \mu_u \geq 0, \quad \alpha_u \geq 0, \quad \eta_u \geq 0, \quad \beta_s \geq 0 \quad \forall\, u, s \in [m]. \end{array} \tag{25}$$

Theorem 3 *The optimization problem defined by Equation* (25) *is jointly convex.*

5.2 Efficient Minorization-Maximization Algorithm

Since the optimization problem is jointly convex with respect to all the parameters, one can simply take any convex optimization method to learn the parameters. However, these methods usually require hyper parameters like step size or initialization, which may significantly influence the convergence. Instead, the structure of our problem allows us to develop an efficient algorithm inspired by previous work (Zhou et al., 2013a,b; Xu et al., 2016), which leverages Minorization Maximization (MM) (Hunter and Lange, 2004) and is parameter free and insensitive to initialization.

Our algorithm utilizes Jensen's inequality to provide a lower bound for the second log-sum term in the log-likelihood given by Equation (24). More specifically, consider a set of arbitrary auxiliary variable ν_{ij}, where $1 \leq i \leq n_l$, $j = 1, 2$ and n_l is the number of link events, *i.e.*, $n_l = |\mathcal{A}|$. Further, assume these variables satisfy

$$\forall\ 1 \leq i \leq n_l : \quad \nu_{i1}, \nu_{i2} \geq 0, \quad \nu_{i1} + \nu_{i2} = 1. \tag{26}$$

The following lemma is preliminary to find a closed form update formula for the parameters.

Lemma 4 *The log-likelihood in Equation* (24) *is lower-bownded as follows:*

$$\begin{aligned}
\mathfrak{L} \geq \mathfrak{L}' = & \sum_{e_i^r \in \mathcal{E}} \mathbb{I}[u_i = s_i] \log\left(\eta_{u_i}\right) + \sum_{e_i^r \in \mathcal{E}} \mathbb{I}[u_i \neq s_i] \log(\beta_{s_i}) \\
& + \sum_{e_i^r \in \mathcal{E}} \mathbb{I}[u_i \neq s_i] \log\Big(\sum\nolimits_{v \in \mathcal{F}_{u_i(t_i)}} \Big(\kappa_{\omega_1}(t) \star \left(A_{u_i v}(t)\ dN_{v s_i}(t)\right)\Big)\Big|_{t=t_i}\Big) \\
& - \sum_{u,s \in [m]} \eta_u T + \beta_s \sum\nolimits_{v \in \mathcal{F}_u(t)} \int_0^T \kappa_{\omega_1}(t) \star \left(A_{uv}(t)\ dN_{vs}(t)\right)\ dt \\
& + \sum_{e_i^l \in \mathcal{A}} \nu_{i1} \log(\mu_{u_i}) + \nu_{i2} \log(\alpha_{u_i}) + \nu_{i2} \log\Big(\sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t)\big)\big|_{t=t_i}\Big) \\
& - \sum_{e_i^l \in \mathcal{A}} \nu_{i1} \log(\nu_{i1}) + \nu_{i2} \log(\nu_{i2}) \\
& - \sum_{u,s \in [m]} \mu_u \int_0^T (1 - A_{us}(t))\, dt + \alpha_u \int_0^T (1 - A_{us}(t))(\kappa_{\omega_2}(t) \star dN_{us}(t))\, dt.
\end{aligned} \tag{27}$$

Given the above lemma, by taking the gradient of the lower-bound with respect to the parameters, we can find the closed form updates to optimize the lower-bound:

$$\eta_u = \frac{\sum_{e_i^r \in \mathcal{E}} \mathbb{I}[u = u_i = s_i]}{T} \tag{28}$$

$$\beta_s = \frac{\sum_{e_i^r \in \mathcal{E}} \mathbb{I}[s = s_i \neq u_i]}{\sum_{u \in [m]} \mathbb{I}[u \neq s] \sum_{v \in \mathcal{F}_u(t)} \int_0^T \kappa_{\omega_1}(t) \star \left(A_{uv}(t)\ dN_{vs}(t)\right)\ dt} \tag{29}$$

$$\mu_u = \frac{\sum_{e_i^l \in \mathcal{A}} \mathbb{I}[u = u_i]\, \nu_{i1}}{\sum_{s \in [m]} \int_0^T (1 - A_{us}(t))\, dt} \tag{30}$$

$$\alpha_u = \frac{\sum_{e_i^l \in \mathcal{A}} \mathbb{I}[u = u_i]\, \nu_{i2}}{\sum_{s \in [m]} \int_0^T (1 - A_{us}(t))(\kappa_{\omega_2}(t) \star dN_{us}(t))\, dt}. \tag{31}$$

Finally, although the lower bound is valid for every choice of ν_{ij} satisfying Equation (26), by maximizing the lower bound with respect to the auxiliary variables we can make sure that the lower bound is tight:

$$\begin{array}{ll}
\text{maximize}_{\{\nu_{ij}\}} & \mathcal{L}'(\{\mu_u\}, \{\alpha_u\}, \{\eta_u\}, \{\beta_s\}, \{\nu_{ij}\}) \\
\text{subject to} & \nu_{i1} + \nu_{i2} = 1 \qquad \forall\, i : 1 \leq i \leq n_l \\
& \nu_{i0}, \nu_{i1} \geq 0 \qquad \forall\, i : 1 \leq i \leq n_l.
\end{array} \tag{32}$$

Fortunately, the above constrained optimization problem can be solved easily via Lagrange multipliers, which leads to closed form updates:

$$\nu_{i1} = \frac{\mu_{u_i}}{\mu_{u_i} + \alpha_{u_i} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t)\big)\big|_{t=t_i}} \tag{33}$$

$$\nu_{i2} = \frac{\alpha_{u_i} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t)\big)\big|_{t=t_i}}{\mu_{u_i} + \alpha_{u_i} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t)\big)\big|_{t=t_i}}. \tag{34}$$

Algorithm 4 summarizes over the learning procedure, which is guaranteed to converge to a global optimum (Hunter and Lange, 2004; Zhou et al., 2013a)

It's notable that, the maximum likelihood is prone to fall into overfitting, therefore, there is a wealth of research on how to add sparsity constraints like low rank or group sparsity regularizers (Zhou et al., 2013a) or via suitable conjugate prior (Linderman and Adams, 2014). The ideas on the next section are applicable with some modification to the case that the aforementioned ideas are utilized to improve the naive solution of the maximum likelihood.

6. Properties of Simulated Co-evolution, Networks and Cascades

In this section, we perform an empirical investigation of the properties of the networks and information cascades generated by our model. In particular, we show that our model can generate co-evolutionary retweet and link dynamics and a wide spectrum of static and temporal network patterns and information cascades.

6.1 Simulation Settings

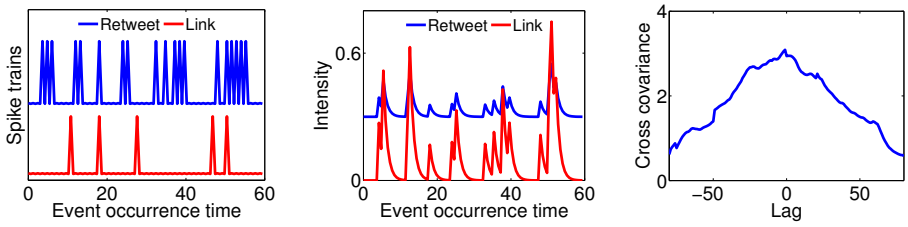
Throughout this section, if not said otherwise, we simulate the evolution of a 8,000-node network as well as the propagation of information over the network by sampling from our model using Algorithm 1. We set the exogenous intensities of the link and diffusion events to $\mu_u = \mu = 4 \times 10^{-6}$ and $\eta_u = \eta = 1.5$ respectively, and the triggering kernel parameter to $\omega_1 = \omega_2 = 1$. The parameter μ determines the independent growth of the network—roughly speaking, the expected number of links each user establishes spontaneously before time T is μT. Whenever we investigate a static property, we choose the same sparsity level of 0.001. The specific range of values we utilized in the current and the next section are such that the process remains stationary and doesn't blow up (Hawkes, 1971). Furthermore, the range of parameters are chosen such that all the results could be reported in a unified scale.

6.2 Retweet and Link Coevolution

Figures 8(a,b) visualize the retweet and link events, aggregated across different sources, and the corresponding intensities for one node and one realization, picked at random. Here, it is already apparent that retweets and link creations are clustered in time and often follow each other. Further, Figure 8(c) shows the cross-covariance of the retweet and link creation intensity, computed across multiple realizations, for the same node, *i.e.*, if $f(t)$ and $g(t)$ are two intensities, the cross-covariance is a function $h(\tau) = \int f(t+\tau)g(t)\,dt$. It can be seen that the cross-covariance has its peak around 0, *i.e.*, retweets and link creations are highly correlated and co-evolve over time. For ease of exposition, we illustrated co-evolution using one node, however, we found consistent results across nodes.

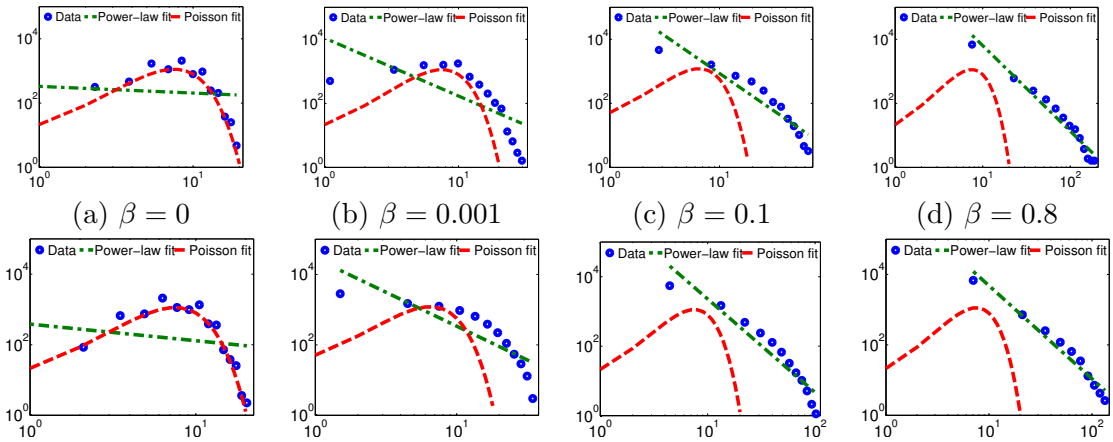
6.3 Degree Distribution

Empirical studies have shown that the degree distribution of online social networks and microblogging sites follow a power law (Chakrabarti et al., 2004; Kwak et al., 2010), and argued that it is a consequence of the rich get richer phenomena. The degree distribution of a network is a power law if the expected number of nodes m_d with degree d is given by $m_d \propto d^{-\gamma}$, where $\gamma > 0$. Intuitively, the higher the values of the parameters α and β, the closer the resulting degree distribution follows a power-law. This is because the network



(a) (b) (c)

Figure 8: Coevolutionary dynamics for synthetic data; (a) Spike trains of link and retweet events; (b) Link and retweet intensities with an exponentially decaying kernel; (c) Cross covariance of link and retweet intensities: The peak around 0 shows the coupling of retweet and link events.



(a) $\beta = 0$ (b) $\beta = 0.001$ (c) $\beta = 0.1$ (d) $\beta = 0.8$

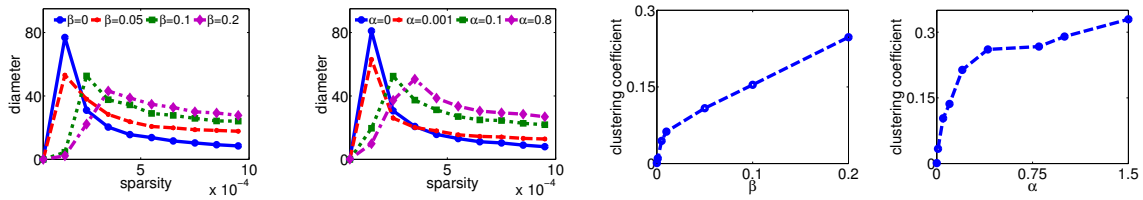
(a) $\alpha = 0$ (b) $\alpha = 0.05$ (c) $\alpha = 0.1$ (d) $\alpha = 0.2$

Figure 9: Degree distributions when network sparsity level reaches 0.001 for different β (α) values and fixed $\alpha = 0.1$ ($\beta = 0.1$) are shown in top (bottom) row. By varying the value of β (α) the degree distributions spans from random to scale-free networks illustrating the flexibility of the framework to model real-world networks.

grows more locally. Interestingly, the lower their values, the closer the distribution to an Erdos-Renyi random graph (Erdos and Rényi, 1960), because, the edges are added almost uniformly and independently without influence from the local structure. Figure 9 confirms this intuition by showing the degree distribution for different values of β and α.

6.4 Small (shrinking) Diameter

There is empirical evidence that the diameter of online social networks and microblogging sites exhibit relatively small diameter and shrinks (or flattens) as the network grows (Backstrom et al., 2012; Chakrabarti et al., 2004; Leskovec et al., 2005). Figures 10(a-b) show the diameter on the largest connected component (LCC) against the sparsity of the network



(a) Diameter, $\alpha = 0.1$ (b) Diameter, $\beta = 0.1$ (c) CC, $\alpha = 0.1$ (d) CC, $\beta = 0.1$

Figure 10: Diameter and clustering coefficient for network sparsity 0.001; (a,b) The diameter against sparsity over time for fixed $\alpha = 0.1$, and for fixed $\beta = 0.1$ respectively; (c,d) The clustering coefficient (CC) against β and α, respectively.

over time for different values of α and β. Although at the beginning, there is a short increase in the diameter due to the merge of small connected components, the diameter decreases as the network evolves. Moreover, larger values of α or β lead to higher levels of local growth in the network and, as a consequence, slower shrinkage. Here, nodes *arrive* to the network when they follow (or are followed by) a node in the largest connected component.

6.5 Clustering Coefficient

Triadic closure (Granovetter, 1973; Leskovec et al., 2008; Romero and Kleinberg, 2010) has been often presented as a plausible link creation mechanism. However, different social networks and microblogging sites present different levels of triadic closure (Ugander et al., 2013). Importantly, our method is able to generate networks with different levels of triadic closure, as shown by Figure 10(c-d), where we plot the clustering coefficient (Watts and Strogatz, 1998), which is proportional to the frequency of triadic closure, for different values of α and β.

6.6 Cascade Patterns

Our model can produce the most commonly occurring cascades structures as well as heavy-tailed cascade size and depth distributions, as observed in historical Twitter data (Goel et al., 2012). Figure 11 summarizes the results, which provide empirical evidence that the higher the α (β) value, the shallower and wider the cascades.

7. Experiments on Model Estimation and Prediction on Synthetic Data

In this section, we first show that our model estimation method can accurately recover the true model parameters from historical link and diffusion events data and then demonstrate that our model can accurately predict the network evolution and information diffusion over time, significantly outperforming two state of the art methods (Weng et al., 2013; Antoniades and Dovrolis, 2015; Myers and Leskovec, 2014) at predicting new links, and a baseline Hawkes process that does not consider network evolution at predicting new events.

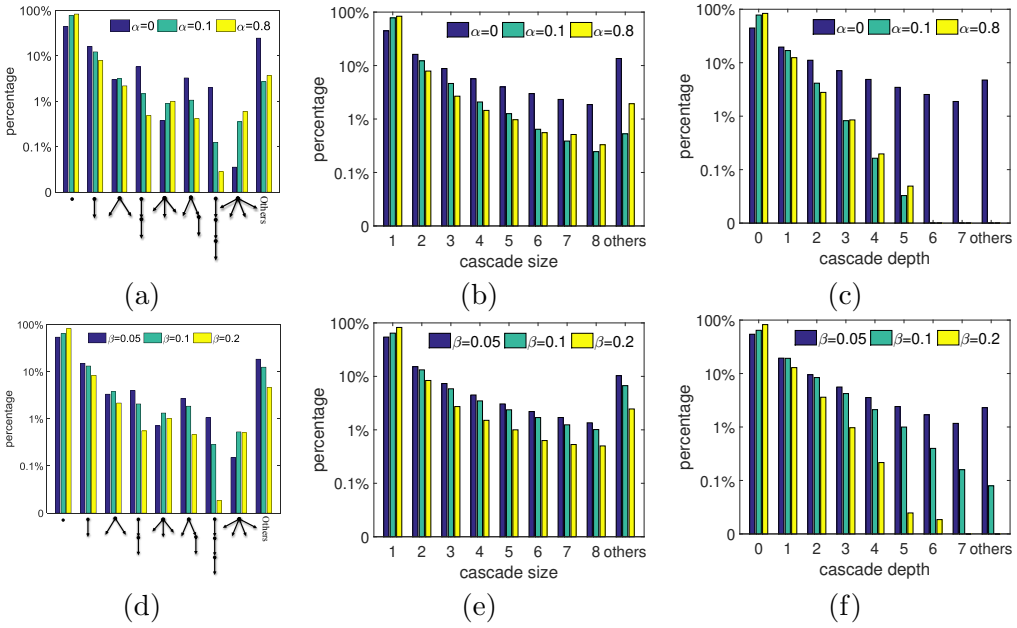


Figure 11: Distribution of cascade structure, size and depth for different values of α (β) and fixed $\beta = 0.2$ ($\alpha = 0.8$) on the top (bottom) row. The commom motifs of propagation are shown in (a) and (d).

7.1 Experimental Setup

Throughout this section, we experiment with our model considering m=400 nodes. We set the model parameters for each node in the network by drawing samples from $\mu \sim U(0, 0.0004)$, $\alpha \sim U(0, 0.1)$, $\eta \sim U(0, 1.5)$ and $\beta \sim U(0, 0.1)$. We then sample up to 60,000 link and information diffusion events from our model using Algorithm 1 and average over 8 different simulation runs.

7.2 Model Estimation

We evaluate the accuracy of our model estimation procedure via two measures: (i) the relative mean absolute error (*i.e.*, $\mathbb{E}[|x - \hat{x}|/x]$, MAE) between the estimated parameters (x) and the true parameters ($\hat{x}$), (ii) the Kendall's rank correlation coefficient between each estimated parameter and its true value, and (iii) test log-likelihood. Tables 1 and 2, and Figure 12 show the rank correlation, relative error, and the log likelihood on the test data, respectively. They show that as we feed more events into the estimation procedure, the estimation becomes more accurate.

7.3 Link Prediction

We use our model to predict the identity of the source for each test link event, given the historical events before the time of the prediction, and compare its performance with two

Table 1: Parameter Estimation Results (Rank Correlation)

# Events	α	β	η	μ
5×10^4	0.66 ± 0.05	0.82 ± 0.05	0.81 ± 0.05	0.72 ± 0.03
4×10^4	0.58 ± 0.03	0.79 ± 0.04	0.72 ± 0.05	0.65 ± 0.03
3×10^4	0.49 ± 0.05	0.76 ± 0.04	0.66 ± 0.04	0.58 ± 0.05
2×10^4	0.38 ± 0.05	0.65 ± 0.05	0.54 ± 0.05	0.47 ± 0.05
1×10^4	0.24 ± 0.03	0.43 ± 0.05	0.29 ± 0.04	0.26± 0.04

Table 2: Parameter Estimation Results (Relative Error)

# Events	α	β	η	μ
5×10^4	0.25 ± 0.05	0.1 ± 0.04	0.16 ± 0.05	0.18 ± 0.05
4×10^4	0.31 ± 0.05	0.17 ± 0.05	0.24 ± 0.03	0.28 ± 0.06
3×10^4	0.47 ± 0.04	0.32 ± 0.04	0.35 ± 0.04	0.39 ± 0.05
2×10^4	0.65 ± 0.05	0.49 ± 0.04	0.55 ± 0.04	0.57 ± 0.05
1×10^4	0.86 ± 0.05	0.71 ± 0.04	0.75 ± 0.06	0.79 ± 0.05

state of the art methods, which we denote as TRF (Antoniades and Dovrolis, 2015) and WENG (Weng et al., 2013).

Central to TRF model is the notion of TRF event: When a follower $L \in \mathcal{F}(R)$ receives a retweet of S through R, L can decide to follow S directly. They call this sequence a *Tweet-Retweet-Follow* (TRF) event, and refer to its three main actors as Speaker S, Repeater R, and Listener L. In addition to characterizing and identifying such events, TRF measures the probability of creating a link from a source at a given time by simply computing the proportion of new links created from the source over all total created links up to the given time. More specifically, it estimates the probability of new link as follows. Consider a tweet of Speaker S at time t_s . Suppose that this tweet is not retweeted by any of the followers of S in the period $[t_s, t_s + \Delta]$. Let $\Phi(S, t_s)$ be the set of followers of followers of S that are not directly following S at t_s. The exogenous probability of of following S is:

$$P_{EXO} = \frac{|L : L \in \Phi(S, t_s), L \in \mathcal{F}(S, t_s + \Delta|}{|\Phi(S, t_s)|}. \tag{35}$$

Similarly, consider a tweet of Speaker S at time t_s that has been retweeted by a follower of S, referred to as Repeater R, at time $t_r > t_s$. Let $\Phi_R(S, t_r)$ be the subset of $\Phi(S, t_r)$ that includes only followers of R. The endogenous probability of following S is:

$$P_{END} = \frac{|L : L \in \Phi_R(S, t_r), L \in \mathcal{F}(S, t_r + \Delta|}{|\Phi_R(S, t_r)|}. \tag{36}$$

And the probability of following S is simply $P_{EXO} + P_{END}$.

Weng et al. (2013) analyzed the complete graph and activity of Yahoo! Meme, to identify the effect of information diffusion on the evolution of the underlying network. WENG considers several link creation strategies and makes a prediction by combining these strategies. More specifically, they consider five link creation mechanisms and their combinations: Random: follow a randomly selected user who is not yet followed; Triadic closure: follow a randomly selected triadic node; Grandparent: follow a randomly selected grandparent;

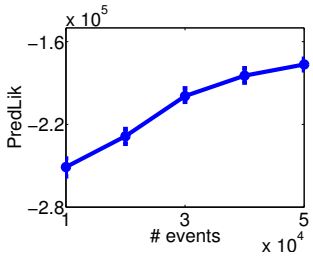
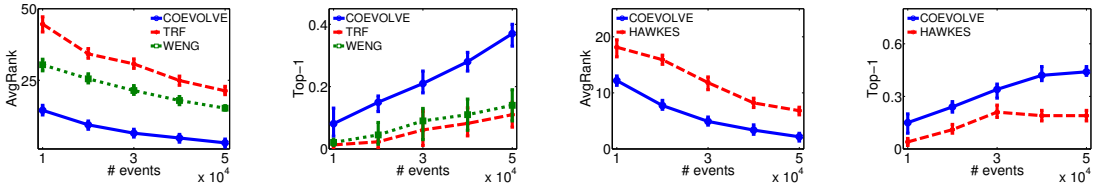


Figure 12: Performance of model estimation for the synthetic network reported in Log-likelihood on unseen test data.



(a) Links: AR (b) Links: Top-1 (c) Activity: AR Activity: Top-1

Figure 13: Prediction performance of future links and activities for a 400-node synthetic network on the test unseen data. The accuracy is reported by means of average rank (AR) and success probability that the true events rank among the top-1 events (Top-1).

Origin: follow a randomly selected origin; Traffic shortcut: follow a randomly selected grandparent or origin. Then, they parameterize the the probability of using a strategy and learned the it using maximum likelihood. Because of the contribution of multiple terms in the log-likelihood expression (due to the mixed effects of strategies) they numerically explore the values of parameters in the unit square to maximize the log-likelihood.

Here, we evaluate the performance by computing the probability of all potential links using our model, TRF and WENG. In our model, the intensity gives us a measure of how likely a link will be created at a specific point of time. The higher the intensity, the more likely the connection. Therefore, we can sort the future events based on the likelihood of happening. Then, we compute (i) the average rank of all true (test) events in our sorted list (AvgRank) and, (ii) the success probability that the true (test) events rank among the top-1 potential events at each test time (Top-1). Figure 13 summarizes the results, where we trained our model with an increasing number of events. Our model outperforms both TRF and WENG for a significant margin.

7.4 Activity Prediction

We use our model to predict the identity of the node that generates each test diffusion event, given the historical events before the time of the prediction, and compare its performance with a baseline consisting of a multi-dimensional Hawkes process without network evolution

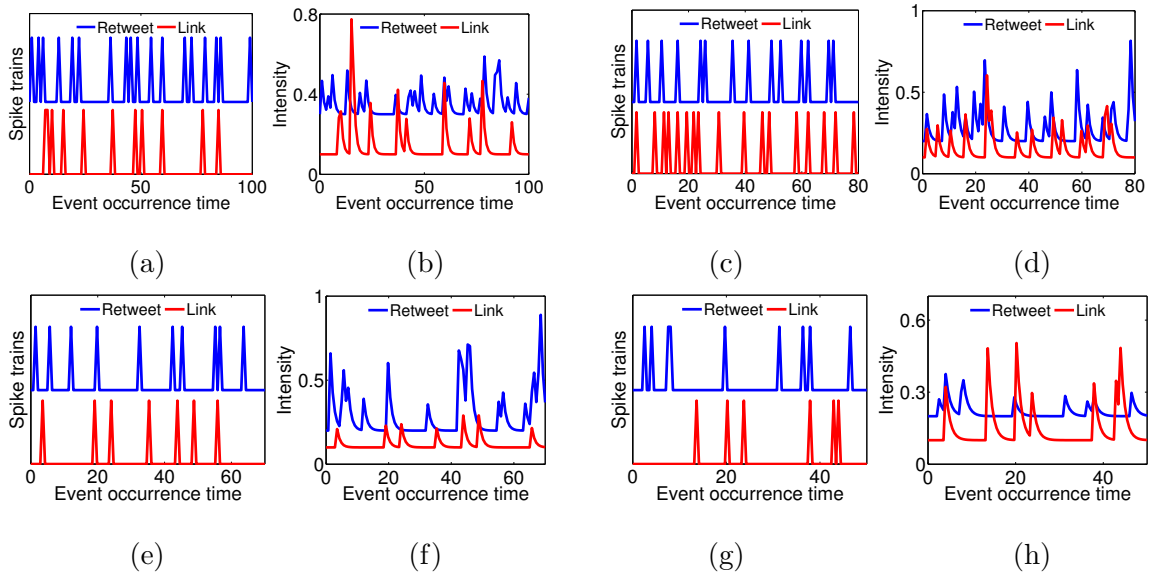


Figure 14: Link and retweet behavior of 4 typical users in the real-world data set. Panels (a,c,e,g) show the spike trains of link and retweet events and Panels (b,d,f,h) show the estimated link and retweet intensities with exponential decaying kernel.

part. For the Hawkes baseline, we take a snapshot of the network right before the prediction time, and use all historical retweeting events to fit the model and finding $(a_{uv})_{u,v\in[m]}$ and $(\eta_u)_{u\in[m]}$ similar to previous works (Farajtabar et al., 2014). Here, each user u performs activity with intensity

$$\gamma_u(t) = \eta_u + \sum_{v\in\mathcal{F}_u(t)} a_{uv}\ \kappa_{\omega_1}(t) \star dN_v(t). \tag{37}$$

Now, we have two Hawkes process based models, CoEvolve, which considers information propagation coevolved with the link creation process, and the baseline Hawkes model which only contains the information propagation part to model activities. Similar to the link prediction we used the intensity to measure how likely is an event. We can sort future events according to their probability. Then, we evaluate the performance via the same two measures as in the link prediction task and summarize the results in Figure 13 against an increasing number of training events. Please refer to the previous subsection to learn about the measures used here. The results show that, by modeling the network evolution, our model performs significantly better than the baseline.

8. Experiments on Coevolution and Prediction on Real Data

In this section, we validate our model using a large Twitter data set containing nearly 550,000 tweet, retweet and link events from more than 280,000 users (Antoniades and Dovrolis, 2015). We will show that our model can capture the co-evolutionary dynamics

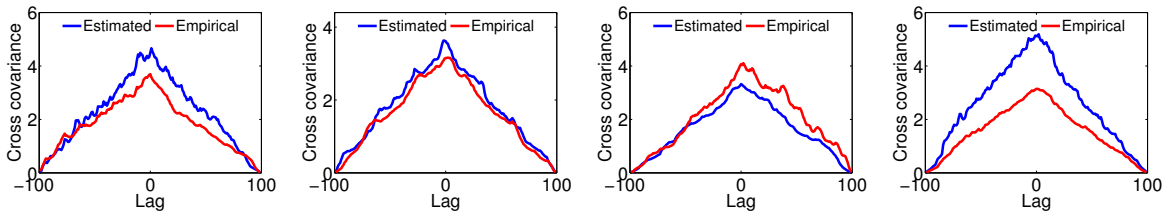


Figure 15: Empirical and simulated cross covariance of link and retweet intensities for four typical users.

and, by doing so, it predicts retweet and link creation events more accurately than several alternatives.

8.1 Data Set Description & Experimental Setup

We use a data set that contains both link events as well as tweets/retweets from millions of Twitter users (Antoniades and Dovrolis, 2015). In particular, the data set contains data from three sets of users in 20 days; nearly 8 million tweet, retweet, and link events by more than 6.5 million users. The first set of users (8,779 users) are source nodes s, for whom all their tweet times were collected. The second set of users (77,200 users) are the followers of the first set of users, for whom all their retweet times (and source identities) were collected. The third set of users (6,546,650 users) are the users that start following at least one user in the first set during the recording period, for whom all the link times were collected.

In our experiments, we focus on all events (and users) during a 10-day period (Sep 21 2012 - 30 Sep 2012) and used the information before Sep 21 to construct the initial social network (original links between users). We model the co-evolution in the second 10-day period using our framework. More specifically, in the coevolution modeling, we have 5,567 users in the first layer who post 221,201 tweets. In the second layer 101,465 retweets are generated by the whole 77,200 users in that interval. And in the third layer we have 198,518 users who create 219,134 links to 1978 users (out of 5567) in the first layer.

We split events into a training set (covering 85% of the retweet and link events) and a test set (covering the remaining 15%) according to time, *i.e.*, all events in the training set occur earlier than those in the test set. We then use our model estimation procedure to fit the parameters from an increasing proportion of events from the training data.

8.2 Retweet and Link Coevolution

Figures 14 visualizes the retweet and link events, aggregated across different targets, and the corresponding intensities given by our trained model for four source nodes, picked at random. Here, it is already apparent that retweets (of his posts) and link creations (to him) are clustered in time and often follow each other, and our fitted model intensities successfully track such behavior. Further, Figure 15 compares the cross-covariance between the empirical retweet and link creation intensities and between the retweet and link creation intensities given by our trained model, computed across multiple realizations, for the same nodes. For all nodes, the similarity between both cross-covariances is striking and both has

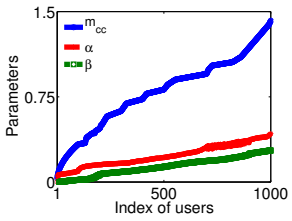


Figure 16: Empirical cross covariance and learned model parameters for 1,000 users, picked at random. The x-axis indexes the users while their α, β and the empirical cross covariance are shown in different colors on the y-axis.

their peak around 0, *i.e.*, retweets and link creations are highly correlated and co-evolve over time. For ease of exposition, as in Section 6, we illustrated co-evolution using four nodes, however, we found consistent results across nodes.

To further verify that our model can capture the coevolution, we compute the average value of the empirical cross covariance function, denoted by m_{cc}, per user. Intuitively, one could expect that our model estimation method should assign higher α and/or β values to users with high m_{cc}. Figure 16 confirms this intuition on 1,000 users, picked at random. Whenever a user has high α and/or β value, she exhibits a high cross covariance between her created links and retweets.

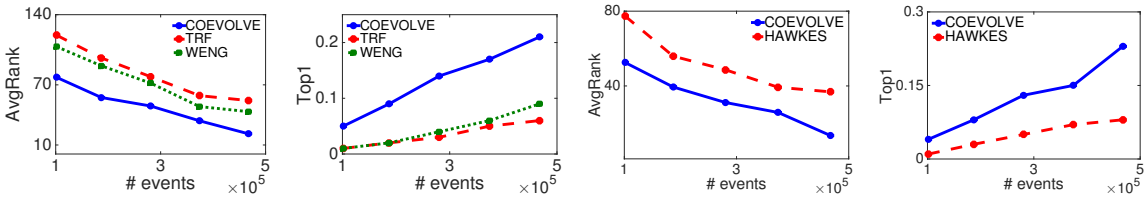
8.3 Link Prediction

We use our model to predict the identity of the source for each test link event, given the historical (link and retweet) events before the time of the prediction, and compare its performance with the same two state of the art methods as in the synthetic experiments, TRF (Antoniades and Dovrolis, 2015) and WENG (Weng et al., 2013).

We evaluate the performance by computing the probability of all potential links using different methods, and then compute (i) the average rank of all true (test) events (AvgRank) and, (ii) the success probability (SP) that the true (test) events rank among the top-1 potential events at each test time (Top-1). We summarize the results in Figure 17(a-b), where we consider an increasing number of training retweet/tweet events. Our model outperforms TRF and WENG consistently. For example, for $8 \cdot 10^4$ training events, our model achieves a SP 2.5x times larger than TRF and WENG.

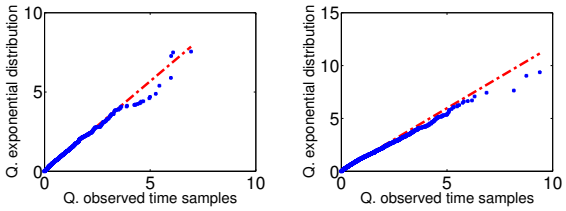
8.4 Activity Prediction

We use our model to predict the identity of the node that generates each test diffusion event, given the historical events before the time of the prediction, and compare its performance with a baseline consisting of a Hawkes process without network evolution. For the Hawkes baseline, we take a snapshot of the network right before the prediction time, and use all historical retweeting events to fit the model. Here, we evaluate the performance the via the same two measures as in the link prediction task and summarize the results in Figure 17(c-d) against an increasing number of training events. The results show that, by modeling the co-evolutionary dynamics, our model performs significantly better than the baseline.



(a) Links: AR (b) Links: Top-1 (c) Activity: AR Activity: Top-1

Figure 17: Prediction performance in the Twitter data set by means of average rank (AR) and success probability that the true (test) events rank among the top-1 events (Top-1).



(a) Link process (b) Retweet process

Figure 18: Quantile plots of the intensity integrals from the real link and retweet event time. The alignment of blue dots (empirical) and red ones (theoretical) shows the suitability of our model in representing observed traces of activities in Twitter.

8.5 Model Checking

Given all the subsequent event times generated using a Hawkes process, *i.e.*, t_i and t_{i+1}, according to the time changing theorem (Daley and Vere-Jones, 2007), the intensity integrals $\int_{t_i}^{t_{i+1}} \lambda(t)\, dt$ should conform to the unit-rate exponential distribution. Figure 18 presents the quantiles of the intensity integrals computed using intensities with the parameters estimated from the real Twitter data against the quantiles of the unit-rate exponential distribution. It clearly shows that the points approximately lie on the same line, giving empirical evidence that a Hawkes process is the right model to capture the real dynamics.

9. Related Work

In this section, we survey related works in modeling temporal networks followed by a subsection on co-evolution dynamics. Next, we review the literature on information diffusion models. Finally, we conclude this section by works that are closely related and are developed for almost the same goal.

Temporal Networks. Much effort has been devoted to modeling the evolution of social networks (Phan and Airoldi, 2015; Doreian and Stokman, 2013; Wang et al., 2011). Of the proposed methods in characterizing link creation, triadic closure (Granovetter, 1973) is a simple but powerful principle to model the evolution based on shared friends. Modeling timing and rich features of social interactions has been attracting increasing interest

in the social network modeling community (Goldenberg et al., 2010). However, most of these models use timing information as discrete indices. The dynamics of the resulting time-discretized model can be quite sensitive to the chosen discretization time steps; Too coarse a discretization will miss important dynamic features of the process, and too fine a discretization will increase the computational and inference costs of the algorithms. In contrast, the events we try to model tend to be asynchronous with a number of different time scales. Recently, Snijders and Luchini (2006) used a Cox-intensity Poisson model with exponential random graphs to model friendship dynamics. Brandes et al. (2009) extended this model to the temporal sequence of interactions that take place in the social network, but with insufficient model flexibility, and limited scalability. However, these works largely fail to model the interdependency between events generated by different users, which is one of the focuses of our proposed framework. Most of this line of work is summarized in a recent survey (Holme, 2015), with a short section devoted to point process based approaches.

Co-evolution Dynamics. In machine learning and several other communities, both the dynamics on the network and the dynamics of the network have been extensively studied, and combining the two is a natural next step. For example, Bhattacharya et al. (2015) claimed that content generation in social networks is influenced not just by their personal features like age and gender, but also by their social network structure. Furthermore, research has been done to address the co-evolution problems, for example, in the complex network literature, under the name of *adaptive system* (Gross and Blasius, 2008). The main premise is that the evolution of the topology depends on the dynamics of the nodes in the network, and a feedback loop can be created between the two, which allows dynamical exchange of information. In a different context, epidemiologists have found that nodes may rewire their links to try to avoid contact with the infected ones (Gross et al., 2006). Co-evolutionary models have been also developed for collective opinion formation, investigating whether the coevolutionary dynamics will eventually lead to consensus or fragmentation of the population (Zschaler et al., 2012). However, this line of research tends to be less data-driven.Moreover, although the general nonlinear dynamic-system based methods usually address co-evolutionary phenomena that are macroscopic in nature, they lack the inference power of statistical generative models which are more adapted to teasing out microscopic details from the data. Finally, we would also like to mention a different line of research exemplified by the actor-oriented models developed by Snijders (2014), where a continuous-time Markov chain on the space of directed networks is specified by local node-centric probabilistic link change rules, and MCMC and method of moments are used for parameter estimation. Hawkes processes we used are generally non-Markovian and making use of event history far into the past.

Information Diffusion. The presence of timing information in event data and the ability to model such information bring up the interesting question of how to use the learned model for time-sensitive inference or decision making. Furthermore, the development of online social networks has attracted a lot of empirical studies of the online influence patterns of online communities (Guo et al., 2015), micro blogs (Bakshy et al., 2011) and so on. However, these works usually consider only relatively simple models for the influence, which may not be very predictive. For more mathematically oriented works, based on information cascades (a special case of asynchronous event data) from social networks, discrete-time

diffusion models have been fitted to the cascades (Goyal et al., 2010a) and used for decision making, such as identifying influencer (Agarwal et al., 2008), maximizing information spread (Rodriguez and Schölkopf, 2012), and marketing planing (Richardson and Domingos, 2002; Bhagat et al., 2012). Several recent experimental comparisons on both synthetic and real world data showed that continuous-time models yield significant improvement in settings such as recovering hidden diffusion network topologies from cascade data (Du et al., 2012; Gomez-Rodriguez et al., 2011; Yang and Zha, 2013; Zarezade et al., 2015), predicting the timings of future events (Rodriguez et al., 2013), finding source of information cascades (Farajtabar et al., 2015a). Besides this, Point process modeling of activity in network is becoming increasingly popular (Parikh et al., 2012; Hall and Willett, 2016; Farajtabar et al., 2017). These time-sensitive modeling and decision making problems can usually be framed into optimization problems and are usually difficult to solve. This brings up interesting optimization problems, such as efficient submodular function optimization with provable guarantees (Goyal et al., 2010b; Kempe et al., 2003), sampling methods (Lian et al., 2014; Gunawardana et al., 2011) for inference and prediction, and convex framework proposed in (Farajtabar et al., 2014; Karimi et al., 2016) to make decisions to shape the activity to a variety of objectives and intervene in networks (Farajtabar et al., 2017). Furthermore, the high dimensional nature of modern event data makes the evaluation of objective function of the optimization problem even more expensive. Therefore, more accurate modeling and sophisticated algorithm needed to be designed to tackle the challenges posed by modern event data applications.

The work most closely related to ours is the empirical study of information diffusion and network evolution (Gross and Blasius, 2008; Singer et al., 2012; Weng et al., 2013; Antoniades and Dovrolis, 2015; Myers and Leskovec, 2014). Among them, (Weng et al., 2013) was the first to show experimental evidence that information diffusion influences network evolution in microblogging sites both at system-wide and individual levels. In particular, they studied *Yahoo! Meme*, a social micro-blogging site similar to Twitter, which was active between 2009 and 2012, and showed that the likelihood that a user u starts following a user s increases with the number of messages from s seen by u. Antoniades and Dovrolis (2015) investigated the temporal and statistical characteristics of retweet-driven connections within the Twitter network and then identified the number of retweets as a key factor to infer such connections, in agreement with (Weng et al., 2013). Besides link prediction, they argue about which potential network motifs TRF events can lead to in the network structure. Myers and Leskovec (2014) showed that the Twitter network can be characterized by steady rates of change, interrupted by sudden bursts of new connections, triggered by retweet cascades. They also developed a method to predict which retweets are more likely to trigger these bursts. Finally, Tran et al. (2015) utilized multivariate Hawkes process to establish a connection between temporal properties of activities and the structure of the network. In contrast to our work they studied the static properties, *e.g.*, community structure and inferred the latent clusters using the observed activities.

However, there are fundamental differences between the above-mentioned studies and our work. First, they only characterize the effect that information diffusion has on the network dynamics, but not the bidirectional influence. In contrast, our probabilistic generative model takes into account the bidirectional influence between information diffusion and network dynamics. Second, previous studies are mostly empirical and only make binary

predictions on link creation events. For example, the work of (Weng et al., 2013; Antoniades and Dovrolis, 2015) predict whether a new link will be created based on the number of retweets; and, (Myers and Leskovec, 2014) predict whether a burst of new links will occur based on the number of retweets and users' similarity. However, our model is able to learn parameters from real world data, and predict the precise timing of both diffusion and new link events.

10. Conclusion and Future Works

In this work, we proposed a joint continuous-time model of information diffusion and network evolution, which can capture the coevolutionary dynamics, can mimic the most common static and temporal network patterns observed in real-world networks and information diffusion data, and can predict the network evolution and information diffusion more accurately than previous state-of-the-arts. Using point processes to model intertwined events in information and social networks opens up many interesting venues for future. Our current model is just a show-case of a rich set of possibilities offered by a point process framework, which have been rarely explored before in large scale social network modeling. There are quite a few directions that remain as future work and are very interesting to explore. For example:

- A large and diverse range of point processes can also be used instead in the framework and augment the current model without changing the efficiency of simulation and the convexity of parameter estimation.
- We can incorporate features from previous state of the diffusion or network structure. For example, one can model information overload by adding a nonlinear transfer function on top of the diffusion intensity, or model peer pressure by adding a nonlinear transfer function depending on the number of neighbors.
- There are situations that the processes are naturally evolve in different time scales. For example, link dynamics is meaningful in the scale of days, however, the resolution in which information propagation occurs is usually in hours or even minutes. Developing an efficient mechanism to account for heterogeneity in time resolution would improve the model's ability to predict.
- We may augment the framework to allow time-varying parameters. The simulation would not be affected and the estimation of time-varying interaction can still be carried out via a convex optimization problem (Zhou et al., 2013b).
- Alternatively, one can use different triggering kernels for the Hawkes processes and learn them to capture finer details of temporal dynamics.

Acknowledgments

The authors would like to thank Demetris Antoniades and Constantine Dovrolis for providing them with the data set. The research was supported in part by NSF/NIH BIGDATA 1R01GM108341, ONR N00014-15-1-2340, NSF IIS-1218749, NSF CAREER IIS-1350983.

Appendix A. Notations

Table 3: Table of Notations of the COEVOLVE framework

Notation	Description
m	number of nodes in the network
$e^r = (u, s, t)$	retweet event: node u retweets at time t a tweet by source node s
$e^l = (u, s, t)$	link event: node u create at time t a link to source node s
$\mathcal{H}^r_{us}(t)$	history of retweet events by node u up to time t due to source s
$\mathcal{H}^r(t)$	history of all retweet events up to time t
$\mathcal{H}^l(t)$	history of all link events up to time t
$N_{us}(t)$	count process of the number of retweets of u due to source s up to time t
$\boldsymbol{N}(t)$	count process matrix of all pairs $\boldsymbol{N}(t) := (N_{us}(t))_{u,s\in[m]}$
$d\boldsymbol{N}(t)$	differential of retweet process matrix $d\boldsymbol{N}(t) := (dN_{us}(t))_{u,s\in[m]}$
$A_{us}(t)$	survival process indicating whether u is directly following s at time t
$\boldsymbol{A}(t)$	survival process matrix of all paris $\boldsymbol{A}(t) := (A_{us}(t))_{u,s\in[m]}$
$d\boldsymbol{A}(t)$	differential of link creation process matrix $d\boldsymbol{A}(t) := (dA_{us}(t))_{u,s\in[m]}$
$\gamma^*_{us}(t)$	conditional intensity function of user u retweet an event due to source s
$\boldsymbol{\Gamma}^*(t)$	matrix of information diffusion intensity functions $\Gamma(t) = (\gamma^*_{us}(t))_{u,s\in[m]}$
$\mathbb{I}[\cdot]$	indicator function
$\mathcal{F}_u(t)$	set of followers of u at time t
η_u	intensity of original tweets by a user u on his own initiative
β_s	amount of increase in retweeting intensity when a tweet by source s hits
$\kappa_\omega(t)$	decaying kernel with bandwidth ω; $\kappa_\omega(t) = e^{-\omega t}$
$\lambda^*_{us}(t)$	conditional intensity function for user u follows source s at time t
$\boldsymbol{\Lambda}^*(t)$	matrix of link creation intensity functions $\boldsymbol{\Lambda}^*(t) = (\lambda^*_{us}(t))_{u,s\in[m]}$
μ_u	intensity of node u deciding to follow a source at her own initiative
α_u	amount of increase in linking intensity when a tweet by the source hits
$\mathcal{E}$	collection of retweet events observed
$\mathcal{A}$	collection of link creation events
$\mathfrak{L}$	likelihood of observation
ν_{ij}	auxiliary variable for lower-bounding the log-likelihood

Appendix B. Basic Simulation Algorithm

In this section, we revisit Ogata's algorithm in more details. Consider a U-dimensional point process in which each dimension u is characterized by a conditional intensity function $\lambda_u^*(t)$.

Ogata's algorithm starts with summing the intensities, $\lambda_{sum}^*(\tau) = \sum_{u=1}^{U} \lambda_u^*(\tau)$. Then, assuming we have simulated up to time t, the next sample time, t', is the first event drawn from the non-homogenous Poisson process with intensity $\lambda_{sum}^*(\tau)$ which begins at time t. Here, the algorithm exploits that, given a fixed history, the Hawkes Process is a non-homogenous Poisson process, which runs until the next event happens. Then, the new event will result in an update of the intensities and a new non-homogenous Poisson process starts until the next event.

It can be shown that the waiting time of a non-homogeneous Poisson process is an exponentially distributed random variable with rate equal to integral of the intensity (Ross, 2011), *i.e.* $s \sim Exponential\left(\int_t^{t+s} \lambda_{sum}^*(\tau)\, d\tau\right)$. Thus, the next sample time can be computed as

$$t' = \underbrace{t}_{\text{current time}} + \underbrace{s.}_{\text{waiting time for the first event}} \tag{38}$$

Sampling from a non-homogenous Poisson process is not straight-forward, therefore, Ogata's algorithm uses rejection sampling with a homogenous Poisson process as the proposal distribution. More in detail, given $\hat{\lambda} = \max_{t \le \tau \le T} \lambda_{sum}^*(\tau)$, t' is the time of first event of homogenous Poisson Process with rate $\hat{\lambda}$. Then, we accept the sample time with probability $\lambda_{sum}^*(t')/\hat{\lambda}$. Finally, the dimension firing the event is determined by sampling proportionally to the contribution of the intensity of that user to the total intensity, *i.e.*, $\lambda_u^*(t')/\lambda_{sum}^*(t')$ for $1 \le u \le U$. This procedure is iterated until we reach the end of simulation time T. Algorithm 5 presents the complete procedure.

Ogata's algorithm would scale poorly with the dimension of the process, because, after each sample, we would need to re-evaluate the affected intensities and find the upper bound. As a consequence, a naive implementation to draw n samples require $O(Un^2)$ time complexity, where U is the number of dimensions. This is because for each sample we need to find the new summation of intensities, which involves $O(U)$ individual ones, each taking $O(n)$ time to accumulate over this history. In our social networks application, we have $m^2 - m$ point processes for link creation and m^2 ones for retweeting, *i.e.*, $U = O(m^2)$. Therefore, Ogata's algorithm takes $O(m^2 n^2)$ time complexity.

Algorithm 5 Ogata's Algorithm

Input: U dimensional Hawkes process $\{\lambda_u^*(t)\}_{u=1...U}$, Due time: T
Output: Set of events: $\mathcal{H} = \{(t_1, u_1), \ldots, (t_n, u_n)\}$
$t \leftarrow 0$
$i \leftarrow 0$
while $t < T$ **do**
    $\lambda^*_{sum}(\tau) \leftarrow \sum_{u=1}^{U} \lambda_u^*(\tau)$ } Sampling next event time
    $\hat{\lambda} \leftarrow \max_{t \le \tau \le T} \lambda^*_{sum}(\tau)$
    $s \sim Exponential(\hat{\lambda})$
    $t' \leftarrow t + s$
    if $t' \geq T$ **then**
        break
    end if
    $\bar{\lambda} \leftarrow \lambda^*_{sum}(t')$ } Rejection test
    $d \sim Uniform(0, 1)$
    if $d \times \hat{\lambda} > \bar{\lambda}$ **then**
        $t \leftarrow t'$
        Goto 6
    end if
    $S \leftarrow 0$ } Attribution test
    $d \sim Uniform(0, 1)$
    for $u \leftarrow 1$ to U **do**
        $S \leftarrow S + \lambda_u^*(t')$
        if $S \geq d$ **then**
            $i \leftarrow i + 1$
            $u_i \leftarrow u$
            $t_i \leftarrow t'$
            $t \leftarrow t'$
            Goto 6
        end if
    end for
    Given the new event just sampled update intensity functions $\lambda_u^*(\tau)$
end while

Appendix C. Proofs

Lemma 1 *Assume we have U independent non-homogeneous Poisson processes with intensity $\lambda_1^*(\tau), \ldots, \lambda_U^*(\tau)$. Take random variable τ_u equal to the time of process u's first event after time t. Define $\tau_{min} = \min_{1 \le u \le U} \{\tau_u\}$ and $u_{min} = \operatorname{argmin}_{1 \le u \le U} \{\tau_u\}$. Then,*

(a) τ_{min} is the first event after time t of the Poisson process with intensity $\lambda_{sum}^*(\tau)$. In other words, τ_{min} has the same distribution as the next event (t') in Ogata's algorithm.

(b) u_{min} follows the conditional distribution $\mathbb{P}(u_{min} = u | \tau_{min} = x) = \frac{\lambda_U^*(x)}{\lambda_{sum}^*(x)}$. I.e. the dimension firing the event comes from the same distribution as the one in Ogata's algorithm.

Proof (a) The waiting time of the first event of a dimension u is exponentially distributed[1] random variable (Ross, 2011); *i.e.*, $\tau_u - t \sim Exponential\left(\int_t^{t+\tau_u} \lambda_u^*(\tau)\, d\tau\right)$. We have:

$$\begin{aligned}
\mathbb{P}(\tau_{min} \le x | x > t) &= 1 - \mathbb{P}(\tau_{min} > x | x > t) = 1 - \mathbb{P}(\min(\tau_1, \ldots, \tau_U) > x | x > t) \\
&= 1 - \mathbb{P}(\tau_1 > x, \ldots, \tau_U > x | x > t) = 1 - \prod_{u=1}^{U} \mathbb{P}(\tau_u > x | x > t) \\
&= 1 - \prod_{u=1}^{U} \exp\left(-\int_t^{t+x} \lambda_u^*(\tau)\, d\tau\right) = 1 - \exp\left(-\int_t^{t+x} \lambda_{sum}^*(\tau)\, d\tau\right).
\end{aligned} \tag{39}$$

Therefore, $\tau_{min} - t$ is exponentially distributed with parameter $\int_t^{\tau_{min}} \lambda_{sum}^*(\tau)\, d\tau$ which can be seen as the first event of a non-homogenous poisson process with intensity $\lambda_{sum}^*(\tau)$ after time t.

(b) To find the distribution of u_{min} we have

$$\begin{aligned}
\mathbb{P}(u_{min} = u | \tau_{min} = x) &= \lambda_u^*(x) \exp\left(-\int_t^{t+x} \lambda_u^*(\tau)\, d\tau\right) \prod_{v \ne u} \exp\left(-\int_t^{t+x} \lambda_v^*(\tau)\, d\tau\right) \\
&= \lambda_u^*(x) \prod_{v} \exp\left(-\int_t^{t+x} \lambda_v^*(\tau)\, d\tau\right).
\end{aligned} \tag{40}$$

After normalization we get $\mathbb{P}(u_{min} = u | \tau_{min} = x) = \frac{\lambda_U^*(x)}{\lambda_{sum}^*(x)}$.

■

Theorem 2 *The sequence of samples from Ogata's algorithm and our proposed algorithm follow the same distribution.*

Proof Using the chain rule the probability of observing $\mathcal{H}_T = \{(t_1, u_1), \ldots, (t_n, u_n)\}$ is written as:

$$\mathbb{P}\{(t_1, u_1), \ldots, (t_n, u_n)\} = \prod_{i=1}^{n} \mathbb{P}\{(t_i, u_i) | (t_{i-1}, u_{i-1}), \ldots, (t_1, u_1)\} = \prod_{i=1}^{n} \mathbb{P}\{(t_i, u_i) | \mathcal{H}_{t_i}\}.$$

By fixing the history up to some time, say t_i, all dimensions of multivariate Hawkes process become independent of each other (until next event happens). Therefore, the above lemma

1. If random variable X is exponentially distributed with parameter r, then $f_X(x) = r \exp(-rx)$ is its probability distribution function and $F_X(x) = 1 - \exp(-rx)$ is the cumulative distribution function.

can be applied to show that the next sample time from Ogata's algorithm and the proposed one come from the same distribution, *i.e.*, for every i, $\mathbb{P}\{(t_i, u_i)|\mathcal{H}_{t_i}\}$ is the same for both algorithms. Thus, the multiplication of individual terms is also equal for both. This will prove the theorem. ■

Theorem 3 *The optimization problem defined by Equation* (25) *is jointly convex.*

Proof We expand the likelihood by replacing the intensity functions into Equation (24):

$$\begin{aligned}
\mathfrak{L} = &\sum_{e_i^r \in \mathcal{E}} \log \left(\mathbb{I}[u_i = s_i]\, \eta_{u_i} + \mathbb{I}[u_i \neq s_i]\, \beta_{s_i} \sum\nolimits_{v \in \mathcal{F}_{u_i(t_i)}} \Big(\kappa_{\omega_1}(t) \star (A_{u_i v}(t)\ dN_{vs_i}(t)) \Big) \Big|_{t=t_i} \right) \\
&- \sum_{u,s \in [m]} \mathbb{I}[u = s]\, \eta_u \int_0^T dt + \mathbb{I}[u \neq s]\, \beta_s \sum\nolimits_{v \in \mathcal{F}_u(t)} \int_0^T \kappa_{\omega_1}(t) \star (A_{uv}(t)\ dN_{vs}(t))\ dt \\
&+ \sum_{e_i^l \in \mathcal{A}} \log \left(\mu_{u_i} + \alpha_{u_i} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t) \big) \big|_{t=t_i} \right) \\
&- \sum_{u,s \in [m]} \mu_u \int_0^T (1 - A_{us}(t))\, dt + \alpha_u \int_0^T (1 - A_{us}(t)) \big(\sum_{v \in \mathcal{F}_u(t)} \kappa_{\omega_2}(t) \star dN_{vs}(t) \big)\, dt.
\end{aligned} \tag{41}$$

If we stack all parameters in a vector $\boldsymbol{x} = (\{\mu_u\}, \{\alpha_u\}, \{\eta_u\}, \{\beta_s\})$, one can easily notice that the log-likelihood $\mathfrak{L}$ can be written as $\sum_j \log(\boldsymbol{a}_j^\top \boldsymbol{x}) - \sum_k \boldsymbol{b}_k^\top \boldsymbol{x}$, which is clearly a concave function with respect to $\boldsymbol{x}$ (Boyd and Vandenberghe, 2004), and thus $-\mathfrak{L}$ is convex. Moreover, the constraints are linear inequalities and thus the domain is a convex set. This completes the proof for convexity of the optimization problem. ■

Lemma 4 *The log-likelihood in Equation* (24) *is lower-bounded as follows:*

$$\begin{aligned}
\mathfrak{L} \geq \mathfrak{L}' = &\sum_{e_i^r \in \mathcal{E}} \mathbb{I}[u_i = s_i]\ \log(\eta_{u_i}) + \sum_{e_i^r \in \mathcal{E}} \mathbb{I}[u_i \neq s_i]\ \log(\beta_{s_i}) \\
&+ \sum_{e_i^r \in \mathcal{E}} \mathbb{I}[u_i \neq s_i]\ \log\big(\sum\nolimits_{v \in \mathcal{F}_{u_i(t_i)}} \Big(\kappa_{\omega_1}(t) \star (A_{u_i v}(t)\ dN_{vs_i}(t)) \Big) \Big|_{t=t_i} \big) \\
&- \sum_{u,s \in [m]} \eta_u T + \beta_s \sum\nolimits_{v \in \mathcal{F}_u(t)} \int_0^T \kappa_{\omega_1}(t) \star (A_{uv}(t)\ dN_{vs}(t))\ dt \\
&+ \sum_{e_i^l \in \mathcal{A}} \nu_{i1} \log(\mu_{u_i}) + \nu_{i2} \log(\alpha_{u_i}) + \nu_{i2} \log\big(\sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t) \big) \big|_{t=t_i} \big) \\
&- \sum_{e_i^l \in \mathcal{A}} \nu_{i1} \log(\nu_{i1}) + \nu_{i2} \log(\nu_{i2}) \\
&- \sum_{u,s \in [m]} \mu_u \int_0^T (1 - A_{us}(t))\, dt + \alpha_u \int_0^T (1 - A_{us}(t)) (\kappa_{\omega_2}(t) \star dN_{us}(t))\, dt.
\end{aligned} \tag{42}$$

Proof We can lower-bound the logarithm in the log-likelihood (Equation (41)) using Jensen's inequality as follows:

$$\begin{aligned}
&\log\left(\mu_{u_i} + \alpha_{u_i} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t)\big)\big|_{t=t_i}\right) \\
&= \log\left(\nu_{i1}\frac{\mu_{u_i}}{\nu_{i1}} + \nu_{i2}\frac{\alpha_{u_i}}{\nu_{i2}} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t)\big)\big|_{t=t_i}\right) \\
&\geq \nu_{i1}\log\left(\frac{\mu_{u_i}}{\nu_{i1}}\right) + \nu_{i2}\log\left(\frac{\alpha_{u_i}}{\nu_{i2}} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t)\big)\big|_{t=t_i}\right) \\
&\geq \nu_{i1}\log(\mu_{u_i}) + \nu_{i2}\log(\alpha_{u_i}) + \nu_{i2}\log\left(\sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t)\big)\big|_{t=t_i}\right) \\
&\quad - \nu_{i1}\log(\nu_{i1}) - \nu_{i2}\log(\nu_{i2}).
\end{aligned} \tag{43}$$

■

Replacing the logarithm with the lower-bound gets the result.

Appendix D. More Properties of the Coevolution Model

In this section we complement the properties of simulated coevolved networks and cascades by visualizing the outcomes, *i.e.*, the networks and cascades.

D.1 Network Visualization

Figure 19 visualizes several snapshots of the largest connected component (LCC) of two 300-node networks for two particular realizations of our model, under two different values of β. In both cases, we used $\mu = 2 \times 10^{-4}$, $\alpha = 1$, and $\eta = 1.5$. The top two rows correspond to $\beta = 0$ and represent one end of the spectrum, *i.e.*, Erdos-Renyi random network. Here, the network evolves uniformly. The bottom two rows correspond to $\beta = 0.8$ and represent the other end, *i.e.*, scale-free networks. Here, the network evolves locally, and clusters emerge naturally as a consequence of the local growth. They are depicted using a combination of forced directed and Fruchterman Reingold layout with Gephi[2]. Moreover, the figure also shows the retweet events (from others as source) for two nodes, A and B, on the bottom row. These two nodes arrive almost at the same time and establish links to two other nodes. However, node A's followees are more central, therefore, A is being exposed to more retweets. Thus, node A performs more retweets than B does. It again shows how information diffusion is affected by network structure. Overall, this figure clearly illustrates that by careful choice of parameters we can generate networks with a very different structure.

2. http://gephi.github.io/

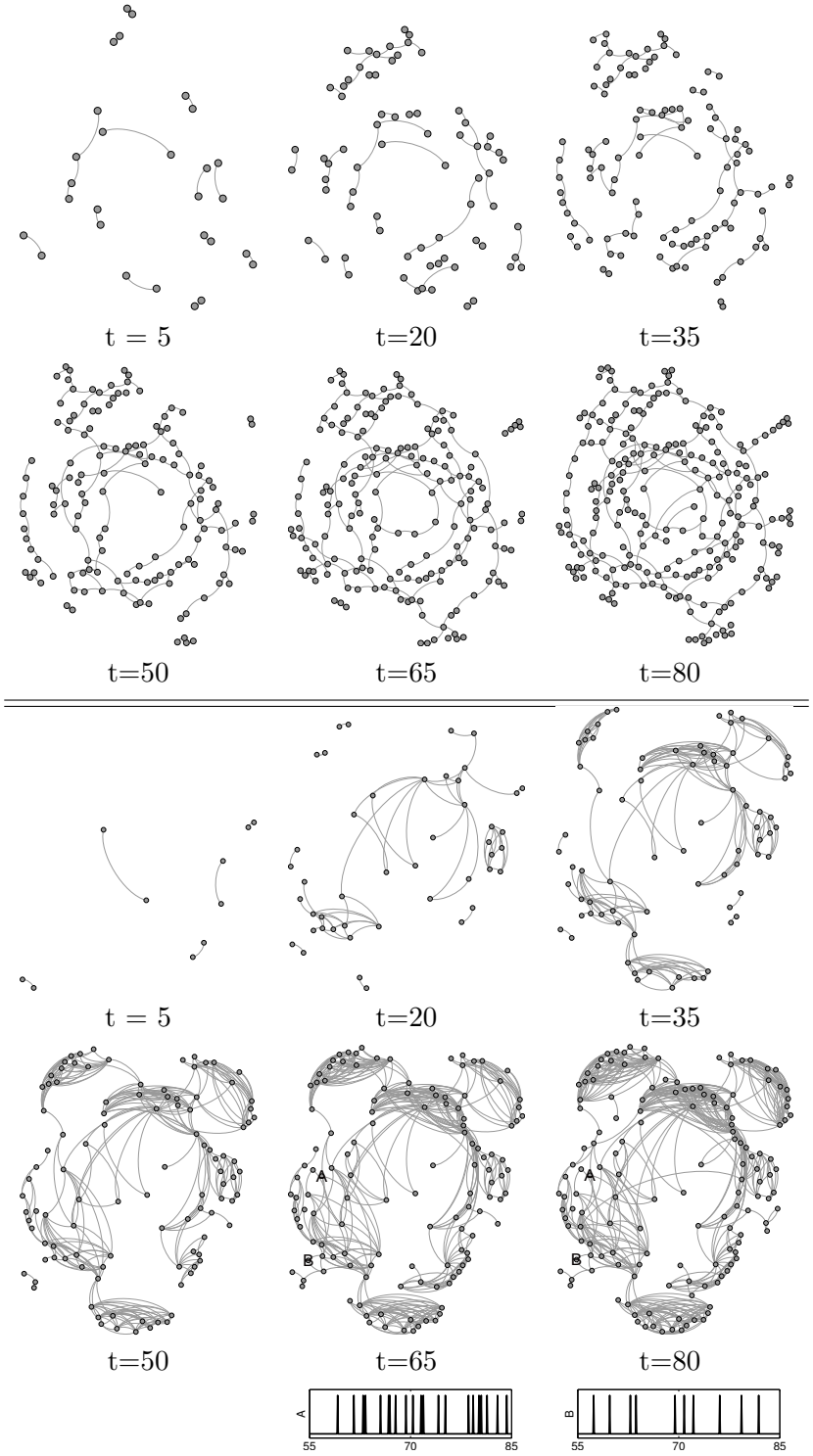


Figure 19: Evolution of two networks: one with $\beta = 0$ (1st and 2nd rows) and another one with $\beta = 0.8$ (3rd and 4th rows), and spike trains of nodes A and B (5th row).

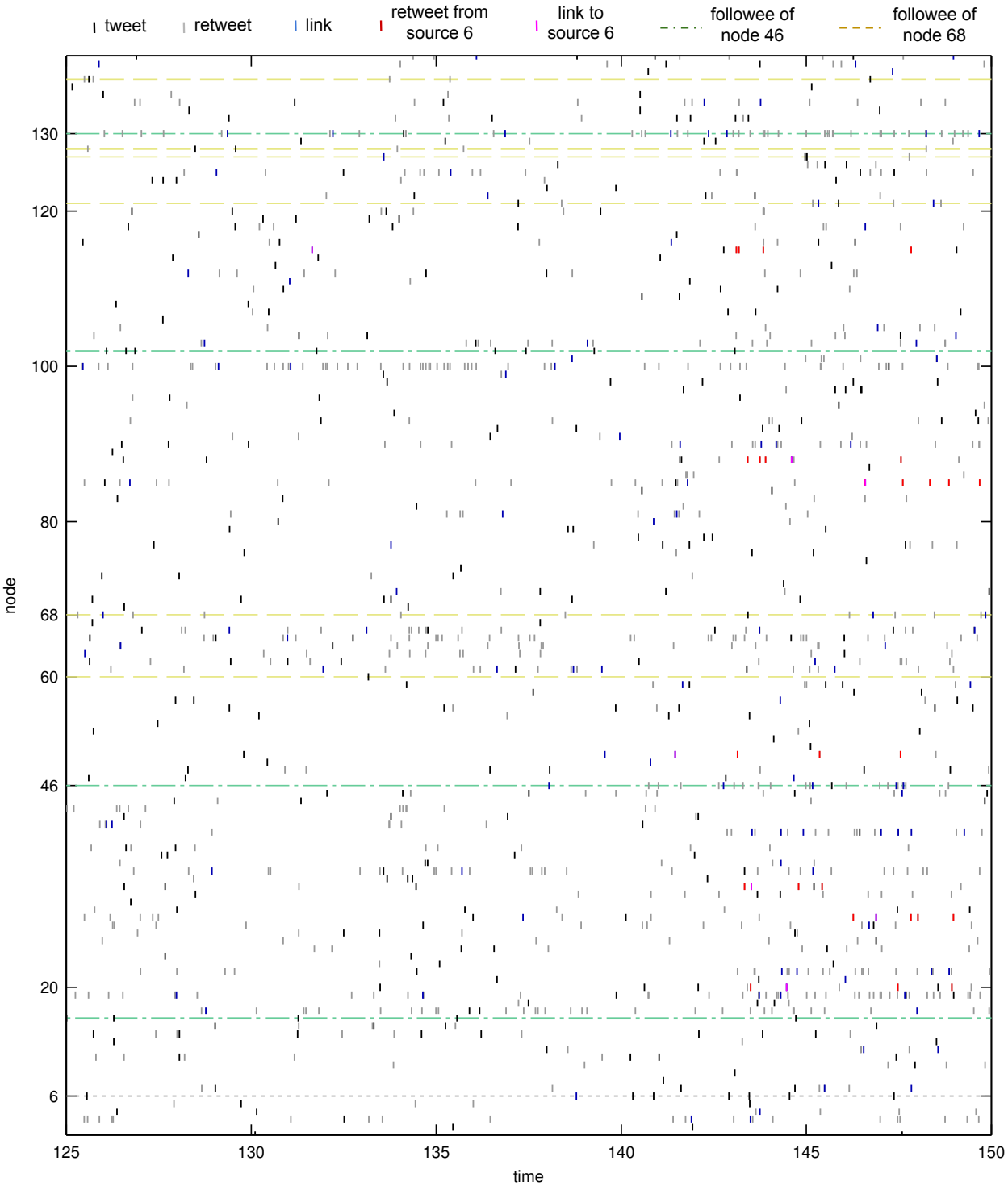


Figure 20: Coevolutionary dynamics of events for the network shown in Figure 21. Explanation is provided in section D.2.

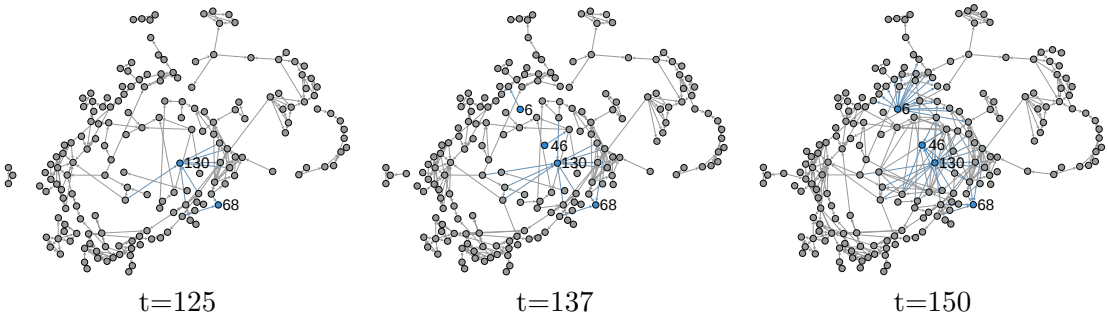


Figure 21: Network structure in which events from Figure 20 take place shown at different time points with the important nodes and links higlighted.

D.2 Cascade Visualization

Figure 20 illustrates the spike trains (tweet, retweet, and link events) for the first 140 nodes of a network simulated with a similar set of parameters as above and Figure 21 shows three snapshots of the network at different times. First, consider node 6 in the network. After she joins the network, a few nodes begin to follow him. Then, when she starts to tweet, her tweets are retweeted many times by others (red spikes) in the figure and these retweets subsequently boost the number of nodes that link to her (Magenta spikes). This clearly illustrates the scenario in which information diffusion triggers changes on the network structure. Second, consider nodes 46 and 68 and compare their associated events over time. After some time, node 46 becomes much more active than node 68. To understand why, note that soon after time 137, node 46 followed node 130, which is a very central node (*i.e.* following a lot of people), while node 68 did not. This clearly illustrates the scenario in which network evolution triggers changes on the dynamics of information diffusion.

Appendix E. Extensions

The basic model presented in Section 3 is just a show-case of the potential of point processes in modeling networks and processes over them. In this section, we extend our model in a variety of ways. More specifically, we explain how the model can be augmented to support link removal, node birth and death, and connection specific parameters. We did not perform experiments with these extensions because our real-world data set does not contain information regarding to link removal and node birth and death. Curating a comprehensive data set that can be used in modeling all these aspects of networks is left as interesting future work.

E.1 Link Deletion

We can generalize our model to support link deletion by introducing an intensity matrix $\boldsymbol{\Xi}^*(t) = (\boldsymbol{\xi}^*_{us}(t))_{u,s\in[m]}$ and model each individual intensity as a survival process. Assume $\boldsymbol{A}^+(t)$ is the previously defined counting matrix $\boldsymbol{A}(t)$, which indicates the existence of an edge at time t. Then, we introduce a new counting matrix $\boldsymbol{A}^-(t) = (A^-_{us}(t))_{u,s\in[m]}$, which

indicates the lack of an edge at time t, and we define it via its intensity function as

$$\mathbb{E}[d\boldsymbol{A}^{-}(t) \,|\, \mathcal{H}^{r}(t) \cup \mathcal{H}^{l}(t)] = \boldsymbol{\Xi}^{*}(t)\, dt, \tag{44}$$

Then, we define the intensity as

$$\boldsymbol{\xi}^{*}_{us}(t) = A^{+}_{us}(t)(\zeta_u + \nu_s \sum_{v \in \mathcal{F}u} \kappa_{\omega_3}(t) \star dA^{-}_{vs}(t)), \tag{45}$$

where the term $A^{+}_{us}(t)$ guarantees that the link has positive intensity to be removed only if it already exists, just like the term $1 - A_{us}(t)$ in Equation (21), the parameter ζ_u is the base rate of link deletion and $\nu_s \sum_{v \in \mathcal{F}u} \kappa_{\omega_3}(t) \star dA^{-}_{vs}(t)$ is the increased link deletion intensity due to increased number of followees of u who decided to unfollow s. This is an excitation term due to deleted links to source s; given s is unfollowed by some followees of u, then u may find s not a good source of information too.

Given a pair of nodes (u, s), the process starts with $A^{+}_{us}(t) = 0$. Whenever a link is created this process ends and a removal process $A^{-}_{us}(t)$ starts. Similarly, when the removal process fires, the connection is removed and a new link creation process is instantiated. These two processes interleave until the end.

E.2 Node Birth and Death

We can augment our model to consider the number of nodes $m(t)$ to change over time:

$$m(t) = m_b(t) - m_d(t) \tag{46}$$

where $m_b(t)$ and $m_d(t)$ are counting processes modeling the numbers of nodes that join and left the network till time t, respectively. The way we construct $m_b(t)$ and $m_d(t)$ guarantees that $m(t)$ is always non-negative.

The birth process, $m_b(t)$, is characterized by a conditional intensity function $\phi^{*}(t)$:

$$\mathbb{E}[dm_b(t) \,|\, \mathcal{H}^{r}(t) \cup \mathcal{H}^{l}(t)] = \phi^{*}(t)\, dt, \tag{47}$$

where

$$\phi^{*}(t) = \epsilon + \theta \sum_{u,s \in [m(t)]} \kappa_{\omega_4}(t) \star dN_{us}(t). \tag{48}$$

Here, ϵ is the constant rate of arrival and $\theta \sum_{u,s \in [m(t)]} \kappa_{\omega_4}(t) \star dN_{us}(t)$ is the increased rate of node arrival due to the increased activity of nodes. Intuitively, the higher the overall activity in the existing network, the larger the number of new users.

The construction of the death process, $m_d(t)$, is more involved. Every time a new user joins the network, we start a survival process that controls whether she leaves the network. Thus, we can stack all these survival processes in a vector, $\boldsymbol{l}(t) = (l_u(t))_{u \in [m]}$, characterized by a multidimensional conditional intensity function $\boldsymbol{\sigma}^{*}(t) = (\sigma_u(t))_{u \in [m_b(t)]}$:

$$\mathbb{E}[d\boldsymbol{l}(t) | \mathcal{H}^{r}(t) \cup \mathcal{H}^{l}(t)] = \boldsymbol{\sigma}^{*}(t)\, dt. \tag{49}$$

Intuitively, we expect the nodes with lower activity to be more likely to leave the network and thus its conditional intensity function to adopt the following form:

$$\sigma^{*}_{u}(t) = (1 - l_u(t)) \left[\sum_{j=1}^{J} \pi_j g_j(t) + \left(h(t) - \sum_{s \in [m(t)]} \kappa_{\omega_5}(t) \star dN_{us}(t) \right)_{+} \right], \tag{50}$$

where the term $(1 - l_u(t))$ ensures that a node is deleted only once, $\sum_{j=1}^{J} \pi_j g_j(t)$ is the history-independent typical rate of death, shared across nodes, which we represent by a grid of known temporal kernels, $\{g_j(t)\}$ with unknown coefficients, $\{\pi_j\}$, and the second term is capturing the effect of activity on the probability of leaving the network. More specifically, if a node is not active, we assume its intensity is upper bounded by $h(t)$ and the most active she becomes, the lower its probability of leaving the network and the larger the term $\sum_{s\in[m(t)]} \kappa_{\omega_5}(t) \star dN_{us}(t)$. The hinge function $(\cdot)_+$ guarantees the intensity is always positive.

Then, given the individual death processes the total death process is

$$m_d(t) = \sum_{u=1}^{m_b(t)} l_u(t), \tag{51}$$

which completes the modeling of the time-varying number of nodes.

E.3 Incorporating Features

One can simply enrich the model by taking into account the longitudinal or static information of the networked data, *e.g.*, by conditioning the intensity on additional external features, such as node attributes or edge types (Li and Zha, 2014; Tran et al., 2015). Let us assume each user u comes with a K-dimensional feature vector $\boldsymbol{x}_u$ including properties such as her age, job, location, number of followers, number of tweets, and more generally structural features (Fadaee et al., 2015). A good feature set can represent the local and global structure of the networks. For example, when the feature set includes number of followers or location information, the link creation process can be reduced to Barabasi-Albert model (Barabási and Albert, 1999), Watts-Strogatz model(Watts and Strogatz, 1998), or their many variations depending on the coefficients.

Interestingly, we can augment the information diffusion intensity as follows. We introduce a K-dimensional link intensity parameter $\boldsymbol{\eta}_u$ in which each dimension reflects the contribution of the corresponding element in the feature vector to the intensity and replace the baseline rate η_u by $\boldsymbol{\eta}_u^\top \boldsymbol{x}_u$. Similarly, we introduce a K-dimensional vector $\boldsymbol{\beta}_s$ where each dimension has a corresponding element in the feature vector $\boldsymbol{x}_s$ and substitute β_s by $\boldsymbol{\beta}_s \boldsymbol{x}_s$. Therefore, one can rewrite the original information diffusion intensity given by Equation (19) as:

$$\gamma^*_{us}(t) = \mathbb{I}[u = s]\, \boldsymbol{\eta}_u^\top \boldsymbol{x}_u + \mathbb{I}[u \neq s]\, \boldsymbol{\beta}_s^\top \boldsymbol{x}_s \sum\nolimits_{v\in\mathcal{F}_u(t)} \kappa_{\omega_1}(t) \star \left(A_{uv}(t)\; dN_{vs}(t)\right). \tag{52}$$

Similarly, we can parameterize the coefficients of the link creation intensity by a K-dimensional vector and write the counter-part of Equation (20) incorporating features of the node for computing the intensity:

$$\lambda^*_{us}(t) = (1 - A_{us}(t))(\boldsymbol{\mu}_u^\top \boldsymbol{x}_u + \boldsymbol{\alpha}_u^\top \boldsymbol{x}_u \sum_{v\in\mathcal{F}_u(t)} \kappa_{\omega_2}(t) \star dN_{vs}(t)). \tag{53}$$

Through this extension we can easily incorporate content quality of events as well; Tweets containing interesting ideas, hot news, or specific language will be retweeted more. Furthermore, these high quality tweets can increase the likelihood of new connections. All these contextual features can be easily incorporated.

Another important feature that can be used is the time. The framework can handle situations that the base rates are time dependent, *i.e.*, $\eta_u(t)$ or $\mu_u(t)$. Please refer to (Tabibian et al., 2017; Du et al., 2015) for good instances of time dependent base rates. For example, where there are seasonal increase or decrease in the follow or retweet processes, we can write the base rate as $\mu_u(t) = \sum_{k=1}^{K} r_k h_k(t)$ where $h_k(t)$'s are sinusoidal kernels with different width and r_k's are the coefficients to be learned.

Surprisingly enough, all the results for convexity for parameter learning, and efficient simulation techniques are still valid for this case too. As far as the features contribute to the intensity linearly, the log-likelihood is concave and we can simulate the model as efficiently as the original model (Li and Zha, 2014; Tran et al., 2015).

E.4 Connection Specific Parameters

Up to this point, the parameters of the link creation and removal, node birth and death and the information diffusion intensities depend on one end point of the interactions. For example β_s and η_u in the information diffusion intensity given by Equation (19) only depend on the source and the actor, respectively. However, proceeding with this example, parameters can be made connection specific, *i.e.*, Equation (19) can be restated as

$$\gamma^*_{us}(t) = \mathbb{I}[u = s]\, \eta_{us} + \mathbb{I}[u \neq s]\, \beta_{us} \sum\nolimits_{v \in \mathcal{F}_u(t)} \kappa_{\omega_1}(t) \star (A_{uv}(t)\ dN_{vs}(t)) \,, \tag{54}$$

where η_{us} is the base intensity of u retweeting a tweet originated by s and β_{us} is the coefficient of excitement of u to retweet s when one of her followees retweets something from s.

Given enough computational resources and large amounts of historical data, one can take into account more complex scenarios and larger and more flexible models. For example, the middle user, say v, who is along the path of diffusion and forwards the tweet originated from s to u can also be taking into consideration, *i.e.*, defining β_{svu} as the amount of increase in intensity of user u retweeting from s when user v has just retweeted a post from s. All desirable properties of simulation algorithm and parameter estimation method still hold.

E.5 Source-oblivious Information Propagation

Our framework, Coevolve, relies on the assumption that the identity of the original source is preserved through the information propagation process. However, some platforms may only keep the last hop in the propagation. Furthermore, this assumption may be violated when users copy the content of the posts rather than reshare or retweet it. The current framework, can be easily extended to handle such cases. Basically, the source-oblivious problem can be posed as a missing (or noisy) data problem which amounts to model and infer the propagation process via incomplete observation and has been studied recently (Zipkin et al., 2016; Du et al., 2015; Stomakhin et al., 2011). Furthermore, the unobserved identity of the source can be modeled as a latent variable and inferred simultaneously with the Hawkes diffusion network (Yang and Zha, 2013).

References

Odd Aalen, Ornulf Borgan, and Hakon Gjessing. *Survival and event history analysis: a process point of view*. Springer, 2008.

Nitin Agarwal, Huan Liu, Lei Tang, and Philip S Yu. Identifying the influential bloggers in a community. In *International Conference on Web Search and Data Mining (WSDM)*, 2008.

Demetris Antoniades and Constantine Dovrolis. Co-evolutionary dynamics in social networks: A case study of twitter. *Computational Social Networks*, 2(1):14, 2015.

Lars Backstrom, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. Four degrees of separation. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 33–42, 2012.

Eytan Bakshy, Jake M. Hofman, Winter A. Mason, and Duncan J. Watts. Everyone's an influencer: Quantifying influence on twitter. In *International Conference on Web Search and Data Mining (WSDM)*, 2011.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

Smriti Bhagat, Amit Goyal, and Laks VS Lakshmanan. Maximizing product adoption in social networks. In *International Conference on Web Search and Data Mining (WSDM)*, 2012.

Prasanta Bhattacharya, Tuan Q Phan, and Edoardo M Airoldi. Analyzing the co-evolution of network structure and content generation in online social networks. *ECIS 2015 Completed Research Papers*, page 18, 2015.

Charles Blundell, Jeff Beck, and Katherine A Heller. Modelling reciprocating relationships with hawkes processes. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, England, 2004.

Ulrik Brandes, Jürgen Lerner, and Tom AB Snijders. Networks evolving step by step: Statistical analysis of dyadic event data. In *IEEE International Conference on Advances in Social Network Analysis and Mining (ASONAM)*, 2009.

Deepayan Chakrabarti, Yiping Zhan, and Christos Faloutsos. R-mat: A recursive model for graph mining. *Computer Science Department*, page 541, 2004.

Justin Cheng, Lada Adamic, P Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. Can cascades be predicted? In *International Conference on World Wide Web (WWW)*, 2014.

Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media, 2007.

Patrick Doreian and Frans Stokman. *Evolution of social networks*. Routledge, 2013.

Nan Du, Le Song, Ming Yuan, and Alex J Smola. Learning networks of heterogeneous influence. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

Nan Du, Le Song, Manuel Gomez-Rodriguez, and Hongyuan Zha. Scalable influence estimation in continuous-time diffusion networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2015.

Paul Erdos and A Rényi. On the evolution of random graphs. *Publications of tke Matkemafical Insfifufe of the Hungarian Academy of Sciences*, 5:17–61, 1960.

Saber Shokat Fadaee, Mehrdad Farajtabar, Ravi Sundaram, Javed A Aslam, and Nikos Passas. On the network you keep: analyzing persons of interest using cliqster. *Social Network Analysis and Mining*, 5(1):1–14, 2015.

Mehrdad Farajtabar, Nan Du, Manuel Gomez-Rodriguez, Isabel Valera, Hongyuan Zha, and Le Song. Shaping social activity by incentivizing users. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

Mehrdad Farajtabar, Manuel Gomez-Rodriguez, Nan Du, Mohammad Zamani, Hongyuan Zha, and Le Song. Back to the past: Source identification in diffusion networks from partially observed cascades. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015a.

Mehrdad Farajtabar, Yichen Wang, Manuel Gomez-Rodriguez, Suang Li, Hongyuan Zha, and Le Song. Coevolve: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems (NIPS)*, 2015b.

Mehrdad Farajtabar, Xiaojing Ye, Sahar Harati, Le Song, and Hongyuan Zha. Multistage campaigning in social networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Mehrdad Farajtabar, Jiachen Yang, Xiaojing Ye, Huan Xu, Rakshit Trivedi, Elias Khalil, Shuang Li, Le Song, and Hongyuan Zha. Fake news mitigation via point process based intervention. *arXiv preprint arXiv:1703.07823*, 2017.

Sharad Goel, Duncan J Watts, and Daniel G Goldstein. The structure of online diffusion networks. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*, 2012.

Anna Goldenberg, Alice X Zheng, Stephen E Fienberg, and Edoardo M Airoldi. A survey of statistical network models. *Foundations and Trends® in Machine Learning*, 2(2): 129–233, 2010.

Manuel Gomez-Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2010.

Manuel Gomez-Rodriguez, David Balduzzi, and Bernhard Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *International Conference on Machine Learning (ICML)*, 2011.

Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. Learning influence probabilities in social networks. In *International Conference on Web Search and Data Mining (WSDM)*, 2010a.

Amit Goyal, Francesco Bonchi, Laks VS Lakshmanan, and Suresh Venkatasubramanian. Approximation analysis of influence spread in social networks. *arXiv preprint arXiv:1008.2005*, 2010b.

Mark Granovetter. The strength of weak ties. *American Journal of Sociology*, pages 1360–1380, 1973.

Thilo Gross and Bernd Blasius. Adaptive coevolutionary networks: a review. *Journal of The Royal Society Interface*, 5(20):259–271, 2008.

Thilo Gross, Carlos J Dommar DLima, and Bernd Blasius. Epidemic dynamics on an adaptive network. *Physical Review Letters*, 96(20):208701, 2006.

Asela Gunawardana, Christopher Meek, and Puyang Xu. A model for temporal dependencies in event streams. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

Fangjian Guo, Charles Blundell, Hanna Wallach, and Katherine Heller. The bayesian echo chamber: modeling social influence via linguistic accommodation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

Eric C Hall and Rebecca M Willett. Tracking dynamic point processes on networks. *IEEE Transactions on Information Theory*, 62(7):4327–4346, 2016.

Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

Petter Holme. Modern temporal network theory: A colloquium. *The European Physical Journal B*, 88(9):1–30, 2015.

Seyed Abbas Hosseini, Keivan Alizadeh, Ali Khodadadi, Ali Arabzadeh, Mehrdad Farajtabar, Hongyuan Zha, and Hamid R Rabiee. Recurrent poisson factorization for temporal recommendation. *arXiv preprint arXiv:1703.01442*, 2017.

David Hunter, Padhraic Smyth, Duy Q Vu, and Arthur U Asuncion. Dynamic egocentric models for citation networks. In *International Conference on Machine Learning (ICML)*, 2011.

David R Hunter and Kenneth Lange. A tutorial on mm algorithms. *The American Statistician*, 58(1):30–37, 2004.

Tomoharu Iwata, Amar Shah, and Zoubin Ghahramani. Discovering latent influence in online social activities via shared cascade poisson processes. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2013.

Mohammad Reza Karimi, Erfan Tavakoli, Mehrdad Farajtabar, Le Song, and Manuel Gomez-Rodriguez. Smart broadcasting: Do you want to be seen? In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2016.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2003.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is Twitter, a social network or a news media? In *International Conference on World Wide Web (WWW)*, 2010.

Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2005.

Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2008.

Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, 11(Feb):985–1042, 2010.

Liangda Li and Hongyuan Zha. Learning parametric models for social infectivity in multi-dimensional hawkes processes. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

Shuang Li, Yao Xie, Mehrdad Farajtabar, Apurv Verma, and Le Song. Detecting weak changes in dynamic events over networks. *IEEE Transactions on Signal and Information Processing over Networks*, 2017.

Wenzhao Lian, Vinayak Rao, Brian Eriksson, and Lawrence Carin. Modeling correlated arrival events with latent semi-markov processes. In *International Conference on Machine Learning (ICML)*, 2014.

Scott W Linderman and Ryan P Adams. Discovering latent network structure in point process data. In *International Conference on Machine Learning (ICML)*, 2014.

Seth A Myers and Jure Leskovec. The bursty dynamics of the twitter information network. In *International Conference on World Wide Web (WWW)*, 2014.

Yosihiko Ogata. On lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31, 1981.

Ankur P Parikh, Asela Gunawardana, and Christopher Meek. Conjoint modeling of temporal dependencies in event streams. In *UAI Bayesian Modelling Applications Workshop*. Citeseer, 2012.

Tuan Q Phan and Edoardo M Airoldi. A natural experiment of social network formation and dynamics. *Proceedings of the National Academy of Sciences*, 112(21):6595–6600, 2015.

Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2002.

Manuel Gomez Rodriguez and Bernhard Schölkopf. Influence maximization in continuous time diffusion networks. In *International Conference on Machine Learning (ICML)*, 2012.

Manuel Gomez Rodriguez, Jure Leskovec, and Bernhard Schölkopf. Modeling information propagation with survival theory. In *International Conference on Machine Learning (ICML)*, 2013.

Daniel Mauricio Romero and Jon Kleinberg. The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter. In *International Conference on Web and Social Media (ICWSM)*, 2010.

Sheldon M. Ross. *Introduction to Probability Models, Tenth Edition*. Academic Press, Inc., 2011. ISBN 978-0-12-375686-2.

Philipp Singer, Claudia Wagner, and Markus Strohmaier. Factors influencing the co-evolution of social and content networks in online social media. In *Modeling and Mining Ubiquitous Social Media*, pages 40–59. Springer, 2012.

Tom AB Snijders. Siena: Statistical modeling of longitudinal network data. In *Encyclopedia of Social Network Analysis and Mining*, pages 1718–1725. Springer, 2014.

Tom AB Snijders and SR Luchini. Statistical methods for network dynamics. In *Proceedings of the XLIII Scientific Meeting, Italian Statistical Society*. CLEUP, 2006.

Alexey Stomakhin, Martin B Short, and Andrea L Bertozzi. Reconstruction of missing data in social networks based on temporal patterns of interactions. *Inverse Problems*, 27(11): 115013, 2011.

Behzad Tabibian, Isabel Valera, Mehrdad Farajtabar, Le Song, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. Distilling information reliability and source trustworthiness from digital traces. In *International Conference on World Wide Web (WWW)*, 2017.

Long Tran, Mehrdad Farajtabar, Le Song, and Hongyuan Zha. Netcodec: Community detection from individual activities. In *SIAM International Conference on Data Mining (SDM)*, 2015.

Johan Ugander, Lars Backstrom, and Jon Kleinberg. Subgraph frequencies: Mapping the empirical and extremal geography of large graph collections. In *International conference on World Wide Web (WWW)*, 2013.

Isabel Valera and Manuel Gomez-Rodriguez. Modeling adoption of competing products and conventions in social media. In *IEEE International Conference on Data Mining (ICDM)*, 2015.

Duy Q Vu, David Hunter, Padhraic Smyth, and Arthur U Asuncion. Continuous-time regression models for longitudinal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

Eric Wang, Jorge Silva, Rebecca Willett, and Lawrence Carin. Time-evolving modeling of social networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

Duncan J Watts and Steven H Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, June 1998.

Lilian Weng, Jacob Ratkiewicz, Nicola Perra, Bruno Gonçalves, Carlos Castillo, Francesco Bonchi, Rossano Schifanella, Filippo Menczer, and Alessandro Flammini. The role of information diffusion in the evolution of social networks. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2013.

Shuai Xiao, Junchi Yan, Mehrdad Farajtabar, Le Song, Xiaoang Yang, and Hongyuan Zha. Joint modeling of event sequence and time series with attentional twin recurrent neural networks. *arXiv preprint arXiv:1703.08524*, 2017.

Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *International Conference on Machine Learning (ICML)*, 2016.

Shuang-Hong Yang and Hongyuan Zha. Mixture of mutually exciting processes for viral diffusion. In *International Conference on Machine Learning (ICML)*, 2013.

Ali Zarezade, Ali Khodadadi, Mehrdad Farajtabar, Hamid R Rabiee, and Hongyuan Zha. Correlated cascades: Compete or cooperate. *arXiv preprint arXiv:1510.00936*, 2015.

Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2013a.

Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional hawkes processes. In *International Conference on Machine Learning (ICML)*, 2013b.

Joseph R Zipkin, Frederic P Schoenberg, Kathryn Coronges, and Andrea L Bertozzi. Point-process models of social network interactions: parameter estimation and missing data recovery. *European Journal of Applied Mathematics*, 27(03):502–529, 2016.

Gerd Zschaler, Gesa A Böhme, Michael Seißinger, Cristián Huepe, and Thilo Gross. Early fragmentation in the adaptive voter model on directed networks. *Physical Review E*, 85 (4):046107, 2012.