# POINT PROCESS MODELING AND OPTIMIZATION OF SOCIAL NETWORKS

A Dissertation
Presented to
The Academic Faculty

By

Mehrdad Farajtabar

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computational Science and Engineering

Georgia Institute of Technology

May 2018

Copyright © Mehrdad Farajtabar 2018

**POINT PROCESS MODELING AND OPTIMIZATION OF SOCIAL NETWORKS**

Approved by:

Dr. Hongyuan Zha, Advisor
School of Computational Science and Engineering
*Georgia Institute of Technology*

Dr. Le Song, Co-advisor
School of Computational Science and Engineering
*Georgia Institute of Technology*

Dr. Mark Davenport
School of Computer and Electrical Engineering
*Georgia Institute of Technology*

Dr. Xiaojing Ye
Department of Mathematics and Statistics
*Georgia State University*

Dr. Bistra Dilkina
Department of Computer Science
*University of Southern California*

Date Approved: March 27, 2018

*Soul receives from soul the knowledge thereof, not by way of book nor from tongue.*

*Rumi*

*To my*

*beloved mother,*

*dear father, and*

*lovely spouse.*

## ACKNOWLEDGEMENTS

I would like to express my deepest gratitude and appreciation to Professor Hongyuan Zha for his visionary, inspiration, and guidance which truly set the course to what is here today. I was always inspired by his in-depth knowledge in mathematics of machine learning, cleverness, and insight from which I learned a lot. My path through PhD would not be productive without my co-advisor Professor Le Song. We sat hours and hour together tackling research questions. His enthusiasm and vast knowledge of machine learning plus his attitude to perform state-of-the-art research were always an inspiration for me.

Additionally, I would like to thank the other members of my Ph.D. dissertation committee: Dr. Mark Davenport, Dr. Bistra Dilkina, Dr. Xiaojing Ye, for their time to attend my thesis and proposal defense and their insightful suggestions and comments on my research.

My research achievements also benefit significantly from interacting with a marvelous group of colleagues and collaborators. I would like to extend my sincere appreciation to each of them: Nan Du, Rakshit Trivedi, Jiachen Yang, Abbas Hosseini, Ali Zarezadeh, Ali Khodadadi, Isabel Valera, Behzad Tabibia, Hongteng Xu, Elias Khalil, Hamidreza Rabiee, Yichen Wang, Mohammadreza Karimi, Erfan Tavakoli, Sahar Harati, Bo Xie, Mohammad Zamani, Amrita Gupta, Amr Ahmed, Alex Smola, Bernard Schlkopf, Mohammad Ghavamzadeh, Yinlam Chow, Ryen White, Emre Kiciman, Girish Nathan, Shuang Li, Long Tran, Yao Xie, Apurv Verma, Huan Xu, Shuai Xiao, Byron Boots, Sauber Shokat-Fadaee, and Junchi Yan. In particular, I'm grateful to Dr. Manuel Gomez-Rodriguez for his collaboration and support during early stages of my PhD. Manuel is one of the most principled and knowledgable researchers in the area of networks I know.

I feel deeply indebted to the selfless love and endless support of my parents. I cannot express in words my gratitude to them for their sacrifices for me. Their love is uncountable. I'm also grateful to my brothers, Mohammad and Ali, who always trust me. I would also like to express respect and gracious to my extended family members, grandparents,

uncles and aunts, cousins, and my in laws for their love, trust, and support. Finally and importantly, I'm always thankful to my wife, Sahar, for having this long journey with me, for the cheerful moments she made for us in these years, for never giving up and always encouraging me, for believing in me unconditionally, and for tolerating an always-busy, always-stressed, always-in-a-deadline husband. Thank you Sahar!

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## SUMMARY

Online social media such as Facebook and Twitter and communities such as Wikipedia and Stackoverflow turn to become an inseparable part of today's lifestyle. Users usually participate via a variety of ways like sharing text and photos, asking questions, finding friends, and favoring contents. Theses activities produce sequences of events data whose complex temporal dynamics need to be studied and is of many practical, economic, and societal interest. We propose a novel framework based on multivariate temporal point processes that is used for *modeling*, *optimization*, and *inference* of processes taken place over networks.

In the modeling part, we propose a temporal point process model for joint dynamics of information propagation and structure evolution in networks. These two highly intertwined stochastic processes have been predominantly studied separately, ignoring their co-evolutionary dynamics. Our model allows us to efficiently simulate interleaved diffusion and network events, and generate traces obeying common diffusion and network patterns observed in real-world networks. In the optimization part, we establish the fundamentals of intervention and control in networks by combining the rich area of temporal point processes and the well-developed framework of Markov decision processes. We use point processes to capture both endogenous and exogenous events in social networks and formulate the problem as a Markov decision problem. Our methodology helps finding the optimal policy that balances the high present reward and large penalty on low future outcome in the presence of extensive uncertainties. In the inference part, we propose an intensity-free approach for point processes modeling that transforms nuisance process to the target one. Furthermore, we train our deep neural network model using a likelihood-free approach leveraging Wasserstein distance between point processes.

# CHAPTER 1
# INTRODUCTION AND BACKGROUND

## 1.1 Motivation

Event sequences are becoming increasingly available in a variety of applications and domains. They are asynchronously generated with random timestamps and are ubiquitous in areas such as e-commerce, social networks, electronic health data, equipment failures, epidemiology, and economics.

In epidemiology, we collect observations about the presence of animals, such as birds, migrating across a wide range of geographic locations in certain time periods. In invasive species management the spread of non-native species to new areas is a cause of major concern, because they harm native species through predation, competition, disease or by otherwise disrupting food webs and ecosystem processes. The ability to model the dynamics of diffusion processes enables the development of strategies for steering them towards desirable outcomes.

Electronic health records contain the information of diagnoses, treatments, and clinical interventions. The progress of a disease and its influence on other diseases can be captured via the historical admissions of different patients and events happened. Furthermore, a disease network can be constructed accordingly that captures the correlation between different diseases.

In malware detection systems, volunteer machines report potentials malicious files. Although the attacks are reported from different machines at different times, they may share some common dynamics, for example, they can be clustered and coordinated or seasonal. Study of these attack events will provide a glimpse into the spread of cyber threats across the Internet.

Social and information networks and virtual communities play a key role in everyday's life. People and entities share opinions, beliefs, and news updates in social medias and engage in social interactions by commenting, liking, mentioning and following each other. This virtual world is an ideal place for studying social behaviors and spread of cultural norms, advertising and marketing, and estimating the culprit in malicious diffusions. Among them, the study of information diffusion or more generally dynamics on the network is of crucial importance and is of many practical, economic, and societal interest.

In economics, the consideration of the peculiar properties of financial transaction data, such as the irregular spacing in time, the bid-ask bounce, and the existence of serial dependence between events and markets call for developing new econometric approaches. It has been realized that the timing of trading events, such as the arrival of particular orders and trades, and the frequency in which the latter occur have information value for the state of the market and play an important role in subsequent analysis.

In this thesis, we study *networks and processes* over them. As exemplified above, a variety of real-world processes produce sequences of data whose complex temporal dynamics need to be studied. A common property of the above problems is that, the event timestamps can carry important information about the underlying dynamics, which otherwise are not available from the time-series evenly sampled from continuous signals. A major line of research has attempted to study event sequence, especially exploring the timestamp data to model the underlying and latent dynamics of the system, whereby *temporal point process* has been a powerful and elegant framework this direction. Furthermore, the interdependence between phenomena and temporal events provokes the surge of *networked* or multi-dimensional modeling approaches. The developed framework leverages tools and concepts from many related fields like machine learning, data mining, statistics, etc. Figure 1.1 shows the related fields of application and tools from computer science and mathematics.

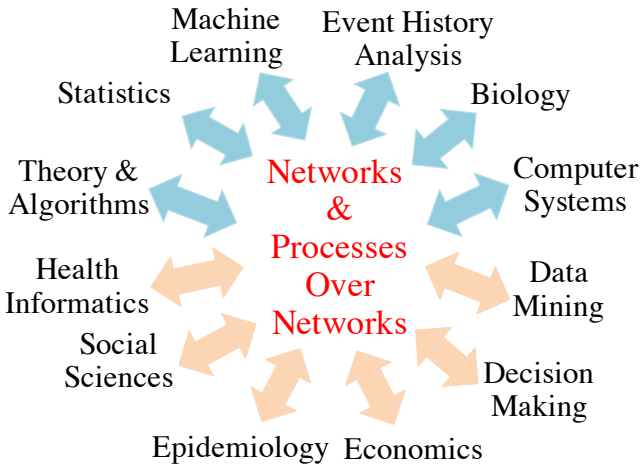



Figure 1.1: Tools and connections

## 1.2 Preliminaries

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

$$dN(t) = \sum_{t_i \in \mathcal{H}(t)} \delta(t - t_i)\, dt, \tag{1.1}$$

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

$$f(t) \star dN(t) := \int_0^t f(t - \tau)\, dN(\tau) = \sum\nolimits_{t_i \in \mathcal{H}(t)} f(t - t_i). \tag{1.2}$$

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 [1, 73].

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

$$\lambda^*(t)dt := \mathbb{P}\left\{\text{event in } [t, t + dt)|\mathcal{H}(t)\right\} = \mathbb{E}[dN(t)|\mathcal{H}(t)], \tag{1.3}$$

where $^*$ means that the function $\lambda^*(t)$ may depend on the history $\mathcal{H}(t)$. 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 [159]:

$$S^*(t') = \exp\left(-\int_t^{t'} \lambda^*(\tau)\, d\tau\right), \tag{1.4}$$

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

$$f^*(t') = \lambda^*(t')\, S^*(t'), \tag{1.5}$$

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

$$F^*(t') = 1 - S^*(t') = \int_t^{t'} f^*(\tau)\, d\tau. \tag{1.6}$$

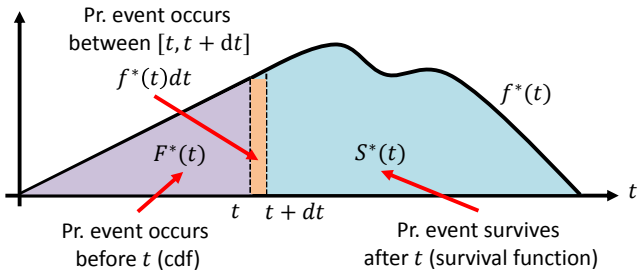



Figure 1.2: Illustration of three inter-related quantities in point processes framework: conditional density function, conditional cumulative density function, and survival function.

Figure 1.2 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

$$\mathfrak{L} = \sum_{i=1}^{n} \log \lambda^*(t_i) - \int_0^T \lambda^*(\tau)\, d\tau, \quad T \geqslant t_n. \tag{1.7}$$

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 [1]:

(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.*,

$$\lambda^*(t) = g(t) \geqslant 0. \tag{1.8}$$

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

$$\lambda^*(t) = \mu + \alpha \kappa_\omega(t) \star dN(t) = \mu + \alpha \sum\nolimits_{t_i \in \mathcal{H}(t)} \kappa_\omega(t - t_i), \tag{1.9}$$

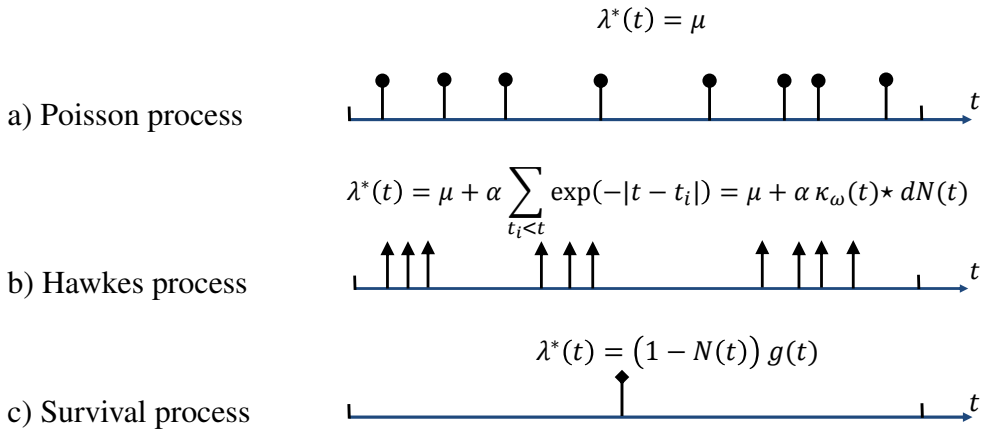



Figure 1.3: Three types of point processes with a typical realization

where,

$$\kappa_\omega(t) := \exp(-\omega t)\mathbb{I}[t \geqslant 0] \tag{1.10}$$

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.*,

$$\lambda^*(t) = (1 - N(t))g(t), \tag{1.11}$$

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 1.3 illustrates these processes. Interested reader should refer to [1] for more details

on the framework of temporal point processes.

Similarly, we can define a multi-variate point process as:

$$N(t) = (N_1(t), N_2(t), \ldots, N_U(t))^\top$$

where $N_i(t)$ is the counting process for the $i$-th dimension. Corresponding to each dimension there is an intensity function $\lambda_i^*(t)$ where in vector format it's represented as:

$$\lambda^*(t) = (\lambda_1^*(t), \lambda_2^*(t), \ldots, \lambda_U^*(t))^\top$$

The dimensions can be correlated or independent. One way to define a multivariate point process is via Hawkes process. An $U$-dimensional Hawkes process with the conditional intensity for the $i$-th dimension is:

$$\lambda_i^*(t) = \mu_i(t) + \sum_{j=1}^{U} \int_0^t \phi_{ij}(t-s)\, dN_j(s),$$

$\phi_{ij}(t)$ - the time-decaying kernel, captures the *mutually exciting* property: the effect of the occurrence of events in dimension $j$ on the likelihood of future events in dimension $i$.

To make the notation more clear we may occasionally remove $*$ from the conditional intensity function.

### 1.3 Simulation

In this section, we introduce Ogata's algorithm which is standard way to simulate multivariate point and Hawkes Processes. 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 $t$.

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.

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 [167], *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

$$t' = \underbrace{t}_{\text{current time}} + \underbrace{s.}_{\text{waiting time for the first event}} \tag{1.12}$$

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 $\hat{\lambda} = \max_{t \leq \tau \leq T} \lambda^*_{sum}(\tau)$, $t'$ 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 $1 \leq u \leq U$. This procedure is iterated until we reach the end of simulation time $T$. Algorithm 1 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 1** Ogata's Algorithm

**Input:** $U$ dimensional Hawkes process $\left\{\lambda_u^*(t)\right\}_{u=1\ldots U}$, Due time: $T$
**Output:** Set of events: $\mathcal{H} = \left\{(t_1, u_1), \ldots, (t_n, u_n)\right\}$
$t \leftarrow 0$
$i \leftarrow 0$ $t < T$
$\lambda_{sum}^*(\tau) \leftarrow \sum_{u=1}^{U} \lambda_u^*(\tau)$
$\hat{\lambda} \leftarrow \max_{t \le \tau \le T} \lambda_{sum}^*(\tau)$
$s \sim Exponential(\hat{\lambda})$
$t' \leftarrow t + s$ $t' \geq T$
break
} Sampling next event time
$\bar{\lambda} \leftarrow \lambda_{sum}^*(t')$
$d \sim Uniform(0,1)$ $d \times \hat{\lambda} > \bar{\lambda}$
$t \leftarrow t'$
Goto 5
} Rejection test
$S \leftarrow 0$
$d \sim Uniform(0,1)$ $u \leftarrow 1$ to $U$
$S \leftarrow S + \lambda_u^*(t')$ $S \geq d$
$i \leftarrow i + 1$
$u_i \leftarrow u$
$t_i \leftarrow t'$
$t \leftarrow t'$
Goto 5
} Attribution test
Given the new event just sampled update intensity functions $\lambda_u^*(\tau)$

### 1.4 EM-type Parameter Learning Algorithm

Since the log-likelihood is concave one can simply take any convex optimization method to find the parameters. However, these methods usually require hyper parameters like convergence rate or their performance is dependent to the initial point. Alternatively, Expectation Maximization algorithms could be useful in the case that they are parameter free and are not that sensitive to initialization. However, they are prone to local minima.

In our experiments, we adopt an efficient algorithm inspired by the previous work [55, 229, 228]. Our algorithm is an EM-type algorithm that enjoys parameter-less property of EM and at the same time is guaranteed to find the global optimum. The structure of our problem allows us to develop such algorithms.

For brevity let's assume we have $U$-dimensional Hawkes process represented by its

conditional intensity function. For simplicity we develop an algorithm for learning the parameters of the most basic form of Hawkes Process. It will easily be adapted to other forms of point processes we used in this thesis.

Starting from Equation (1.9) the intensity for the $u$-th dimension is

$$\lambda_u^*(t) = \mu_u + \sum_{t_i<t} a_{uu_i} g(t - t_i) \tag{1.13}$$

where $\boldsymbol{A} = (a_{uv})_{u,v=1\ldots U}$ and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_U)^\top$ are endogenous and exogenous intensity parameters, respectively [55] and $g()$ is the decaying kernel.

Having observed a cascade of events we can learn the influence network $\boldsymbol{A} = (a_{uv})_{u,v=1\ldots U}$ and the exogenous intensity $\boldsymbol{\lambda}^0 = (\lambda_1, \ldots, \lambda_U)^\top$. Assume $n$ events $(t_i, u_i)_{i=1\ldots n}$ are observed in the interval $[0, T]$.

The log-likelihood is

$$\begin{aligned}\mathcal{L}(\boldsymbol{A}, \boldsymbol{\mu}) = &\sum_{i=1}^{n} \log\big(\mu_{u_i}(t_i) + \sum_{j=1}^{i-1} a_{u_i u_j} g(t_i - t_j)\big) \\ &- \sum_{u=1}^{U} T\mu_u - \sum_{u=1}^{U}\sum_{j=1}^{n} a_{uu_j} G(T - t_j)\end{aligned} \tag{1.14}$$

where $G(t) = \int_0^t g(\tau)\, d\tau$. The logarithm of the same is the term causing problem. We can use Jensen inequality to break lower bound the log-sum. We define auxiliary variables $\nu_{ij}$ for all pairs of events ($1 \leq j \leq i \leq n$) such that

$$\begin{aligned}&\sum_{j=1}^{i} \nu_{ij} = 1 \qquad \forall\, i : 1 \leq i \leq n \\ &\nu_{ij} \geq 0 \qquad \forall\, i : 1 \leq j \leq i \leq n\end{aligned} \tag{1.15}$$

Then, according to Jensen inequality we have

$$\mathcal{L}(\boldsymbol{A}, \boldsymbol{\mu}) \geq \sum_{i=1}^{n} \Big( \nu_{ii} \log \left( \mu_{u_i}(t_i) \right) + \sum_{j=1}^{i-1} \nu_{ij} \log \left( a_{u_i u_j} g(t_i - t_j) \right) - \sum_{j=1}^{i} \nu_{ij} \log(\nu_{ij}) \Big) \\ - \sum_{u=1}^{U} T \mu_u - \sum_{u=1}^{U} \sum_{j=1}^{n} a_{u u_j} G(T - t_j) \triangleq \mathcal{L}'(\boldsymbol{A}, \boldsymbol{\mu}, \left\{ \nu_{ij} \right\}) \tag{1.16}$$

Now, by taking gradient of the lower-bound with respect to parameters we can find the close form updates as follows:

$$\frac{\partial \mathcal{L}}{\partial \mu_u} = \sum_{i:u_i=u}^{n} \frac{\nu_{ii}}{\mu_{u_i}} - T \Longrightarrow \mu_u = \frac{\sum_{i:u_i=u}^{n} \nu_{ii}}{T} \tag{1.17}$$

$$\frac{\partial \mathcal{L}}{\partial a_{uv}} = \sum_{i:u_i=u} \sum_{j<i:u_j=v} \nu_{ij} \log(a_{uv}) - \sum_{j:u_j=v} a_{uv} G(T - t_j) \Longrightarrow a_{uv} = \frac{\sum_{i:u_i=u} \sum_{j<i:u_j=v} \nu_{ij}}{\sum_{j:u_j=v} G(T - t_j)} \tag{1.18}$$

The lower bound is valid for every choice of $\left\{ \nu_{ij} \right\}$ which satisfies constraints in (1.15). However, by maximizing the lower bound with respect to auxiliary variables we can make sure that the lower bound sticks to the actual value [218, 191].

$$\begin{aligned} &\text{maximize}_{\left\{ \nu_{ij} \right\}} && \mathcal{L}'(\boldsymbol{A}, \boldsymbol{\mu}, \left\{ \nu_{ij} \right\}) \\ &\text{subject to} && \textstyle\sum_{j=1}^{i} \nu_{ij} = 1 \qquad \forall i : 1 \leq i \leq n \\ & && \nu_{ij} \geq 0 \qquad \forall i : 1 \leq j \leq i \leq n \end{aligned} \tag{1.19}$$

The above constrained optimization problem can be solved easily via Lagrange multipliers which will result in:

$$\eta_{ii} = \frac{\mu_{u_i}}{\mu_{u_i} + \sum_{j=1}^{i-1} a_{u_i u_j} g(t_i - t_j)} \qquad 1 \leq i \leq n \tag{1.20}$$

$$\eta_{ij} = \frac{a_{u_i u_j} g(t_i - t_j)}{\mu_{u_i} + \sum_{j=1}^{i-1} a_{u_i u_j} g(t_i - t_j)} \qquad 1 \leq j < i \leq n \tag{1.21}$$

In contrast to the above algebraic argument [126] propose a probabilistic point of view to parameter learning problem by introducing latent variables indicating the branching structure of events.

## 1.5 Related Works

In this section, we briefly overview the related works in the domain of networks and processes over them, and point processes.

### 1.5.1 Temporal Networks

Much effort has been devoted to modeling the evolution of social networks [153, 45, 202, 146, 15]. Of the proposed methods in characterizing link creation, triadic closure [79] 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 [68]. 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. [96] used rule-based methods to model the evolution of the graph over time. [66] analyzed community structure over time and [121] studied the interaction of the friendship graph among group members and group growth. Recently, [181] used a Cox-intensity Poisson model with exponential random graphs to model friendship dynamics. [27] extended this model to the temporal sequence of interactions that take place in the social network, but with insufficient model flexibility, and limited scalability. Modeling temporal dynamics of interactions in this way provides new opportunities for identifying network topology at multiple scales [65] and for early detection of popular resources [98, 120]. 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 [99], with a short section devoted to point process based approaches.

### 1.5.2 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 [3, 83, 184, 86], micro blogs [207, 14] 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 [170, 77] and used for decision making, such as identifying influencer [3], maximizing information spread [111, 165], and marketing planing [161, 44, 17, 19]. 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 [51, 70, 218], predicting the timings of future events [50, 164], finding source of information cascades [56]. Besides this, Point process modeling of activity in network is becoming increasingly popular [130, 150, 89]. 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 [78, 111], sampling methods [129, 85, 132] for inference and prediction, and convex framework proposed in [55] to make decisions to shape the activity to a variety of objectives. 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.

### 1.5.3 Point Processes

Multivariate point processes are mathematical frameworks for modeling multidimensional event data [1]. However, only very recently machine learning community are starting to use it for modeling practical problems. For instance, these mathematical tools has been applied to analyze human actions in emergency situations [28], to model citations between scholarly works [200], and information diffusion process in social networks [144, 72, 70] and complex networks [142]. Point processes, in particular, have been shown to provide effective tools for in a number of applications [40, 178, 158, 2, 200]. However, the limitation of these previous works is that they do not model the interdependency between different users and can only be applied to relatively small dimensions. In the case of high dimensional multivariate point processes, the dependency structures parameters between the dimensions are often unknown. It is an interesting and challenging question whether we can uncover these dependency structures based on the time stamps and features of the events. This problem has been addressed only by a paucity of recent studies in the literature [144, 72, 70]. Furthermore, there is a whole range of other unsupervised learning and inference tasks which have not been addressed by previous work. In our proposed framework, we propose to solve the problem by explicitly modeling the interaction between dimensions and investigate unsupervised learning and novel inference tasks for large-scale datasets.

Point processes [40] are proposed to model event sequences in the continuous observation domain. Typical temporal point processes like self exciting processes [94, 34] and Self-correcting processes [105, 38] have been used for financial analysis [11], social network modeling [228] and bioinformatics [160], knowledge graph dynamics [192]. Spatial-temporal point processes can be viewed as an extension of temporal point processes from time domain to spatial-temporal domain. Classical applications of them include seismic

analysis [149], transportation analysis [52], and invasive species management [88].

Most of existing work uses parametric models to describe the dynamics of observed events [55, 225]. To enhance the flexibility, nonparametric models and learning algorithms of point processes have been explored from different viewpoints, e.g., the method based on ordinary differential equation [160, 90, 126], the methods based on basis representation [119, 216], and those based on Bayesian nonparametrics [134, 130, 172], and online nonparametric algorithms [219]

One way to model the temporal dependency between event data, such as self- or mutual-excitation effects, can be carried out via Hawkes processes [93]. This continuous-time point processes has been applied to a wide range of applications such as market modeling [190], earth quake prediction [135], crime modeling [183], wireless network analysis [141]. The maximum likelihood estimation of one-dimensional Hawkes process is studied in [126] under the EM framework. Additionally, [178] models cascades of events using marked Poisson processes with marks representing the types of events while [22] propose a model based on Hawkes process that models events between pairs of nodes. However, most existing work focuses on model a small number of dimensions, our proposed work try to develop a more general framework which will address a broader range of problems arising from asynchronously generated interdependent event data. For learning point processes one can refer to [215, 214, 229, 228]. Point process models have been applied to many tasks in networks such as fake news mitigation [59], recommendation systems [100], outlier detection [128], disease propagation [112], activity shaping in social networks [55], verifying crowd-generated data [186], sequence modeling using deep recurrent neural networks [212], campaigning in networks [60], preventive maintenance [29].

## 1.6 Contributions and Organization of the Thesis

This thesis is a collection of work to explore and develop a set of robust machine learning methods for modeling and optimizing large real-world networks and processes that take

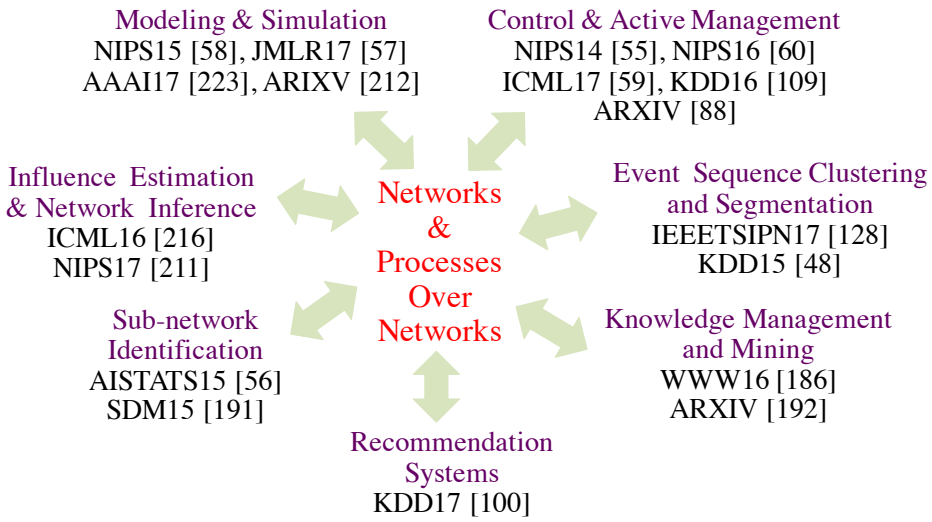



Figure 1.4: Sub-fields of the point processes framework in social networks and the contributions of the author.

place over them. We propose a novel framework based on multivariate temporal point processes (e.g. Poisson Process and Hawkes Process) and survival analysis which explicitly model the rate of event occurrence as a function of timing and features of previous events for temporal networks. Furthermore, our modeling framework can explicitly take into account latent variables, low intrinsic dimensionality, and sparsity of the datasets (Figure 1.1).

Modeling networks and processes over them consists of the broad span of sub-problems which can be roughly categorized into several classes: (I) Models and algorithms of social networks and information diffusion and their simulation; (II) Parameter learning and network inference; (III) Event stream clustering and segmentation; (IV) Sub-network identification from event dynamics; (V) Recommendation; (VI) Influence estimation and network inference; and (VII) Control and management for temporal networks for more desirable network outcomes. Refer to Figure 1.4 for the resultant publication on this thesis by the author. Our research covers this span and will make a coherent framework which can simultaneously model the timing and event features. It can make predictions in a setting where conventional machine learning techniques are incapable of, such as the "*who* will do *what* and *when*" question critical to event stream modeling. In contrast to traditional

event history analysis methods based on static networks, our coevolution models address the complex dependent aspects of such event datasets in the context of novel challenging problems arising from real-world applications. Furthermore, our research topics are driven by real-world event stream modeling problems in social networks, health informatics and micro-finance.

The proposed research on social media, health informatics, cyber-security and P2P finance will have strong societal and environmental impacts. As a broader impact, our framework systematically bring together several research areas, such as point processes, theory and algorithms, statistics, and sparsity recovery theory, in a common platform to study social networks, social media, micro-finance, and health informatics and EHR-driven phenotyping (Figure 1.1). In social media, It brings practical values to Internet industry by better understanding and modeling of user behaviors.

Our first contribution is a joint model of co-evolution dynamics of information diffusion and network structure. Traditionally, 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. In chapter 2, we propose a temporal point process model, 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 traces obeying common diffusion and network patterns observed in real-world networks. Furthermore, we also develop a convex optimization framework to learn the parameters of the model from historical diffusion and network evolution traces. We experiment 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.

In chapters 3, 4, and 5 we, step by step, build our *point process intervention* framework. Given that we can learn accurately the model in temporal networks, we now consider the possibility of actively manage the network structure to achieve more desirable network outcomes. For example, in the era of precision medicine, can we alter the disease network with new medical treatment in order to achieve more favorable health outcomes preferred by individual patients? Can we model and exploit social network data to steer the online community to a desired activity level? Specifically, can we drive the overall usage of a service to a certain level (e.g., at least twice per day per user) by incentivizing a small number of users to take more initiatives? What about maximizing the overall service usage for a target group of users? Furthermore, these activity shaping problems need to be addressed by taking into account budget constraints, since incentives are usually provided in the form of monetary or credit rewards. In this thesis, we model social events using multivariate Hawkes processes, which can capture both endogenous and exogenous event intensities, and derive a time dependent linear relation between the intensity of exogenous events and the overall network activity. We develop a convex framework for determining the required level of external drive in order for the network to reach a desired activity level. We experiment with large event data gathered from Twitter, and show that our method can steer the activity of the network more accurately than alternatives. In chapter 3 we formulate the problem as a single stage optimization algorithm where all the intervention actions are decided at the beginning of the process. In chapter 4, we allow the intervention to be done at several stages where the associated Markov Decision problem is solved via approximate dynamic programming. Here, we evaluate the effectiveness of the proposed *point process intervention* procedure on social campaigning task. Next, in chapter 5 we formulate a reinforcement learning solution for a process who is going to mitigate the effects of a fake news propagation processes.

Point processes are often characterized via intensity function which limits model's expressiveness due to unrealistic assumptions on its parametric form used in practice. Fur-

thermore, they are learned via a maximum likelihood approach which is prone to failure in multi-modal distributions of sequences. In chapter 6, we propose an intensity-free approach for point processes modeling that transforms nuisance processes to a target one. Furthermore, we train the model using a likelihood-free leveraging Wasserstein distance between point processes. Experiments on various synthetic and real-world data substantiate the superiority of the proposed point process model over conventional ones.

Finally, the thesis is concluded in chapter 7 and a few future directions are discussed.

# CHAPTER 2
# POINT PROCESS MODELING OF INFORMATION DIFFUSION AND NETWORK EVOLUTION

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.

In this chapter, 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 traces obeying common diffusion and network patterns observed in real-world networks. 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.

## 2.1 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 [116]. 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 [36], 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 [6, 208, 145].

While there have been many recent works on modeling information diffusion [72, 70, 49, 36, 56] and network evolution [31, 123, 124], most of them treat these two stochastic processes independently and separately, ignoring the influence one may have on the other over time. 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 chapter, we propose a probabilistic generative model, COEVOLVE, for the joint dynamics of information diffusion and network evolution. Our 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 2.1:

I. **Information diffusion process.** We design an "identity revealing" multivariate Hawkes process [133] 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 [22, 106, 229, 228, 55, 131, 48, 195], 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 [102, 201], 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 retweet-

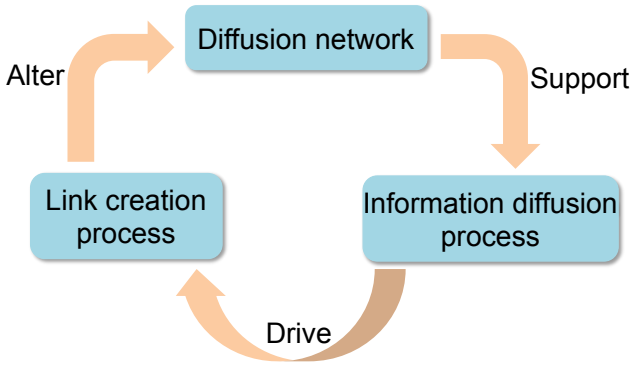



Figure 2.1: Illustration of how information diffusion and network structure processes interact

ing 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 2.2.

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.

- **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),

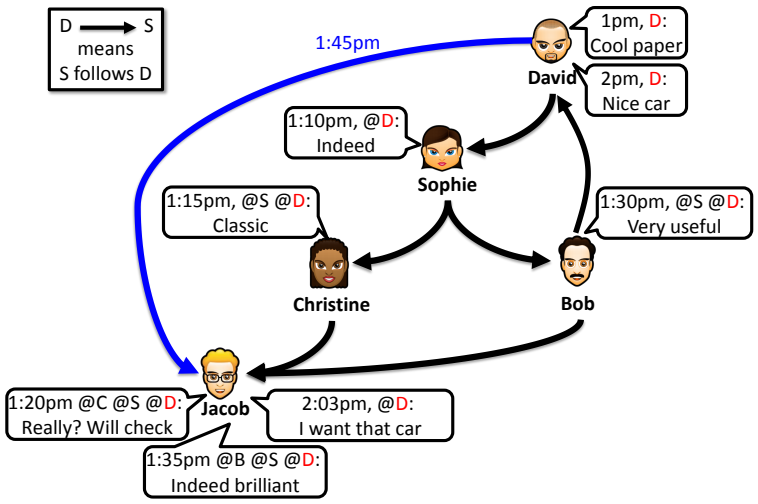
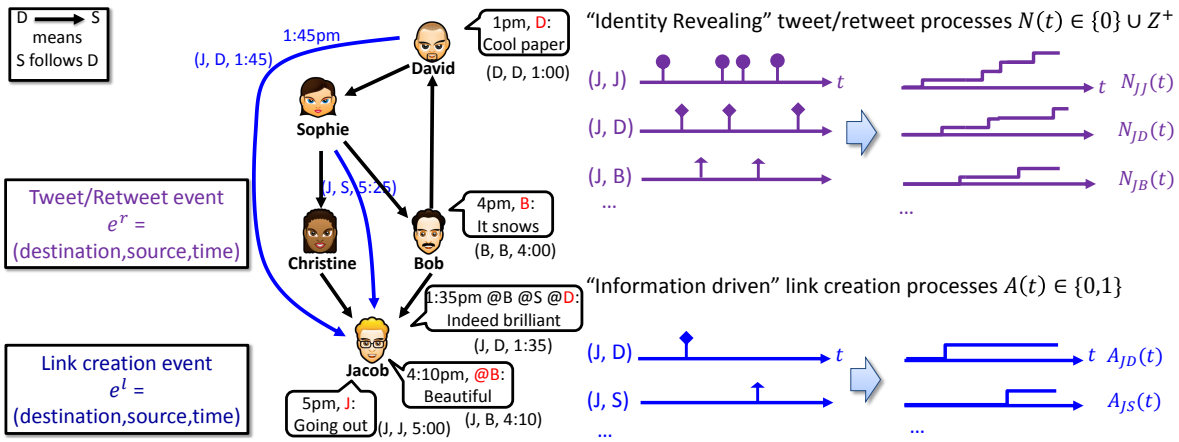



Figure 2.2: Illustration of information diffusion and network structure co-evolution: David's tweet at 1:00 pm about a paper is retweeted by Sophie and Christine respectively at 1:10 pm and 1:15 pm to reach out to Jacob. Jacob retweets about this paper at 1:20 pm and 1:35 pm and then finds David a good source of information and decides to follow him directly at 1:45 pm. Therefore, a new path of information to him (and his downstream followers) is created. As a consequence, a subsequent tweet by David about a car at 2:00 pm directly reaches out to Jacob without need to Sophie and Christine retweet.

static network patterns (*e.g.*, node degree) and temporal network patterns (*e.g.*, shrinking diameter) described in related literature [125, 124, 67]. 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 dataset [6].

The remainder of this chapter is organized as follows. We introduce our joint model of information diffusion and network structure co-evolution in Section 2.2. Sections 2.3 and 2.4 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. In Sections 2.5, 2.6, and 2.7 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 real-world dataset, respectively. Finally, section 2.8 reviews the related work.




a) Event representation b) Point and counting processes

Figure 2.3: Events as point and counting processes. Panel (a) shows a trace of events generated by a tweet from David followed by new links Jacob creates to follow David and Sophie. Panel (b) shows the associated points in time and the counting process realization.

## 2.2 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.

### 2.2.1 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 2.3(a):

$$e^r \quad \text{or} \quad e^l \quad := \quad (\underset{\uparrow}{u}, \quad \overset{\downarrow}{s}, \quad \underset{\uparrow}{t}). \qquad (2.1)$$

destination source time

**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 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

$$\mathcal{H}^r_{us}(t) = \left\{ e^r_i = (u_i, s_i, t_i) | u_i = u \text{ and } s_i = s \right\}. \tag{2.2}$$

The entire history of retweet events is denoted as

$$\mathcal{H}^r(t) := \cup_{u,s\in[m]} \mathcal{H}^r_{us}(t) \tag{2.3}$$

**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. We denote the link creation history as $\mathcal{H}^l(t)$.

### 2.2.2 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 $d\boldsymbol{N}(t) := \left( dN_{us}(t) \right)_{u,s\in[m]}$.

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 $d\boldsymbol{A}(t) := \big( \, dA_{us}(t) \, \big)_{u,s \in [m]}$.

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

$$\mathbb{E}[d\boldsymbol{N}(t) \,|\, \mathcal{H}^r(t) \cup \mathcal{H}^l(t)] = \boldsymbol{\Gamma}^*(t)\, dt \tag{2.4}$$

$$\mathbb{E}[d\boldsymbol{A}(t) \,|\, \mathcal{H}^r(t) \cup \mathcal{H}^l(t)] = \boldsymbol{\Lambda}^*(t)\, dt, \tag{2.5}$$

where,

$$\boldsymbol{\Gamma}^*(t) = \big( \, \gamma^*_{us}(t) \, \big)_{u,s \in [m]} \tag{2.6}$$

$$\boldsymbol{\Lambda}^*(t) = \big( \, \lambda^*_{us}(t) \, \big)_{u,s \in [m]}. \tag{2.7}$$

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.

### 2.2.3 Information Diffusion Process

We model the intensity, $\boldsymbol{\Gamma}^*(t)$, for retweeting events using multivariate Hawkes process [133]:

$$\gamma^*_{us}(t) = \mathbb{I}[u = s]\, \eta_u + \mathbb{I}[u \neq s]\, \beta_s \sum\nolimits_{v \in \mathcal{F}_u(t)} \kappa_{\omega_1}(t) \star \big( A_{uv}(t)\; dN_{vs}(t) \big), \tag{2.8}$$

where $\mathbb{I}[\cdot]$ is the indicator function and $\mathcal{F}_u(t) := \big\{ v \in [m] : A_{uv}(t) = 1 \big\}$ 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 $\beta_s \sum_{v \in \mathcal{F}_u(t)} \kappa_\omega(t) \star \big( A_{uv}(t)\; dN_{vs}(t) \big)$ 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 $\mathbf{\Gamma}^*(t)$ 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 2.2.4. Hence the model design captures the influence of the network topology and each source's influence, $\beta_s$, 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 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 [111] which allow at most one event per node per cascade.

### 2.2.4 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 datasets. 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:

$$\lambda^*_{us}(t) = (1 - A_{us}(t)) \left( \mu_u + \alpha_u \sum_{v \in \mathcal{F}_u(t)} \kappa_{\omega_2}(t) \star dN_{vs}(t) \right), \tag{2.9}$$

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.

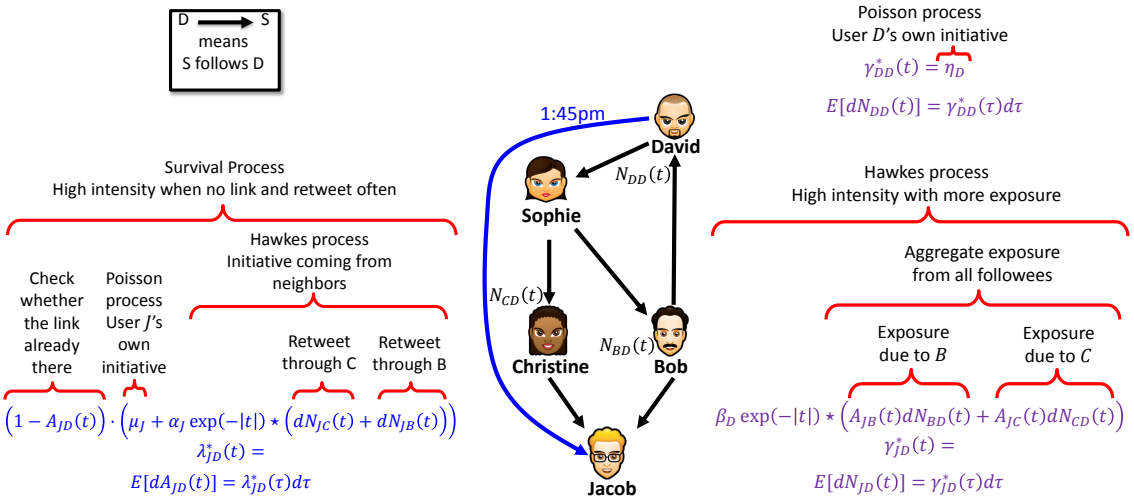
In the second category, the person decides to explicitly like or forward the associated post and influencing events correspond to the times when she does so. In this case, we model the intensity, $\mathbf{\Lambda}^*(t)$, for link creation as:

$$\lambda^*_{us}(t) = (1 - A_{us}(t))(\mu_u + \alpha_u \, \kappa_{\omega_2}(t) \star dN_{us}(t)), \tag{2.10}$$

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 (2.9). 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 2.4 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



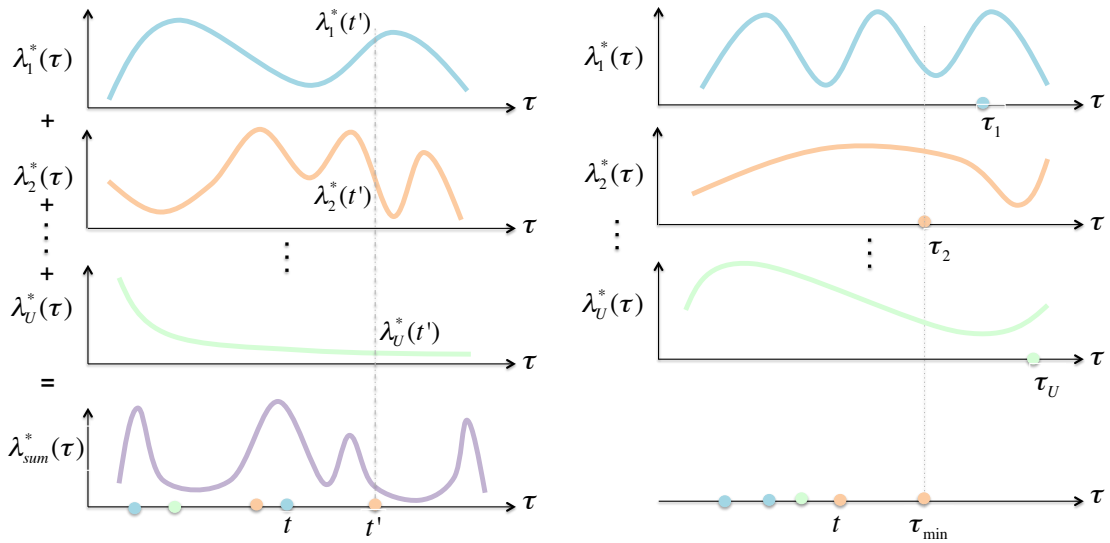

a) Link creation process b) Social network c) Information diffusion process

Figure 2.4: The breakdown of conditional intensity functions for 1) information diffusion process of Jacob retweeting posts originated from David $N_{JD}(t)$; 2) information diffusion process of David tweeting on his own initiative $N_{DD}(t)$; 3) link creation process of Jacob following David $A_{JD}(t)$

datasets, 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 datasets, 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.

## 2.3 Efficient Simulation of Coevolutionary Dynamics

We could simulate samples (link creations, tweets and retweets) from our model by adapting Ogata's thinning algorithm [148], 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 $\mathbf{\Gamma}^*(t)$ and $\mathbf{\Lambda}^*(t)$. Thus, to draw $n$ sample events, we would need to perform $O(m^2n^2)$ operations, where $m$ is the number of nodes. Figure 2.5(a) schematically demonstrates the main steps of Ogata's algorithm. Please refer to section 1.3 for further details.




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

Figure 2.5: Ogata's algorithm vs our simulation algorithm in simulating $U$ interdependent point processes characterized by intensity functions $\lambda_1(t), \ldots, \lambda_U(t)$. Panel (a) illustrates 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. Panel (b) illustrates our proposed algorithm, which first draws a sample from each dimension independently and then takes the minimum time among them.

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 $\mathbf{\Gamma}^*(t)$ and $\mathbf{\Lambda}^*(t)$ 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.

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 2.5 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.

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.

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 $\tau_u$ 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) $\tau_{min}$ is the first event after time $t$ of the Poisson process with intensity $\lambda_{sum}^*(\tau)$. In other words, $\tau_{min}$ 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 [167]; *i.e.*, $\tau_u - t \sim Exponential\left(\int_t^{t+\tau_u} \lambda_u^*(\tau)\, d\tau\right)$. We have:

$$\begin{aligned}
\mathbb{P}(\tau_{min} \le x | x > t) &= 1 - \mathbb{P}(\tau_{min} > x | x > t) = 1 - \mathbb{P}(\min(\tau_1, \ldots, \tau_U) > x | x > t) \\
&= 1 - \mathbb{P}(\tau_1 > x, \ldots, \tau_U > x | x > t) = 1 - \prod_{u=1}^{U} \mathbb{P}(\tau_u > x | x > t) \\
&= 1 - \prod_{u=1}^{U} \exp\left(-\int_t^{t+x} \lambda_u^*(\tau)\, d\tau\right) = 1 - \exp\left(-\int_t^{t+x} \lambda_{sum}^*(\tau)\, d\tau\right).
\end{aligned} \tag{2.11}$$

[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.

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

$$\begin{aligned}\mathbb{P}(u_{min} = u | \tau_{min} = x) &= \lambda^*_u(x) \exp\left(-\int_t^{t+x} \lambda^*_u(\tau)\, d\tau\right) \prod_{v \neq u} \exp\left(-\int_t^{t+x} \lambda^*_v(\tau)\, d\tau\right) \\ &= \lambda^*_u(x) \prod_v \exp\left(-\int_t^{t+x} \lambda^*_v(\tau)\, d\tau\right).\end{aligned} \tag{2.12}$$

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

□

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.*

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

$$\mathbb{P}\left\{(t_1, u_1), \ldots, (t_n, u_n)\right\} = \prod_{i=1}^{n} \mathbb{P}\left\{(t_i, u_i) | (t_{i-1}, u_{i-1}), \ldots, (t_1, u_1)\right\} = \prod_{i=1}^{n} \mathbb{P}\left\{(t_i, u_i) | \mathcal{H}_{t_i}\right\} \tag{2.13}$$

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 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 $i$, $\mathbb{P}\left\{(t_i, u_i) | \mathcal{H}_{t_i}\right\}$ is the same for both algorithms. Thus, the multiplication of individual terms is also equal for both. This will prove the theorem. □

This new algorithm is specially suitable for the structure of our inter-coupled processes.

**Algorithm 2** Simulation Algorithm for COEVOLVE

**Initialization:**
Initialize the priority queue $Q \ \forall \ u, s \in [m]$
Sample next link event $e^l_{us}$ from $A_{us}$ (Algorithm 4)
$Q.insert(e^l_{us})$
Sample next retweet event $e^r_{us}$ from $N_{us}$ (Algorithm 4)
$Q.insert(e^r_{us})$
**General Subroutine:**
$t \leftarrow 0 \ t < T$
$e \leftarrow Q.extract\_min()$ $e = (u, s, t')$ is a retweet event
Update the history $\mathcal{H}^r_{us}(t') = \mathcal{H}^r_{us}(t) \cup \{e\} \ \forall \ v \ s.t. \ u \rightsquigarrow v$
Update event intensity: $\gamma_{vs}(t') = \gamma_{vs}(t'^{-}) + \beta$
Sample retweet event $e^r_{vs}$ from $\gamma_{vs}$ (Algorithm 4)
$Q.update\_key(e^r_{vs})$ NOT $s \rightsquigarrow v$
Update link intensity: $\lambda^*_{vs}(t') = \lambda^*_{vs}(t'^{-}) + \alpha$
Sample link event $e^l_{vs}$ from $\lambda_{vs}$ (Algorithm 4)
$Q.update\_key(e^l_{vs})$
Update the history $\mathcal{H}^l_{us}(t') = \mathcal{H}^l_{us}(t) \cup \{e\}$
$\lambda^*_{us}(t) \leftarrow 0 \ \forall \ t > t'$
$t \leftarrow t'$

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 $\lambda^*(t) = \mu + \sum_{t_i \in \mathcal{H}_t} \alpha \, \omega \exp(-\omega(t - t_i))$. Note that both link creation and information diffusion processes have this structure. Now, let $t_i < t_{i+1}$ be two arbitrary times,

**Algorithm 3** 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 4** 1-D next event sampling

**Input:** Current time: $t$
**Output:** Next event time: $s$
$s \leftarrow t$
$\hat{\lambda} \leftarrow \lambda^*(s)$ (Algorithm 3) $s < T$
$g \sim Exponential(\hat{\lambda})$
$s \leftarrow s + g$
$\bar{\lambda} \leftarrow \lambda^*(s)$ (Algorithm 3)
Rejection test:
$d \sim Uniform(0, 1)$ $d \times \hat{\lambda} < \bar{\lambda}$
**return** $s$
$\hat{\lambda} = \bar{\lambda}$
**return** $s$

we have

$$\lambda^*(t_{i+1}) = (\lambda^*(t_i) - \mu)\exp(-\omega(t_{i+1} - t_i)) + \mu. \tag{2.14}$$

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 3 summarizes the procedure to compute intensities with exponential kernels, and Algorithm 4 shows the procedure to sample the next event in each dimension making use of the special property of exponential kernel functions.

The simulation algorithm is shown in Algorithm 2. By using this algorithm we reduce the complexity from $O(n^2m^2)$ 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 interleaving fashion, since every new retweet event will modify the intensity for link creation and vice versa.

## 2.4 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.

### 2.4.1 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 (1.7), *i.e.*,

$$\begin{aligned}\mathfrak{L}(\{\mu_u\}, \{\alpha_u\}, \{\eta_u\}, \{\beta_s\}) = &\underbrace{\sum_{e_i^r \in \mathcal{E}} \log\left(\gamma^*_{u_i s_i}(t_i)\right) - \sum_{u,s \in [m]} \int_0^T \gamma^*_{us}(\tau)\, d\tau}_{\text{tweet / retweet}} \\ &+ \underbrace{\sum_{e_i^l \in \mathcal{A}} \log\left(\lambda^*_{u_i s_i}(t_i)\right) - \sum_{u,s \in [m]} \int_0^T \lambda^*_{us}(\tau)\, d\tau}_{\text{links}}.\end{aligned} \tag{2.15}$$

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:

$$\begin{aligned} &\text{minimize}_{\{\mu_u\},\{\alpha_u\},\{\eta_u\},\{\beta_s\}} && -\mathfrak{L}(\{\mu_u\}, \{\alpha_u\}, \{\eta_u\}, \{\beta_s\}) \\ &\text{subject to} && \mu_u \geq 0, \quad \alpha_u \geq 0 \quad \eta_u \geq 0, \quad \beta_s \geq 0 \quad \forall\, u, s \in [m]. \end{aligned} \tag{2.16}$$

**Theorem 3.** *The optimization problem defined by Equation* (2.16) *is jointly convex.*

*Proof.* We expand the likelihood by replacing the intensity functions into Equation (2.15):

$$\begin{aligned} \mathfrak{L} = &\sum_{e_i^r \in \mathcal{E}} \log \left( \mathbb{I}[u_i = s_i]\, \eta_{u_i} + \mathbb{I}[u_i \neq s_i]\, \beta_{s_i} \sum\nolimits_{v \in \mathcal{F}_{u_i(t_i)}} \Big( \kappa_{\omega_1}(t) \star \big( A_{u_i v}(t)\ dN_{v s_i}(t) \big) \Big) \Big|_{t=t_i} \right) \\ &- \sum_{u,s \in [m]} \mathbb{I}[u = s]\, \eta_u \int_0^T dt + \mathbb{I}[u \neq s]\, \beta_s \sum\nolimits_{v \in \mathcal{F}_u(t)} \int_0^T \kappa_{\omega_1}(t) \star \big( A_{uv}(t)\ dN_{vs}(t) \big)\ dt \\ &+ \sum_{e_i^l \in \mathcal{A}} \log \left( \mu_{u_i} + \alpha_{u_i} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \big( \kappa_{\omega_2}(t) \star dN_{vs}(t) \big) \Big|_{t=t_i} \right) \\ &- \sum_{u,s \in [m]} \mu_u \int_0^T (1 - A_{us}(t))\, dt + \alpha_u \int_0^T (1 - A_{us}(t)) \Big( \sum_{v \in \mathcal{F}_u(t)} \kappa_{\omega_2}(t) \star dN_{vs}(t) \Big)\, dt \end{aligned} \tag{2.17}$$

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 $\boldsymbol{x}$ [24], 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. □

It's notable that the optimization problem decomposes in $m$ independent problems, one per node $u$, and can be readily parallelized.

### 2.4.2 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 [229, 228], which leverages Minorization Maximization (MM) [103] 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 (2.15). More specifically, consider a set of arbitrary auxiliary variable $\nu_{ij}$, 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

$$\forall\ 1 \leq i \leq n_l : \quad \nu_{i1}, \nu_{i2} \geq 0, \quad \nu_{i1} + \nu_{i2} = 1 \tag{2.18}$$

Then, we can lower bound the logarithm in Equation (2.17) using Jensen's inequality as follows:

$$\begin{aligned}
&\log\left(\mu_{u_i} + \alpha_{u_i} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t)\big)\Big|_{t=t_i}\right) \\
&= \log\left(\nu_{i1}\frac{\mu_{u_i}}{\nu_{i1}} + \nu_{i2}\frac{\alpha_{u_i}}{\nu_{i2}} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t)\big)\Big|_{t=t_i}\right) \\
&\geq \nu_{i1}\log\left(\frac{\mu_{u_i}}{\nu_{i1}}\right) + \nu_{i2}\log\left(\frac{\alpha_{u_i}}{\nu_{i2}} \sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t)\big)\Big|_{t=t_i}\right) \\
&\geq \nu_{i1}\log(\mu_{u_i}) + \nu_{i2}\log(\alpha_{u_i}) + \nu_{i2}\log\left(\sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t)\big)\Big|_{t=t_i}\right) \\
&\quad - \nu_{i1}\log(\nu_{i1}) - \nu_{i2}\log(\nu_{i2}).
\end{aligned} \tag{2.19}$$

Now, we can lower bound the log-likelihood given by Equation (2.17) as:

$$
\begin{aligned}
\mathfrak{L} \geq \mathfrak{L}' = & \sum_{e_i^r \in \mathcal{E}} \mathbb{I}[u_i = s_i] \, \log\left(\eta_{u_i}\right) + \sum_{e_i^r \in \mathcal{E}} \mathbb{I}[u_i \neq s_i] \, \log(\beta_{s_i}) \\
& + \sum_{e_i^r \in \mathcal{E}} \mathbb{I}[u_i \neq s_i] \, \log\Big(\sum\nolimits_{v \in \mathcal{F}_{u_i(t_i)}} \Big(\kappa_{\omega_1}(t) \star \big(A_{u_i v}(t) \; dN_{v s_i}(t)\big)\Big)\Big|_{t=t_i}\Big) \\
& - \sum_{u,s \in [m]} \eta_u T + \beta_s \sum\nolimits_{v \in \mathcal{F}_u(t)} \int_0^T \kappa_{\omega_1}(t) \star \big(A_{uv}(t) \; dN_{vs}(t)\big) \; dt \\
& + \sum_{e_i^l \in \mathcal{A}} \nu_{i1} \log(\mu_{u_i}) + \nu_{i2} \log(\alpha_{u_i}) + \nu_{i2} \log\Big(\sum_{v \in \mathcal{F}_{u_i}(t_i)} \big(\kappa_{\omega_2}(t) \star dN_{vs}(t)\big)\Big|_{t=t_i}\Big) \\
& - \sum_{e_i^l \in \mathcal{A}} \nu_{i1} \log(\nu_{i1}) + \nu_{i2} \log(\nu_{i2}) \\
& - \sum_{u,s \in [m]} \mu_u \int_0^T (1 - A_{us}(t)) \, dt + \alpha_u \int_0^T (1 - A_{us}(t))(\kappa_{\omega_2}(t) \star dN_{us}(t)) \, dt
\end{aligned}
\tag{2.20}
$$

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:

$$\eta_u = \frac{\sum_{e_i^r \in \mathcal{E}} \mathbb{I}[u = u_i = s_i]}{T} \tag{2.21}$$

$$\beta_s = \frac{\sum_{e_i^r \in \mathcal{E}} \mathbb{I}[s = s_i \neq u_i]}{\sum_{u \in [m]} \mathbb{I}[u \neq s] \; \sum_{v \in \mathcal{F}_u(t)} \int_0^T \kappa_{\omega_1}(t) \star \big(A_{uv}(t) \; dN_{vs}(t)\big) \; dt} \tag{2.22}$$

$$\mu_u = \frac{\sum_{e_i^l \in \mathcal{A}} \mathbb{I}[u = u_i] \, \nu_{i1}}{\sum_{s \in [m]} \int_0^T (1 - A_{us}(t)) \, dt} \tag{2.23}$$

$$\alpha_u = \frac{\sum_{e_i^l \in \mathcal{A}} \mathbb{I}[u = u_i] \, \nu_{i2}}{\sum_{s \in [m]} \int_0^T (1 - A_{us}(t))(\kappa_{\omega_2}(t) \star dN_{us}(t)) \, dt}. \tag{2.24}$$

Finally, although the lower bound is valid for every choice of $\nu_{ij}$ satisfying Equation (2.18), by maximizing the lower bound with respect to the auxiliary variables we can make sure that the lower bound is tight:

$$
\begin{aligned}
\text{maximize}_{\{\nu_{ij}\}} \quad & \mathcal{L}'(\{\mu_u\}, \{\alpha_u\}, \{\eta_u\}, \{\beta_s\}, \{\nu_{ij}\}) \\
\text{subject to} \quad & \nu_{i1} + \nu_{i2} = 1 \qquad \forall \, i : 1 \leq i \leq n_l \\
& \nu_{i0}, \nu_{i1} \geq 0 \qquad \forall \, i : 1 \leq i \leq n_l.
\end{aligned}
\tag{2.25}
$$

Fortunately, the above constrained optimization problem can be solved easily via Lagrange

**Algorithm 5** 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:** $u \leftarrow 1$ to $m$

Initialize $\mu_u$ and $\alpha_u$ randomly $u \leftarrow 1$ to $m$

$\eta_u = \frac{\sum_{e_i^r \in \mathcal{E}} \mathbb{I}[u=u_i=s_i]}{T}$ $s \leftarrow 1$ to $m$

$\beta_s = \frac{\sum_{e_i^r \in \mathcal{E}} \mathbb{I}[s=s_i \neq u_i]}{\sum_{u\in[m]} \mathbb{I}[u\neq s] \sum_{v\in\mathcal{F}_u(t)} \int_0^T \kappa_{\omega_1}(t) \star \big(A_{uv}(t)\ dN_{vs}(t)\big)\, dt}$ not converged $i \leftarrow 1$ to $n_l$

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

$\nu_{i2} = \frac{\alpha_{u_i} \sum_{v\in\mathcal{F}_{u_i}(t_i)} \left(\kappa_{\omega_2}(t) \star dN_{vs}(t)\right)\Big|_{t=t_i}}{\mu_{u_i} + \alpha_{u_i} \sum_{v\in\mathcal{F}_{u_i}(t_i)} \left(\kappa_{\omega_2}(t) \star dN_{vs}(t)\right)\Big|_{t=t_i}}$ $u \leftarrow 1$ to $m$

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

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

multipliers, which leads to closed form updates:

$$\nu_{i1} = \frac{\mu_{u_i}}{\mu_{u_i} + \alpha_{u_i} \sum_{v\in\mathcal{F}_{u_i}(t_i)} \left(\kappa_{\omega_2}(t) \star dN_{vs}(t)\right)\Big|_{t=t_i}} \tag{2.26}$$

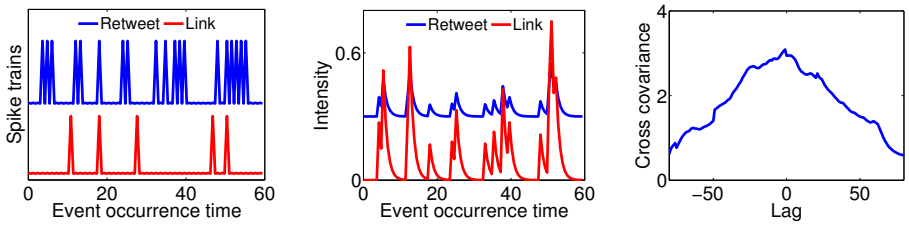
$$\nu_{i2} = \frac{\alpha_{u_i} \sum_{v\in\mathcal{F}_{u_i}(t_i)} \left(\kappa_{\omega_2}(t) \star dN_{vs}(t)\right)\Big|_{t=t_i}}{\mu_{u_i} + \alpha_{u_i} \sum_{v\in\mathcal{F}_{u_i}(t_i)} \left(\kappa_{\omega_2}(t) \star dN_{vs}(t)\right)\Big|_{t=t_i}}. \tag{2.27}$$

Algorithm 5 summarizes the learning procedure. It is guaranteed to converge to a global optimum [103, 229]

## 2.5 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 [2]. 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.

[2]The code is available at https://github.com/farajtabar/coevolution




(a) (b) (c)

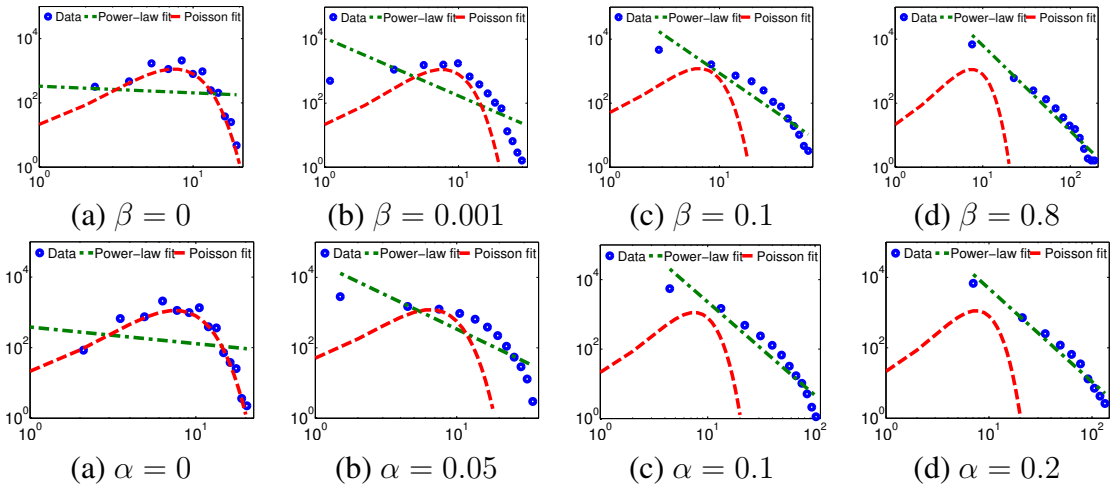
Figure 2.6: Coevolutionary dynamics for synthetic data. a) Spike trains of link and retweet events. b) Link and retweet intensities. c) Cross covariance of link and retweet intensities.

### 2.5.1 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 2. 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 $\mu$ determines the independent growth of the network – roughly speaking, the expected number of links each user establishes spontaneously before time $T$ is $\mu T$. Whenever we investigate a static property, we choose the same sparsity level of $0.001$.

### 2.5.2 Retweet and Link Coevolution

Figures 2.6(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 2.6(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.




(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) $\alpha = 0.2$

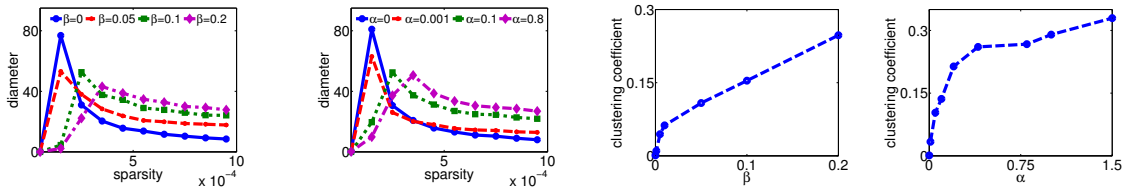
Figure 2.7: Degree distributions when network sparsity level reaches 0.001 for different $\beta$ ($\alpha$) values and fixed $\alpha = 0.1$ ($\beta = 0.1$).

### 2.5.3 Degree Distribution

Empirical studies have shown that the degree distribution of online social networks and microblogging sites follow a power law [31, 116], 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 $\alpha$ and $\beta$, the closer the resulting degree distribution follows a power-law. This is because the network grows more locally. Interestingly, the lower their values, the closer the distribution to an Erdos-Renyi random graph [53], because, the edges are added almost uniformly and independently without influence from the local structure. Figure 2.7 confirms this intuition by showing the degree distribution for different values of $\beta$ and $\alpha$.

### 2.5.4 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 [10, 31, 125]. Figures 2.8(a-b) show the diameter on the largest connected component (LCC) against the sparsity of the network over time for different values of $\alpha$ and $\beta$. Although at




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

Figure 2.8: Diameter and clustering coefficient for network sparsity 0.001. Panels (a) and (b) show the diameter against sparsity over time for fixed $\alpha = 0.1$, and for fixed $\beta = 0.1$ respectively. Panels (c) and (d) show the clustering coefficient (CC) against $\beta$ and $\alpha$, respectively.

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 $\alpha$ or $\beta$ 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.

### 2.5.5 Clustering Coefficient

Triadic closure [79, 123, 166] has been often presented as a plausible link creation mechanism. However, different social networks and microblogging sites present different levels of triadic closure [194]. Importantly, our method is able to generate networks with different levels of triadic closure, as shown by Figure 2.8(c-d), where we plot the clustering coefficient [206], which is proportional to the frequency of triadic closure, for different values of $\alpha$ and $\beta$.

### 2.5.6 Network Visualization

Figure 2.9 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 $\beta$. 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 $\beta = 0.8$

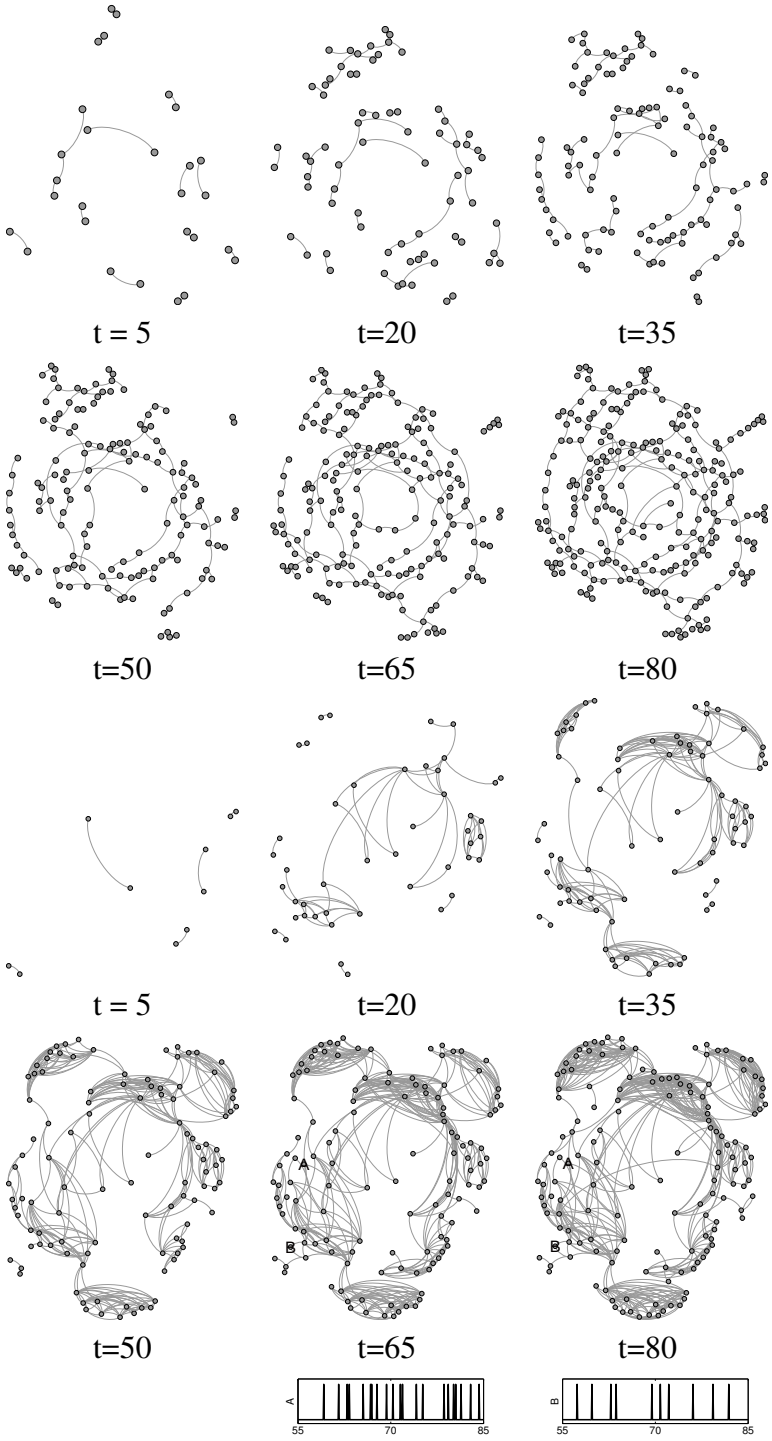



Figure 2.9: Evolution of two networks: one with $\beta = 0$ (1st and 2nd rows) and another one with $\beta = 0.8$ (3rd and 4th rows), and spike trains of nodes A and B (5th row).

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[3]. 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.

Figure 2.10 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 2.11 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.

### 2.5.7 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 reported in [67]. Figure 2.12 summarizes the results, which provide empirical evidence that the higher

[3]http://gephi.github.io/

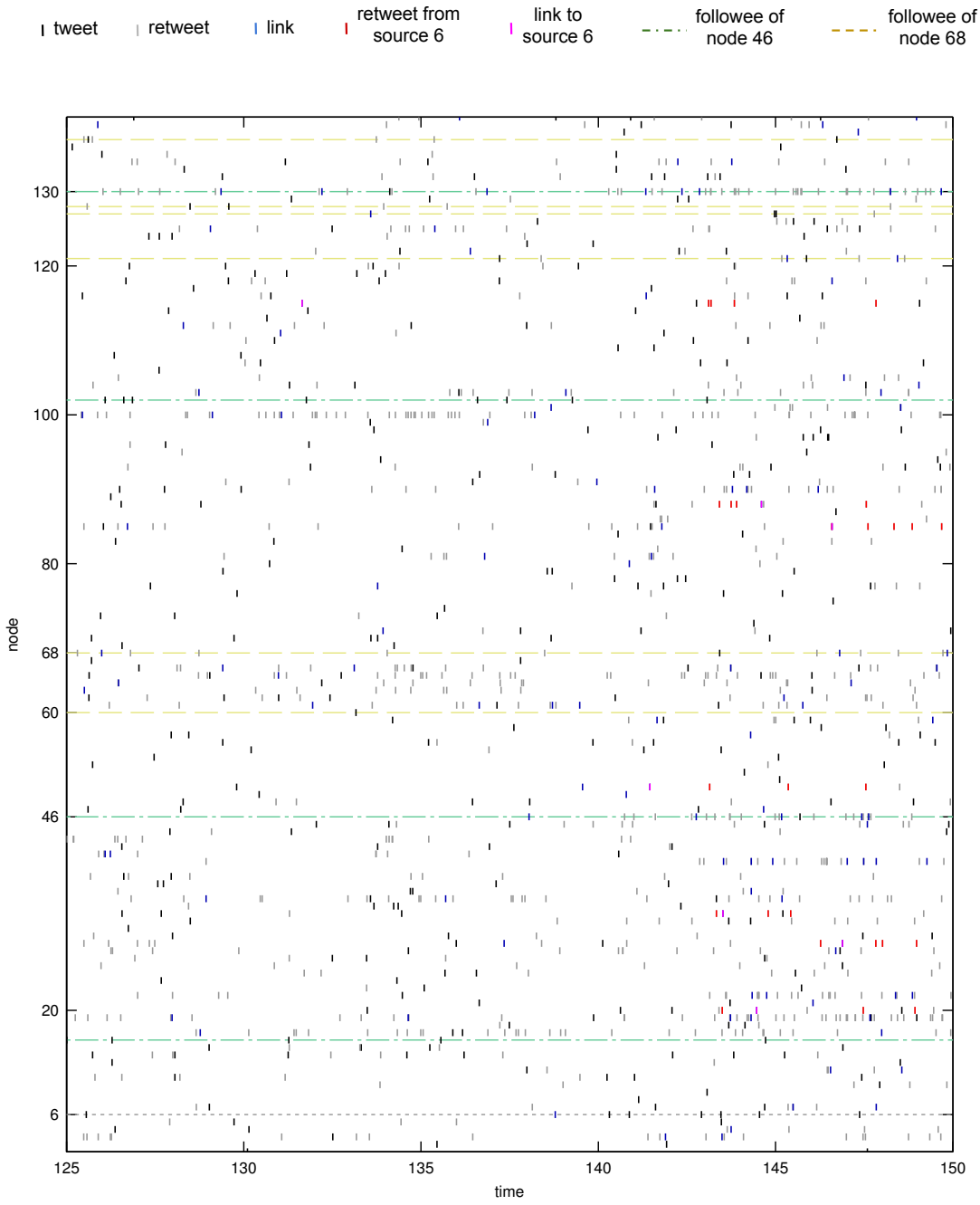



Figure 2.10: Coevolutionary dynamics of events for the network shown in Figure 2.11. Information Diffusion $\longrightarrow$ Network Evolution: When node 6 joins the network a few nodes follow her and retweet her posts. Her tweets being propagated (shown in red) turning her to a valuable source of information. Therefore, those retweets are followed by links created to her (shown in magenta). Network Evolution $\longrightarrow$ Information Diffusion: Nodes 46 and 68 both have almost the same number of followees. However, as soon as node 46 connects to node 130 (which is a central node and retweets very much) her activity dramatically increases compared to node 68.

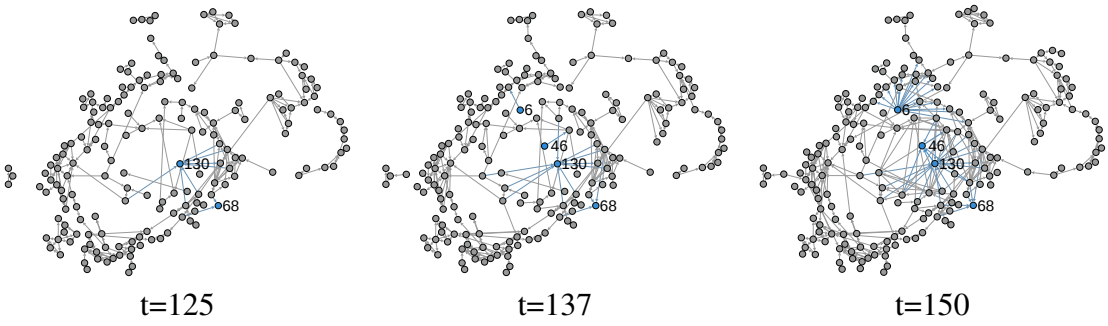



Figure 2.11: Network structure in which events from Figure 2.10 take place, at different times.

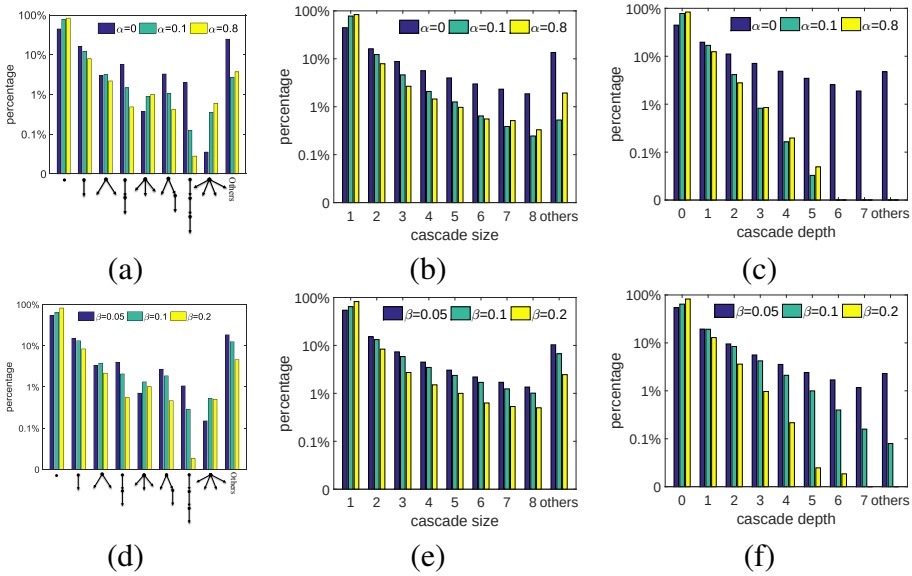
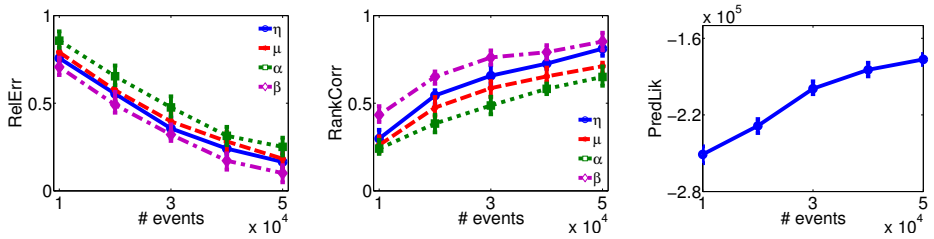



Figure 2.12: Distribution of cascade structure, size and depth for different $\alpha$ ($\beta$) values and fixed $\beta = 0.2$ ($\alpha = 0.8$).

the $\alpha$ ($\beta$) value, the shallower and wider the cascades.

## 2.6 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 [208, 6, 145] at predicting new links, and a baseline Hawkes process that does not consider network evolution at predicting new events.




(a) Relative MAE (b) Rank correlation (c) Test log-likelihood

Figure 2.13: Performance of model estimation for a 400-node synthetic network.

### 2.6.1 Experimental Setup

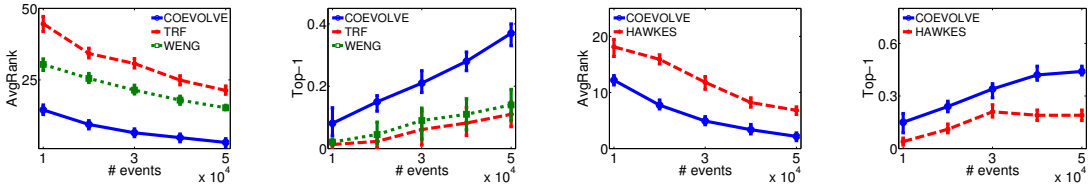
Throughout this section, we experiment with our model considering $m{=}400$ nodes. We set the model parameters for each node in the network by drawing samples from $\mu{\sim}U(0, 0.0004)$, $\alpha{\sim}U(0, 0.1)$, $\eta{\sim}U(0, 1.5)$ and $\beta{\sim}U(0, 0.1)$. We then sample up to 60,000 link and information diffusion events from our model using Algorithm 2 and average over 8 different simulation runs.

### 2.6.2 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. Figure 2.13 shows that as we feed more events into the estimation procedure, the estimation becomes more accurate.

### 2.6.3 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 state of the art methods, which we denote as TRF [6] and WENG [208]. TRF measures the probability of creating a link from a source at a given time by simply computing the




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

Figure 2.14: Prediction performance for a 400-node synthetic network by means of average rank (AR) and success probability that the true (test) events rank among the top-1 events (Top-1).

proportion of new links created from the source over all total created links up to the given time. WENG considers several link creation strategies and makes a prediction by combining these strategies.

Here, we evaluate the performance by computing the probability of all potential links using our model, TRF and WENG and then compute (i) the average rank of all true (test) events (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 2.14 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.

### 2.6.4 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 via the same two measures as in the link prediction task and summarize the results in Figure 2.14 against an increasing number of training events. The results show that, by modeling the network evolution, our model performs significantly better than the baseline.

## 2.7 Experiments on Coevolution and Prediction on Real Data

In this section, we validate our model using a large Twitter dataset containing nearly 550,000 tweet, retweet and link events from more than 280,000 users [6]. We will show that our model can capture the co-evolutionary dynamics and, by doing so, it predicts retweet and link creation events more accurately than several alternatives.

### 2.7.1 Dataset Description & Experimental Setup

We use a dataset that contains both link events as well as tweets/retweets from millions of Twitter users [6]. In particular, the dataset 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.

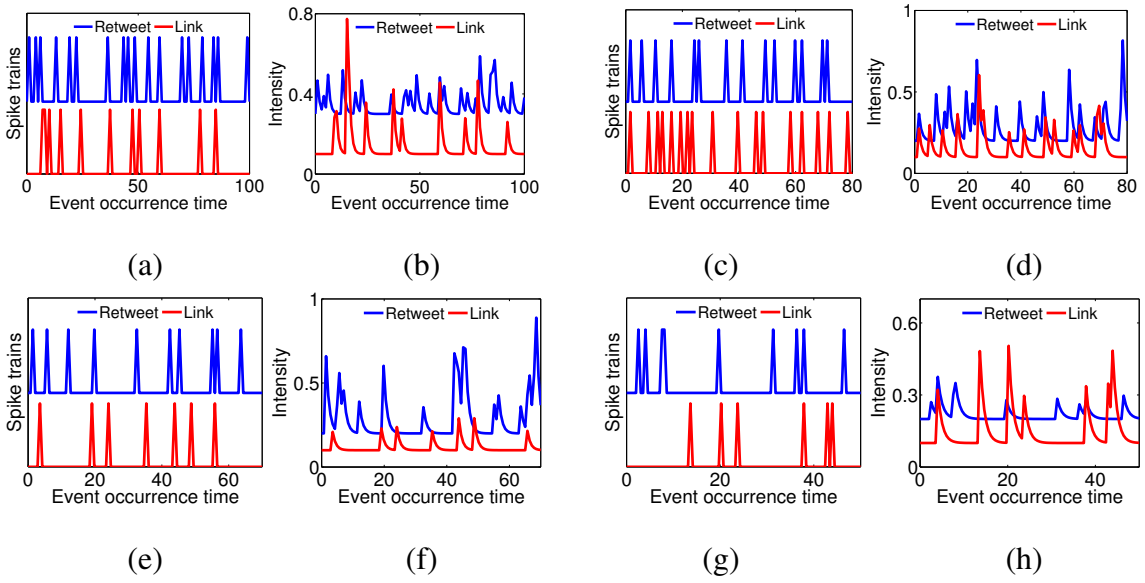



Figure 2.15: Link and retweet behavior of 4 typical users in the real-world dataset. Panels (a,c,e,g) show the spike trains of link and retweet events and Panels (b,d,f,h) show the estimated link and retweet intensities

### 2.7.2 Retweet and Link Coevolution

Figures 2.15 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 2.16 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 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 2.5, 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

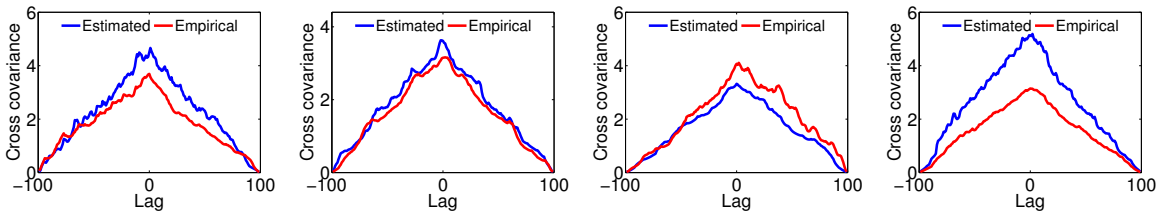



Figure 2.16: Empirical and simulated cross covariance of link and retweet intensities for 4 typical users.

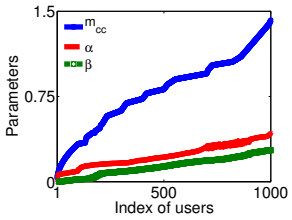


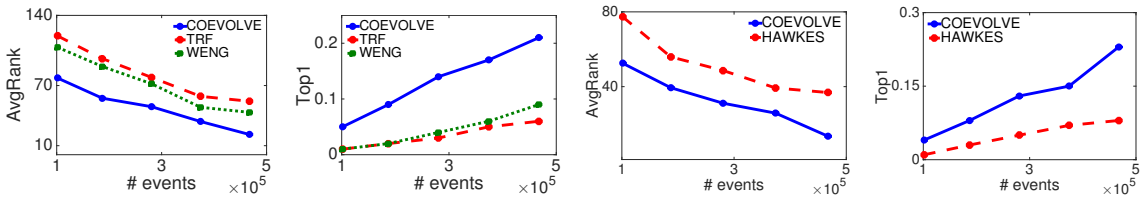

Figure 2.17: Empirical cross covariance and learned model parameters for 1,000 users, picked at random

could expect that our model estimation method should assign higher $\alpha$ and/or $\beta$ values to users with high $m_{cc}$. Figure 2.17 confirms this intuition on 1,000 users, picked at random. Whenever a user has high $\alpha$ and/or $\beta$ value, she exhibits a high cross covariance between her created links and retweets.

### 2.7.3 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 [6] and WENG [208].

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 2.18(a-b), where we consider an increasing number of training retweet/tweet events. Our model outperforms




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

Figure 2.18: Prediction performance in the Twitter dataset by means of average rank (AR) and success probability that the true (test) events rank among the top-1 events (Top-1).

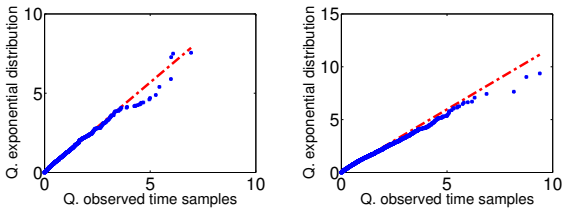
TRF and WENG consistently. For example, for $8 \cdot 10^4$ training events, our model achieves a SP $2.5$x times larger than TRF and WENG.

### 2.7.4 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 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 2.18(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.

### 2.7.5 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 [40], the intensity integrals $\int_{t_i}^{t_{i+1}} \lambda(t)\, dt$ should conform to the unit-rate exponential distribution. Figure 2.19 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.




(a) Link process (b) Retweet process

Figure 2.19: Quantile plots of the intensity integrals from the real link and retweet event time

## 2.8 Related Work

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, [20] 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* [80, 82, 174]. 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. It has been shown that adaptive networks are capable of self-organizing towards dynamically critical states, like phase transitions by the interplay between the two processes on different time scales [23]. In a different context, epidemiologists have found that nodes may rewire their links to try to avoid contact with the infected ones [81, 221]. 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 [230]. 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 [180], 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.

The work most closely related to ours is the empirical study of information diffusion and network evolution [80, 179, 208, 6, 145]. Among them, [208] 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 $u$. [6] 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. [145] 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, [191] 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 [208, 6] predict whether a new link will be created based on the number of retweets; and, [145] 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.

## 2.9 Summary and Conclusion

We proposed a joint continuous-time model of information diffusion and network evolution, which can capture the coevolutionary dynamics, mimics the most common static and temporal network patterns observed in real-world networks and information diffusion data, and predicts the network evolution and information diffusion more accurately than previous state-of-the-arts. Using point processes to model intertwined events in information networks opens up many interesting future modeling work. 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. For example, we can generalize our model to support link deletion by introducing an intensity matrix $\mathbf{\Xi}^*(t)$ modeling link deletions as survival processes, *i.e.*, $\mathbf{\Xi}^*(t) = (g^*_{us}(t)A_{us}(t))_{u,s\in[m]}$, and then consider the counting process $\boldsymbol{A}(t)$ associated with the adjacency matrix to evolve as $\mathbb{E}[d\boldsymbol{A}(t)|\mathcal{H}^r(t) \cup \mathcal{H}^l(t)] = \mathbf{\Lambda}^*(t)\,dt - \mathbf{\Xi}^*(t)\,dt$. We also can consider the number of nodes varying over time. Furthermore, a large and diverse range of point processes can also be used in the framework without changing the efficiency of the simulation and the convexity of the parameter estimation, *e.g.*, condition the intensity on additional external features, such as node attributes.

The content presented in this chapter are mostly based on papers [58, 57]. However the author has applied the framework of point processes to other modeling tasks as well. For example, paper [191] considered the problem of clustering not from static links but from activities over the social network. The author has applied it to model cooperation and competition between information cascades too [223]. Also, he co-authored a paper on using Hawkes process for recommendation systems where there is a peer influence between

users [100]. In another work, he and his colleagues tackle the problem of anomaly/change detection in social network by looking at activities over it [128]. The framework of point processes has been used by the author to model content addition and edition times and dynamics in crowd-generated data like Wikipedia and Stackoverflow too [186] and to model temporal knowledge graphs between real-world entities [192]. Last but not least, is the work of the author with his colleagues for invasive species management in epidemiology where Hawkes process is used to model the dissemination of species in a landscape [88].

# CHAPTER 3

# OPTIMIZATION FOR SINGLE STAGE INTERVENTION

Events in an online social network can be categorized roughly into endogenous events, where users just respond to the actions of their neighbors within the network, or exogenous events, where users take actions due to drives external to the network. How much external drive should be provided to each user, such that the network activity can be steered towards a target state? In this chapter, we model social events using multivariate Hawkes processes, which can capture both endogenous and exogenous event intensities, and derive a time dependent linear relation between the intensity of exogenous events and the overall network activity. Exploiting this connection, we develop a convex optimization framework for determining the required level of external drive in order for the network to reach a desired activity level. We experimented with event data gathered from Twitter, and show that our method can steer the activity of the network more accurately than alternatives.

## 3.1 Introduction

Online social platforms routinely track and record a large volume of event data, which may correspond to the usage of a service (*e.g.*, url shortening service, bit.ly). These events can be categorized roughly into *endogenous* events, where users just respond to the actions of their neighbors within the network, or *exogenous* events, where users take actions due to drives external to the network. For instance, a user's tweets may contain links provided by bit.ly, either due to his forwarding of a link from his friends, or due to his own initiative to use the service to create a new link.

Can we model and exploit these data to steer the online community to a desired activity level? Specifically, can we drive the overall usage of a service to a certain level (*e.g.*, at least twice per day per user) by incentivizing a small number of users to take more

initiatives? What if the goal is to make the usage level of a service more homogeneous across users? What about maximizing the overall service usage for a target group of users? Furthermore, these *activity shaping* problems need to be addressed by taking into account budget constraints, since incentives are usually provided in the form of monetary or credit rewards.

Activity shaping problems are significantly more challenging than traditional influence maximization problems, which aim to identify a set of users, who, when convinced to adopt a product, shall influence others in the network and trigger a large cascade of adoptions [111, 161]. First, in influence maximization, the state of each user is often assumed to be binary, either adopting a product or not [111, 35, 165, 49]. However, such assumption does not capture the recurrent nature of product usage, where the frequency of the usage matters. Second, while influence maximization methods identify a set of users to provide incentives, they do not typically provide a quantitative prescription on how much incentive should be provided to each user. Third, activity shaping concerns about a larger variety of target states, such as minimum activity requirement and homogeneity of activity, not just activity maximization.

In this chapter, we will address the activity shaping problems using multivariate Hawkes processes [133], which can model both endogenous and exogenous recurrent social events, and were shown to be a good fit for such data in a number of recent works (*e.g.*, [22, 229, 228, 106, 131, 195]). More importantly, we will go beyond model fitting, and derive a novel predictive formula for the overall network activity given the intensity of exogenous events in individual users. Based on this relation, we propose a convex optimization framework to address a diverse range of activity shaping problems given budget constraints. Compared to previous methods for influence maximization, our framework can provide more fine-grained control of network activity, not only steering the network to a desired steady-state activity level but also do so in a time-sensitive fashion. For example, our framework allows us to answer complex time-sensitive queries, such as, which users should be incentivized,

and by how much, to steer a set of users to use a product twice per week after one month?

In addition to the novel framework, we also develop an efficient gradient based optimization algorithm, where the matrix exponential needed for gradient computation is approximated using the truncated Taylor series expansion [4]. This algorithm allows us to validate our framework in a variety of activity shaping tasks and scale up to networks with tens of thousands of nodes. We also conducted experiments on a network of 60,000 Twitter users and more than 7,500,000 uses of a popular url shortening service. Using held-out data, we show that our algorithm can shape the network behavior much more accurately.

### 3.2 Modeling Endogenous-Exogenous Recurrent Social Events

We model the events generated by $m$ users in a social network as a $m$-dimensional counting process $\boldsymbol{N}(t) = (N_1(t), N_2(t), \ldots, N_m(t))^\top$, where $N_i(t)$ records the total number of events generated by user $i$ up to time $t$. Furthermore, we represent each event as a tuple $(u_i, t_i)$, where $u_i$ is the user identity and $t_i$ is the event timing. Let the history of the process up to time $t$ be $\mathcal{H}_t := \big\{(u_i, t_i) \,|\, t_i \leqslant t\big\}$, and $\mathcal{H}_{t-}$ be the history until just before time $t$. Then the increment of the process, $d\boldsymbol{N}(t)$, in an infinitesimal window $[t, t + dt]$ is parametrized by the intensity $\boldsymbol{\lambda}(t) = (\lambda_1(t), \ldots, \lambda_m(t))^\top \geqslant 0$, *i.e.*,

$$\mathbb{E}[d\boldsymbol{N}(t)|\mathcal{H}_{t-}] = \boldsymbol{\lambda}(t)\, dt. \tag{3.1}$$

Intuitively, the larger the intensity $\boldsymbol{\lambda}(t)$, the greater the likelihood of observing an event in the time window $[t, t + dt]$. For instance, a Poisson process in $[0, \infty)$ can be viewed as a special counting process with a constant intensity function $\boldsymbol{\lambda}$, independent of time and history. To model the presence of both endogenous and exogenous events, we will decompose the intensity into two terms

$$\underbrace{\boldsymbol{\lambda}(t)}_{\text{overall event intensity}} = \underbrace{\boldsymbol{\lambda}^{(0)}(t)}_{\text{exogenous event intensity}} + \underbrace{\boldsymbol{\lambda}^{*}(t)}_{\text{endogenous event intensity}}, \tag{3.2}$$

where the exogenous event intensity models drive outside the network, and the endogenous event intensity models interactions within the network. We assume that hosts of social

platforms can potentially drive up or down the exogenous events intensity by providing incentives to users; while endogenous events are generated due to users' own interests or under the influence of network peers, and the hosts do not interfere with them directly. The key questions in the activity shaping context are how to model the endogenous event intensity which are realistic to recurrent social interactions, and how to link the exogenous event intensity to the endogenous event intensity. We assume that the exogenous event intensity is independent of the history and time, *i.e.*, $\boldsymbol{\lambda}^{(0)}(t) = \boldsymbol{\lambda}^{(0)}$.

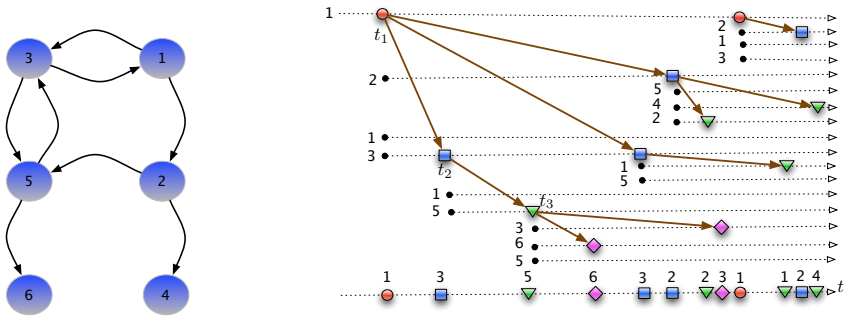
### 3.2.1 Multivariate Hawkes Process

Recurrent endogenous events often exhibit the characteristics of self-excitation, where a user tends to repeat what he has been doing recently, and mutual-excitation, where a user simply follows what his neighbors are doing due to peer pressure. These social phenomena have been made analogy to the occurrence of earthquake [136] and the spread of epidemics [218], and can be well-captured by multivariate Hawkes processes [133] as shown in a number of recent works (*e.g.*, [22, 229, 228, 106, 131, 195]).

More specifically, a multivariate Hawkes process is a counting process who has a particular form of intensity. More specifically, we assume that the strength of influence between users is parameterized by a sparse nonnegative *influence matrix* $\boldsymbol{A} = (a_{uu'})_{u,u' \in [m]}$, where $a_{uu'} > 0$ means user $u'$ directly excites user $u$. We also allow $\boldsymbol{A}$ to have nonnegative diagonals to model self-excitation of a user. Then, the intensity of the $u$-th dimension is

$$\lambda_u^*(t) = \sum\nolimits_{i:t_i<t} a_{uu_i}\, g(t - t_i) = \sum\nolimits_{u' \in [m]} a_{uu'} \int_0^t g(t-s)\, dN_{u'}(s), \tag{3.3}$$

where $g(s)$ is a nonnegative kernel function such that $g(s) = 0$ for $s \leq 0$ and $\int_0^\infty g(s)\, ds < \infty$; the second equality is obtained by grouping events according to users and use the fact that $\int_0^t g(t-s)\, dN_{u'}(s) = \sum_{u_i = u', t_i < t} g(t - t_i)$. Intuitively, $\lambda_u^*(t)$ models the propagation of peer influence over the network — each event $(u_i, t_i)$ occurred in the neighbor of a user will boost her intensity by a certain amount which itself decays over time. Thus, the more frequent the events occur in the user's neighbor, the more likely she will be persuaded to




(a) An example social network (b) Branching structure of events

Figure 3.1: (a) An example social network where each directed edge indicates that the target node *follows*, and can be influenced by, the source node. The activity in this network is modeled using Hawkes processes, which result in branching structure of events in (b). Each exogenous event is the root node of a branch (*e.g.*, top left most red circle at $t_1$), and it occurs due to a user's own initiative; and each event can trigger one or more endogenous events (blue square at $t_2$). The new endogenous events can create the next generation of endogenous events (green triangles at $t_3$), and so forth. The social network in (a) will constrain the branching structure of events in (b), since an event produced by a user (*e.g.*, user 1) can only trigger endogenous events in the same user or one or more of her followers (*e.g.*, user 2 or user 3).

generate a new event.

For simplicity, we will focus on an exponential kernel, $g(t - t_i) = \exp(-\omega(t - t_i))$ in the reminder of the chapter. However, multivariate Hawkes processes is independent of the kernel choice and can be extended to other kernels such as power-law, Rayleigh or any other long tailed distribution over nonnegative real domain. Furthermore, we can rewrite equation (3.3) in vectorial format

$$\boldsymbol{\lambda}^*(t) = \int_0^t \boldsymbol{G}(t - s)\, d\boldsymbol{N}(s), \tag{3.4}$$

by defining a $m \times m$ time-varying matrix $\boldsymbol{G}(t) = (a_{uu'} g(t))_{u,u' \in [m]}$. Note that, for multivariate Hawkes processes, the intensity, $\boldsymbol{\lambda}(t)$, itself is a random quantity, which depends on the history $\mathcal{H}_t$. We denote the expectation of the intensity with respect to history as

$$\boldsymbol{\mu}(t) := \mathbb{E}_{\mathcal{H}_{t-}} \left[ \boldsymbol{\lambda}(t) \right] \tag{3.5}$$

### 3.3 Linking Exogenous Event Intensity to Overall Network Activity

We will develop a closed form relation between the expected intensity $\boldsymbol{\mu}(t) = \mathbb{E}_{\mathcal{H}_{t-}}\left[\boldsymbol{\lambda}(t)\right]$ and the intensity, $\boldsymbol{\lambda}^{(0)}(t)$, of the exogenous events. This relation will form the basis of our activity shaping framework.

**Theorem 4.** $\boldsymbol{\mu}(t) = \boldsymbol{\Psi}(t)\boldsymbol{\lambda}^{(0)} = \big(\boldsymbol{I} + \boldsymbol{A}(\boldsymbol{A} - \omega\boldsymbol{I})^{-1}(e^{(\boldsymbol{A}-\omega\boldsymbol{I})t} - \boldsymbol{I})\big)\boldsymbol{\lambda}^{(0)}$.

*Proof.* Let $u(t)$ refer to step function [1].

$$\boldsymbol{\mu}(t) = \mathbb{E}_{\mathcal{H}_t}[\boldsymbol{\lambda}] = \boldsymbol{\lambda}^{(0)}u(t) + \int_{-\infty}^{t} g(t-\tau)\,\boldsymbol{A}\,\mathbb{E}_{\mathcal{H}_t}[d\boldsymbol{N}(\tau)] \tag{3.6}$$

$$= \boldsymbol{\lambda}^{(0)}u(t) + \int_{-\infty}^{t} g(t-\tau)\,\boldsymbol{A}\,\mathbb{E}_{\mathcal{H}_{d\tau}}[d\boldsymbol{N}(\tau)] \tag{3.7}$$

$$= \boldsymbol{\lambda}^{(0)}u(t) + \int_{-\infty}^{t} g(t-\tau)\,\boldsymbol{A}\,\mathbb{E}_{\mathcal{H}_\tau}[\boldsymbol{\lambda}(s)]\,d\tau \tag{3.8}$$

$$= \boldsymbol{\lambda}^{(0)}u(t) + \int_{0}^{t} g(t-\tau)\,\boldsymbol{A}\,\boldsymbol{\mu}(\tau)\,d\tau \tag{3.9}$$

$$= \boldsymbol{\lambda}^{(0)}u(t) + (\boldsymbol{A}g)(t) * \boldsymbol{\mu}(t). \tag{3.10}$$

Taking the Laplace transform of the above relation, results in

$$\hat{\boldsymbol{\mu}}(s) = \boldsymbol{\lambda}^{(0)}\hat{u}(s) + \boldsymbol{A}\,\hat{g}(s)\,\hat{\boldsymbol{\mu}}(s). \tag{3.11}$$

Solving the above equation for $\hat{\boldsymbol{\mu}}$ we have:

$$\hat{\boldsymbol{\mu}}(s) = (\boldsymbol{I} - \boldsymbol{A}\hat{g}(s))^{-1}\boldsymbol{\lambda}^{(0)}\hat{u}(s) \tag{3.12}$$

$$= (\boldsymbol{I} + \boldsymbol{A}\hat{g}(s) + \boldsymbol{A}^2\hat{g}^2(s) + \boldsymbol{A}^3\hat{g}^3(s) + \cdots)\boldsymbol{\lambda}^{(0)}\hat{u}(s). \tag{3.13}$$

Now, by taking the inverse Laplace we find the expectation of intensity.

$$\boldsymbol{\mu}(t) = (\boldsymbol{I}\delta(t) + \boldsymbol{A}g(t) + \boldsymbol{A}^2g^{*2}(t) + \boldsymbol{A}^3g^{*3}(t) + \cdots) * (\boldsymbol{\lambda}^{(0)}u(t)), \tag{3.14}$$

where, $g^{*k}(t)$ is the inverse Laplace transform of $\hat{g}^k(s)$. For our $g(t) = e^{-\omega t}\,u(t)$ we have $\hat{g}(s) = \frac{1}{s+\omega}$, therefore,

$$g^{*k}(t) = \mathcal{L}^{-1}\{\hat{g}^k(s)\} = \mathcal{L}^{-1}\left\{\frac{1}{(s+\omega)^k}\right\} = \frac{1}{(k-1)!}t^{k-1}e^{-\omega t}u(t) \tag{3.15}$$

[1] Thanks to Ali Zarezade for the discussion on the proof.

The transformer function is then equal to

$$\psi(t) = \big(\boldsymbol{I}\delta(t) + \sum_{k=1}^{\infty} \frac{1}{(k-1)!}\boldsymbol{A}^k t^{k-1} e^{-\omega t}\big)u(t) \tag{3.16}$$

$$= \big(\boldsymbol{I}\delta(t) + \boldsymbol{A}\, e^{-\omega t}\sum_{k=0}^{\infty} \frac{1}{k!}\boldsymbol{A}^k t^k\big)u(t) \tag{3.17}$$

$$= \big(\boldsymbol{I}\delta(t) + \boldsymbol{A}\, e^{(\boldsymbol{A}-\omega\boldsymbol{I})t}\big)u(t) \tag{3.18}$$

To find $\boldsymbol{\mu}(t)$ for the constant base intensity $\boldsymbol{\lambda}^{(0)}$ applied at time $t = 0$ we have:

$$\boldsymbol{\mu}(t) = \psi(t) * (\boldsymbol{\lambda}^{(0)}u(t)) = \int_{-\infty}^{\infty} \big(\boldsymbol{I}\delta(\tau) + \boldsymbol{A}\, e^{(\boldsymbol{A}-\omega\boldsymbol{I})\tau}\big)u(\tau)\, \boldsymbol{\lambda}^{(0)}u(t-\tau)\, d\tau \tag{3.19}$$

$$= \big(\boldsymbol{I} + \int_0^t \boldsymbol{A}\, e^{(\boldsymbol{A}-\omega\boldsymbol{I})\tau}\, d\tau\big)\boldsymbol{\lambda}^{(0)} = \big(\boldsymbol{I} + \boldsymbol{A}(\boldsymbol{A}-\omega\boldsymbol{I})^{-1}(e^{(\boldsymbol{A}-\omega\boldsymbol{I})t} - \boldsymbol{I})\big)\boldsymbol{\lambda}^{(0)}. \tag{3.20}$$

Therefore, $\boldsymbol{\mu}(t) = \boldsymbol{\Psi}(t)\boldsymbol{\lambda}^{(0)}$ with $\boldsymbol{\Psi}(t) = \big(\boldsymbol{I} + \boldsymbol{A}(\boldsymbol{A}-\omega\boldsymbol{I})^{-1}(e^{(\boldsymbol{A}-\omega\boldsymbol{I})t} - \boldsymbol{I})\big)$.

□

Theorem 4 provides us a linear relation between exogenous event intensity and the expected overall intensity at any point in time but not just stationary intensity. The significance of this result is that it allows us later to design a diverse range of convex programs to determine the intensity of the exogenous event in order to achieve a target intensity.

In fact, we can recover the previous results in the stationary case as a special case of our general result. More specifically, a multivariate Hawkes process is stationary if the spectral radius

$$\boldsymbol{\Gamma} := \left(\int_0^{\infty} g(t)\, dt\right)\left(a_{uu'}\right)_{u,u'\in[m]} = \frac{\boldsymbol{A}}{\omega} \tag{3.21}$$

is strictly smaller than 1 [133]. In this case, the expected intensity is $\boldsymbol{\mu} = (\boldsymbol{I} - \boldsymbol{\Gamma})^{-1}\boldsymbol{\lambda}^{(0)}$ independent of the time. We can obtain this relation from theorem 4 if we let $t \to \infty$.

**Corollary 5.** $\boldsymbol{\mu} = (\mathbf{I} - \boldsymbol{\Gamma})^{-1}\boldsymbol{\lambda}^{(0)} = \lim_{t\to\infty} \boldsymbol{\Psi}(t)\,\boldsymbol{\lambda}^{(0)}$.

*Proof.* If the process is stationary, the spectral radius of $\boldsymbol{\Gamma} = \frac{\boldsymbol{A}}{w}$ is smaller than 1, which implies that all eigenvalues of $\boldsymbol{A}$ are smaller than $\omega$ in magnitude. Thus, all eigenvalues of $\boldsymbol{A} - \omega\boldsymbol{I}$ are negative. Let $\boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}$ be the eigenvalue decomposition of $\boldsymbol{A} - \omega\boldsymbol{I}$, and all the elements (in diagonal) of $\boldsymbol{D}$ are negative. Then based on the property of matrix

exponential, we have $e^{(\boldsymbol{A}-\omega \boldsymbol{I})t} = \boldsymbol{P} e^{\boldsymbol{D}t} \boldsymbol{P}^{-1}$. As we let $t \to \infty$, the matrix $e^{\boldsymbol{D}t} \to \mathbf{0}$ and hence $e^{(\boldsymbol{A}-\omega \boldsymbol{I})t} \to \mathbf{0}$. Thus $\lim_{t\to\infty} \boldsymbol{\Psi}(t) = \left(\boldsymbol{I} + \boldsymbol{A}(\boldsymbol{A} - \omega \boldsymbol{I})^{-1}\right)$ which is equal to $(\boldsymbol{I} - \boldsymbol{\Gamma})^{-1}$, and completes the proof. □

### 3.4 Convex Activity Shaping Framework

Given the linear relation between exogenous event intensity and expected overall event intensity, we now propose a convex optimization framework for a variety of activity shaping tasks. In all tasks discussed below, we will optimize the exogenous event intensity $\boldsymbol{\lambda}^{(0)}$ such that the expected overall event intensity $\boldsymbol{\mu}(t)$ is maximized with respect to some concave utility $U(\cdot)$ in $\boldsymbol{\mu}(t)$, *i.e.*,

$$\begin{aligned} &\text{maximize}_{\boldsymbol{\mu}(t),\boldsymbol{\lambda}^{(0)}} \quad U(\boldsymbol{\mu}(t)) \\ &\text{subject to} \qquad\qquad \boldsymbol{\mu}(t) = \boldsymbol{\Psi}(t)\boldsymbol{\lambda}^{(0)}, \quad \boldsymbol{c}^\top \boldsymbol{\lambda}^{(0)} \leqslant C, \quad \boldsymbol{\lambda}^{(0)} \geqslant 0 \end{aligned} \tag{3.22}$$

where $\boldsymbol{c} = (c_1, \ldots, c_m)^\top \geqslant 0$ is the cost per unit event for each user and $C$ is the total budget. Additional regularization can also be added to $\boldsymbol{\lambda}^{(0)}$ either to restrict the number of incentivized users (with $\ell_0$ norm $\|\boldsymbol{\lambda}^{(0)}\|_0$), or to promote a sparse solution (with $\ell_1$ norm $\|\boldsymbol{\lambda}^{(0)}\|_1$, or to obtain a smooth solution (with $\ell_2$ regularization $\|\boldsymbol{\lambda}^{(0)}\|_2$). We next discuss several instances of the general framework which achieve different goals (their constraints remain the same and hence omitted).

**Capped Activity Maximization.** In real networks, there is an upper bound (or a cap) on the activity each user can generate due to limited attention of a user. For example, a Twitter user typically posts a limited number of shortened urls or retweets a limited number of tweets [71]. Suppose we know the upper bound, $\alpha_u$, on a user's activity, *i.e.*, how much activity each user is willing to generate. Then we can perform the following *capped activity maximization* task

$$\text{maximize}_{\boldsymbol{\mu}(t),\boldsymbol{\lambda}^{(0)}} \quad \textstyle\sum_{u\in[m]} \min\left\{\mu_u(t), \alpha_u\right\} \tag{3.23}$$

**Minimax Activity Shaping.** Suppose our goal is instead maintaining the activity of each user in the network above a certain minimum level, or, alternatively make the user with the minimum activity as active as possible. Then, we can perform the following *minimax activity shaping* task

$$\text{maximize}_{\boldsymbol{\mu}(t),\boldsymbol{\lambda}^{(0)}} \quad \min_u \ \mu_u(t) \tag{3.24}$$

**Least-Squares Activity Shaping.** Sometimes we want to achieve a pre-specified target activity levels, $\boldsymbol{v}$, for users. For example, we may like to divide users into groups and desire a different level of activity in each group. Inspired by these examples, we can perform the following *least-squares activity shaping* task

$$\text{maximize}_{\boldsymbol{\mu}(t),\boldsymbol{\lambda}^{(0)}} \quad -\|\boldsymbol{B}\boldsymbol{\mu}(t) - \boldsymbol{v}\|_2^2 \tag{3.25}$$

where $\boldsymbol{B}$ encodes potentially additional constraints (*e.g.*, group partitions). Besides Euclidean distance, the family of Bregman divergences can be used to measure the difference between $\boldsymbol{B}\boldsymbol{\mu}(t)$ and $\boldsymbol{v}$ here. That is, given a function $f(\cdot) : \mathbb{R}^m \mapsto \mathbb{R}$ convex in its argument, we can use $D(\boldsymbol{B}\boldsymbol{\mu}(t)\|\boldsymbol{v}) := f(\boldsymbol{B}\boldsymbol{\mu}(t)) - f(\boldsymbol{v}) - \left\langle \nabla f(\boldsymbol{v}), \boldsymbol{B}\boldsymbol{\mu}(t) - \boldsymbol{v} \right\rangle$ as our objective function.

**Activity Homogenization.** Many other concave utility functions can be used. For example, we may want to steer users activities to a more homogeneous profile. If we measure homogeneity of activity with Shannon entropy, then we can perform the following activity homogenization task

$$\text{maximize}_{\boldsymbol{\mu}(t),\boldsymbol{\lambda}^{(0)}} \quad -\sum_{u\in[m]} \mu_u(t) \ln \mu_u(t) \tag{3.26}$$

### 3.5 Scalable Algorithm

All the activity shaping problems defined above require an efficient evaluation of the instantaneous average intensity $\boldsymbol{\mu}(t)$ at time $t$, which entails computing matrix exponentials to obtain $\boldsymbol{\Psi}(t)$. In small or medium networks, we can rely on well-known numerical meth-

ods to compute matrix exponentials [69]. However, in large networks with sparse graph structure $\boldsymbol{A}$, the explicit computation of $\boldsymbol{\Psi}(t)$ quickly becomes intractable.

Fortunately, we can exploit the following key property of our convex activity shaping framework: the instantaneous average intensity only depends on $\boldsymbol{\Psi}(t)$ through matrix-vector product operations. In particular, we start by using Theorem 4 to rewrite the multiplication of $\boldsymbol{\Psi}(t)$ and a vector $\boldsymbol{v}$ as $\boldsymbol{\Psi}(t)\boldsymbol{v}$. We then get a tractable solution by having $e^{(\boldsymbol{A}-\omega \boldsymbol{I})t}\boldsymbol{v}$ and a sparse linear system of equations, $(\boldsymbol{A} - \omega \boldsymbol{I})x = y$, for some $x$ and $y$. Next, we elaborate on two very efficient algorithms for computing the product of matrix exponential with a vector and for solving a sparse linear system of equations.

For the computation of the product of matrix exponential with a vector, we rely on the iterative algorithm by Al-Mohy et al. [4], which combines a scaling and squaring method with a truncated Taylor series approximation to the matrix exponential.

For solving the sparse linear system of equation, we use the well-known GMRES method [169], which is an Arnoldi process for constructing an $l_2$-orthogonal basis of Krylov subspaces. The method solves the linear system by iteratively minimizing the norm of the residual vector over a Krylov subspace. In detail, consider the $n^{th}$ Krylov subspace for the problem $\boldsymbol{C}\boldsymbol{x} = \boldsymbol{b}$ as $\boldsymbol{K}_n = \operatorname{span}\{\boldsymbol{b}, \boldsymbol{C}\boldsymbol{b}, \boldsymbol{C}^2\boldsymbol{b}, \ldots, \boldsymbol{C}^{n-1}\boldsymbol{b}\}$. GMRES approximates the exact solution of $\boldsymbol{C}\boldsymbol{x} = \boldsymbol{b}$ by the vector $\boldsymbol{x}_n \in \boldsymbol{K}_n$ that minimizes the Euclidean norm of the residual $\boldsymbol{r}_n = \boldsymbol{C}\boldsymbol{x}_n - \boldsymbol{b}$. Because the span consists of orthogonal vectors, the Arnoldi iteration is used to find an alternative basis composing rows of $\boldsymbol{Q}_n$. Hence, the vector $\boldsymbol{x}_n \in \boldsymbol{K}_n$ can be written as $\boldsymbol{x}_n = \boldsymbol{Q}_n\boldsymbol{y}_n$ with $\boldsymbol{y}_n \in \mathbb{R}^n$. Then, $\boldsymbol{y}_n$ can be found by minimizing the Euclidean norm of the residual $\boldsymbol{r}_n = \tilde{\boldsymbol{H}}_n\boldsymbol{y}_n - \beta\boldsymbol{e}_1$, where $\tilde{\boldsymbol{H}}_n$ is the Hessenberg matrix produced in the Arnoldi process, $\boldsymbol{e}_1 = (1, 0, 0, \ldots, 0)^T$ is the first vector in the standard basis of $\mathbb{R}^{n+1}$, and $\beta = \|\boldsymbol{b} - \boldsymbol{C}\boldsymbol{x}_0\|$. Finally, $\boldsymbol{x}_n$ is computed as $\boldsymbol{x}_n = \boldsymbol{Q}_n\boldsymbol{y}_n$. The whole procedure is repeated until reaching a small enough residual.

Perhaps surprisingly, we will now show that it is possible to compute the gradient of the objective functions of all our activity shaping problems using the algorithm de-

veloped above for computing the average instantaneous intensity. We only need to define the vector $\boldsymbol{v}$ appropriately for each problem, as follows: *(i)* Activity maximization: $\boldsymbol{g}(\boldsymbol{\lambda}^{(0)}) = \boldsymbol{\Psi}(t)^\top \boldsymbol{v}$, where $\boldsymbol{v}$ is defined such that $v_j = 1$ if $\alpha_j > \mu_j$, and $v_j = 0$, otherwise. *(ii)* Minimax activity shaping: $\boldsymbol{g}(\boldsymbol{\lambda}^{(0)}) = \boldsymbol{\Psi}(t)^\top \boldsymbol{e}$, where $\boldsymbol{e}$ is defined such that $e_j = 1$ if $\mu_j = \mu_{min}$, and $e_j = 0$, otherwise. *(iii)* Least-squares activity shaping: $\boldsymbol{g}(\boldsymbol{\lambda}^{(0)}) = 2\boldsymbol{\Psi}(t)^\top \boldsymbol{B}^\top \left( \boldsymbol{B}\boldsymbol{\Psi}(t)\boldsymbol{\lambda}^{(0)} - \boldsymbol{v} \right)$. *(iv)* Activity homogenization: $\boldsymbol{g}(\boldsymbol{\lambda}^{(0)}) = \boldsymbol{\Psi}(t)^\top \ln\left(\boldsymbol{\Psi}(t)\boldsymbol{\lambda}^{(0)}\right) + \boldsymbol{\Psi}(t)^\top \mathbf{1}$, where $\ln(\cdot)$ on a vector is the element-wise natural logarithm. The activity maximization and the minimax activity shaping tasks require only one evaluation of $\boldsymbol{\Psi}(t)$ times a vector. However, computing the gradient for least-squares activity shaping and activity homogenization is slightly more involved and it requires to be careful with the order in which we perform the operations.

For the gradient in the least-squares activity shaping task $\boldsymbol{B}$ is usually sparse and it includes two multiplications of a sparse matrix and a vector, two matrix exponentials multiplied by a vector, and two sparse linear systems of equations. The gradient computation in the activity homogenization task consists of two multiplication of a matrix exponential and a vector and two sparse linear systems of equations.

Equipped with an efficient way to compute of gradients, we solve the corresponding convex optimization problem for each activity shaping problem by applying the projected gradient descent [24] optimization framework with the appropriate gradient[2].

## 3.6 Experimental Evaluation

We evaluate our activity shaping framework using both simulated and real world held-out data, and show that our approach significantly outperforms several baselines [3].

### 3.6.1 Experimental Setup

Here, we briefly present our data, evaluation schemas, and settings.

[2]For nondifferential objectives, subgradient algorithms can be used instead.

[3]The code is available at https://www.cc.gatech.edu/ mfarajta/resources/activity-shaping-code.zip

Table 3.1: Number of adopters and usages for each URL shortening service.

| Service | # adopters | # usages |
|---|---|---|
| Bitly | 55,883 | 5,046,710 |
| TinyURL | 46,577 | 1,682,459 |
| Isgd | 28,050 | 596,895 |
| TwURL | 15,215 | 197,568 |
| SnURL | 4,462 | 41,823 |
| Doiop | 88 | 643 |

**Dataset description and network inference.** We use data gathered from Twitter as reported in [30], which comprises of all public tweets posted by 60,000 users during a 8-month period, from January 2009 to September 2009. For every user, we record the times she uses any of the following six url shortening services: Bitly , TinyURL, Isgd, TwURL, SnURL, Doiop. Table 3.1 shows the number of adopters and usages for the six different URL shortening services. It includes a total of 7,566,098 events (adoptions) during the 8-month period.

We evaluate the performance of our framework on a subset of 2,241 active users, linked by 4,901 edges, which we call 2K dataset, and we evaluate its scalability on the overall 60,000 users, linked by $\sim$ 200,000 edges, which we call 60K dataset. The 2K dataset accounts for 691,020 url shortened service uses while the 60K dataset accounts for $\sim$7.5 million uses. Finally, we treat each service as independent cascades of events.

In the experiments, we estimated the nonnegative influence matrix $\boldsymbol{A}$ and the exogenous intensity $\boldsymbol{\lambda}^{(0)}$ using maximum log-likelihood, as in previous work [229, 228, 195]. We used a temporal resolution of one minute and selected the bandwidth $\omega = 0.1$ by cross validation. Loosely speaking, $\omega = 0.1$ corresponds to loosing 70% of the initial influence after 10 minutes, which may be explained by the rapid rate at which each user' news feed gets updated.

**Evaluation schemes.** We focus on three tasks: capped activity maximization, minimax activity shaping, and least square activity shaping. We set the total budget to $C = 0.5$, which corresponds to supporting a total extra activity equal to $0.5$ actions per unit time,

and assume all users entail the same cost. In the capped activity maximization, we set the upper limit of each user's intensity, $\boldsymbol{\alpha}$, by adding a nonnegative random vector to their inferred initial intensity. In the least-squares activity shaping, we set $\boldsymbol{B} = \mathbf{I}$ and aim to create three groups of users, namely less-active, moderate, and super-active users. We use three different evaluation schemes, with an increasing resemblance to a real world scenario:

*Theoretical objective*: We compute the expected overall (theoretical) intensity by applying Theorem 4 on the optimal exogenous event intensities, $\boldsymbol{\lambda}_{opt}^{(0)}$, to each of the three activity shaping tasks, as well as the learned $\boldsymbol{A}$ and $\omega$. We then compute and report the value of the objective functions.

*Simulated objective*: We simulate $50$ cascades with Ogata's thinning algorithm [148], using the optimal exogenous event intensities, $\boldsymbol{\lambda}_{opt}^{(0)}$, to each of the three activity shaping tasks, and the learned $\boldsymbol{A}$ and $\omega$. We then estimate empirically the overall event intensity based on the simulated cascades, by computing a running average over non-overlapping time windows, and report the value of the objective functions based on this estimated overall intensity.

*Held-out data*: The most interesting evaluation scheme would entail carrying out real interventions in a social platform. However, since this is very challenging to do, instead, in this evaluation scheme, we use held-out data to simulate such process, proceeding as follows. We first partition the $8$-month data into $50$ five-day long contiguous intervals. Then, we use one interval for training and the remaining $49$ intervals for testing. Suppose interval $1$ is used for training, the procedure is as follows:

1. We estimate $\boldsymbol{A}_1$, $\omega_1$ and $\boldsymbol{\lambda}_1^{(0)}$ using the events from interval $1$. Then, we fix $\boldsymbol{A}_1$ and $\omega_1$, and estimate $\boldsymbol{\lambda}_i^{(0)}$ for all other intervals, $i = 2, \ldots, 49$.
2. Given $\boldsymbol{A}_1$ and $\omega_1$, we find the optimal exogenous event intensities, $\boldsymbol{\lambda}_{opt}^{(0)}$, for each of the three activity shaping task, by solving the associated convex program. We then sort the estimated $\boldsymbol{\lambda}_i^{(0)}$ ($i = 2, \ldots, 49$) according to their similarity to $\boldsymbol{\lambda}_{opt}^{(0)}$, using the Euclidean distance $\|\boldsymbol{\lambda}_{opt}^{(0)} - \boldsymbol{\lambda}_i^{(0)}\|_2$.

3. We estimate the overall event intensity for each of the $49$ intervals ($i = 2, \ldots, 49$), as in the "simulated objective" evaluation scheme, and sort these intervals according to the value of their corresponding objective function.
4. Last, we compute and report the rank correlation score between the two orderings obtained in step 2 and 3.[4] The larger the rank correlation, the better the method.

We repeat this procedure 50 times, choosing each different interval for training once, and compute and report the average rank correlations.

It is beneficial to emphasize that the held-out experiments are essentially evaluating prediction performance on test sets. For instance, suppose we are given a diffusion network and two different configuration of incentives. We will shortly show our method can predict more accurately which one will reach the activity shaping goal better. This means, in turn, that if we incentivize the users according to our method's suggestion, we will achieve the target activity better than other heuristics.

Alternatively, one can understand our evaluation scheme like this: if one applies the incentive (or intervention) levels prescribed by a method, how well the predicted outcome coincides with the reality in the test set? A good method should behavior like this: the closer the prescribed incentive (or intervention) levels to the estimated base intensities in test data, the closer the prediction based on training data to the activity level in the test data. In our experiment, the closeness in incentive level is measured by the Euclidean distance, the closeness between prediction and reality is measured by rank correlation.

In the following, we describe the considered baselines proposed to compare to our approach for i) the capped activity maximization; ii) the minimax activity shaping; and iii) the least-squares activity shaping problems.

For ***capped activity maximization*** problem, we consider the following four heuristic baselines:

- XMU allocates the budget based on users' current activity. In particular, it assigns the

[4] rank correlation = number of pairs with consistent ordering / total number of pairs.

budget to each of the half top-most active users proportional to their average activity, $\boldsymbol{\mu}(t)$, computed from the inferred parameters.

- WEI assigns positive budget to the users proportionally to their sum of out-going influence ($\sum_u a_{uu'}$). This heuristic allows us (by comparing its results to CAM) to understand the effect of considering the whole network with respect to only consider the direct (out-going) influence.

- DEG assumes that more central users, *i.e.*, more connected users, can leverage the total activity, therefore, assigns the budget to the more connected users proportional to their degree in the network.

- PRK sorts the users according to the their pagerank in the weighted influence network ($\boldsymbol{A}$) with the damping factor set to $0.85\%$, and assigns the budget to the top users proportional their pagerank value.

In order to show how network structure leverages the ***minimax activity shaping*** we implement following four baselines:

- UNI allocates the total budget equally to all users.

- MINMU divides uniformly the total budget among half of the users with lower average activity $\boldsymbol{\mu}(t)$, which is computed from the inferred parameters.

- LP finds the top half of least-active users in the current network and allocates the budget such that after the assignment the network has the highest minimum activity possible. This method uses linear programming to learn exogenous activity of the users, but, in contrast to the proposed method, does not consider the network and propagation of adoptions.

- GRD finds the user with minimum activity, assigns a portion of the budget, and computes the resulting $\boldsymbol{\mu}(t)$. It then repeats the process to incentivize half of users.

We compare ***least-square activity shaping*** with the following baselines:

- PROP shapes the activity by allocating the budget proportional to the desired shape, *i.e.*, the shape of the assignment is similar to the target shape.
- LSGRD greedily finds the user with the highest distance between her current and target activity, assigns her a budget to reach her target, and proceeds this way to consume the whole budget.

Each baseline relies on a specific property to allocate the budget (*e.g.* connectedness in DEG). However, most of them face two problems: The first one is how many users to incentivize and the second one is how much should be paid to the selected users. They usually rely on heuristics to reveal these two problems (*e.g.* allocating an amount proportional to that property and/or to the top half users sorted based on the specific property). In contrast, our framework is comprehensive enough to address those difficulties based on well-developed theoretical basis. This key factor accompanied with the appropriate properties of Hawkes process for modeling social influence (*e.g.* mutually exciting) make the proposed method the best.

### 3.6.2 Temporal Properties

For the experiments on simulated objective function and held-out data we have estimated intensity from the events data. In this section, we will see how this empirical intensity resembles the theoretical intensity. We generate a synthetic network over 100 users. For each user in the generated network, we uniformly sample from $[0, 0.1]$ the exogenous intensity, and the endogenous parameters $a_{uu'}$ are uniformly sampled from $[0, 0.1]$. A bandwidth $\omega = 1$ is used in the exponential kernel. Then, the intensity is estimated empirically by dividing the number of events by the length of the respective interval.

We compute the mean and variance of the empirical activity for 100 independent runs. As illustrated in Figure 3.2, the average empirical intensity (the blue curve) clearly follows

the theoretical instantaneous intensity (the red curve) but, as expected, as we are further from the starting point (*i.e.*, as time increases), the standard deviation of the estimates (shown in the whiskers) increases. Additionally, the green line shows the average stationary intensity. As it is expected, the instantaneous intensity tends to the stationary value when the network has been run for sufficient long time.

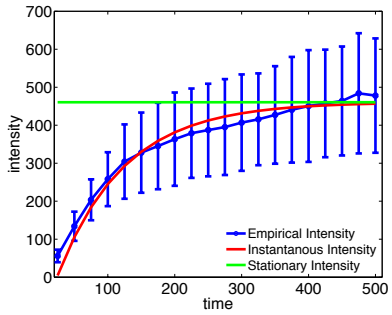
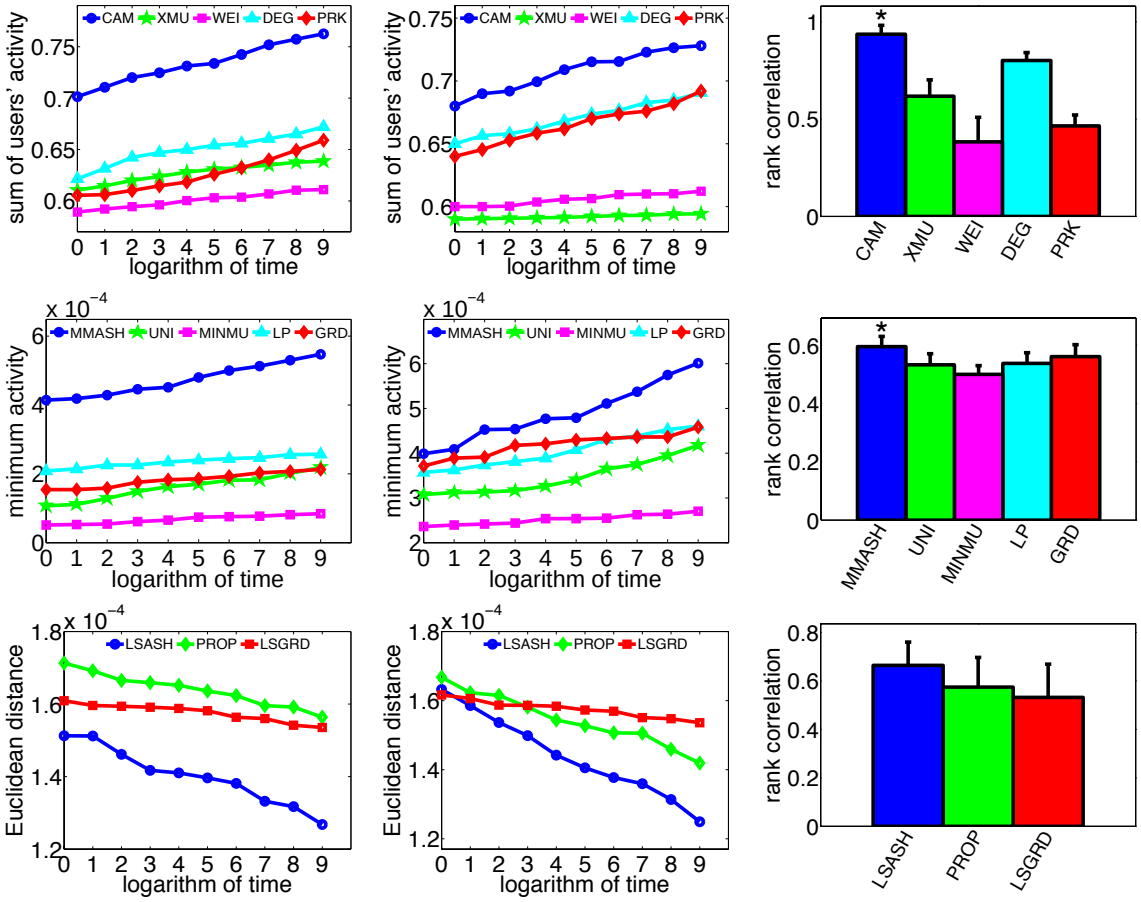



Figure 3.2: Evolution in time of empirical and theoretical intensity.

### 3.6.3 Activity Shaping Results

In this section, the results for three activity shaping tasks evaluated on the three schemas are presented.

**Capped activity maximization (CAM).** The first row of Figure 3.3 summarizes the results for the three different evaluation schemes. We find that our method (CAM) consistently outperforms the alternatives. For the theoretical objective, CAM is 11 % better than the second best, DEG. The difference in overall users' intensity from DEG is about $0.8$ which, roughly speaking, leads to at least an increase of about $0.8 \times 60 \times 24 \times 30 = 34,560$ in the overall number of events in a month. In terms of simulated objective and held-out data, the results are similar and provide empirical evidence that, compared to other heuristics, degree is an appropriate surrogate for influence, while, based on the poor performance of XMU, it seems that high activity does not necessarily entail being influential. To elaborate on the interpretability of the real-world experiment on held-out data, consider for example the difference in rank correlation between CAM and DEG, which is almost $0.1$.




(a) Theoretical objective (b) Simulated objective (c) Held-out data

Figure 3.3: Row 1: Capped activity maximization. Row 2: Minimax activity shaping. Row 3: Least-squares activity shaping. * means statistical significant at level of 0.01 with paired t-test between our method and the second best

Then, roughly speaking, this means that incentivizing users based on our approach accommodates with the ordering of real activity patterns in $0.1 \times \frac{50 \times 49}{2} = 122.5$ more pairs of realizations.

**Minimax activity shaping (MMASH).** We compare to a number of alternatives. UNI: heuristic based on equal allocation; MINMU: heuristic based on $\boldsymbol{\mu}(t)$ without optimization; LP: linear programming based heuristic; GRD: a greedy approach to leverage the activity. The second row of Figure 3.3 summarizes the results for the three different evaluation schemes. We find that our method (MMASH) consistently outperforms the alternatives. For the theoretical objective, it is about $2\times$ better than the second best, LP. Importantly,

the difference between MMASH and LP is not trifling and the least active user carries out $2 \times 10^{-4} \times 60 \times 24 \times 30 = 4.3$ more actions in average over a month. As one may have expected, GRD and LP are the best among the heuristics. The poor performance of MINMU, which is directly related to the objective of MMASH, may be because it assigns the budget to a low active user, regardless of their influence. However, our method, by cleverly distributing the budget to the users whom actions trigger many other users' actions (like those ones with low activity), it benefits from the budget most. In terms of simulated objective and held-out data, the algorithms' performance become more similar.

**Least-squares activity shaping (LSASH).** We compare to two alternatives. PROP: Assigning the budget proportionally to the desired activity; LSGRD: greedily allocating budget according the difference between current and desired activity. The third row of Figure 3.3 summarizes the results for the three different evaluation schemes. We find that our method (LSASH) consistently outperforms the alternatives. Perhaps surprisingly, PROP, despite its simplicity, seems to perform slightly better than LSGRD. This is may be due to the way it allocates the budget to users, *e.g.*, it does not aim to strictly fulfill users' target activity but benefit more users by assigning budget proportionally.

In all three tasks, longer times lead to larger differences between our method and the alternatives. This occurs because the longer the time, the more endogenous activity is triggered by network influence, and thus our framework, which models both endogenous and exogenous events, becomes more suitable.

### 3.6.4 Sparsity and Activity Shaping

In some applications there is a limitation on the number of users we can incentivize. In our proposed framework, we can handle this requirement by including a sparsity constraint on the optimization problem. In order to maintain the convexity of the optimization problem, we consider a $l_1$ regularization term, where a regularization parameter $\gamma$ provides the trade-

off between sparsity and the activity shaping goal:

$$\begin{aligned} &\text{maximize}_{\boldsymbol{\mu}(t),\boldsymbol{\lambda}^{(0)}} \quad U(\boldsymbol{\mu}(t)) - \gamma||\boldsymbol{\lambda}^{(0)}||_1 \\ &\text{subject to} \qquad\qquad \boldsymbol{\mu}(t) = \boldsymbol{\Psi}(t)\boldsymbol{\lambda}^{(0)}, \quad \boldsymbol{c}^\top \boldsymbol{\lambda}^{(0)} \leqslant C, \quad \boldsymbol{\lambda}^{(0)} \geqslant 0 \end{aligned} \tag{3.27}$$

Tables 3.2 and 3.3 demonstrate the effect of different values of regularization parameter on *capped activity maximization* and *minimax activity shaping*, respectively. When $\gamma$ is small, the minimum intensity is very high. On the contrary, large values of $\gamma$ imposes large penalties on the number of non-zero intensities which results in a sparse and applicable manipulation. Furthermore, this may avoid using all the budget. When dealing with unfamiliar application domains, cross validation may help to find an appropriate trade-off between sparsity and objective function.

Table 3.2: Sparsity properties of capped activity maximization.

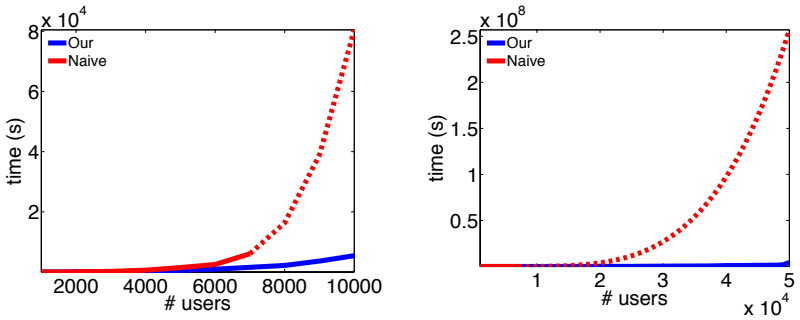
| $\gamma$ | # Non-zeros | Budget consumed | Sum of activities |
|---|---|---|---|
| 0.5 | 2101 | 0.5 | 0.69 |
| 0.6 | 1896 | 0.46 | 0.65 |
| 0.7 | 1595 | 0.39 | 0.62 |
| 0.8 | 951 | 0.21 | 0.58 |
| 0.9 | 410 | 0.18 | 0.55 |
| 1.0 | 137 | 0.13 | 0.54 |

Table 3.3: Sparsity properties of minimax activity shaping.

| $\gamma(\times 10^{-3})$ | # Non-zeros | Budget Consumed | $u_{min}(\times 10^{-3})$ |
|---|---|---|---|
| 0.6 | 1941 | 0.49 | 0.38 |
| 0.7 | 881 | 0.17 | 0.22 |
| 0.8 | 783 | 0.15 | 0.21 |
| 0.9 | 349 | 0.09 | 0.16 |
| 1.0 | 139 | 0.06 | 0.12 |
| 1.1 | 102 | 0.04 | 0.11 |

### 3.6.5 Scalability

The most computationally demanding part of the proposed algorithm is the evaluation of matrix exponentials, which we scale up by utilizing techniques from matrix algebra, such as GMRES and Al-Mohy methods. As a result, we are able to run our methods in a reasonable




(a) For 10,000 users. (b) For 50,000 users.

Figure 3.4: Scalability of least-squares activity shaping.

amount of time on the 60K dataset, specifically, in comparison with a naive implementation of matrix exponential evaluations. The naive implementation of the algorithm requires computing the matrix exponential once, and using it in (non-sparse huge) matrix-vector multiplications, *i.e.*,

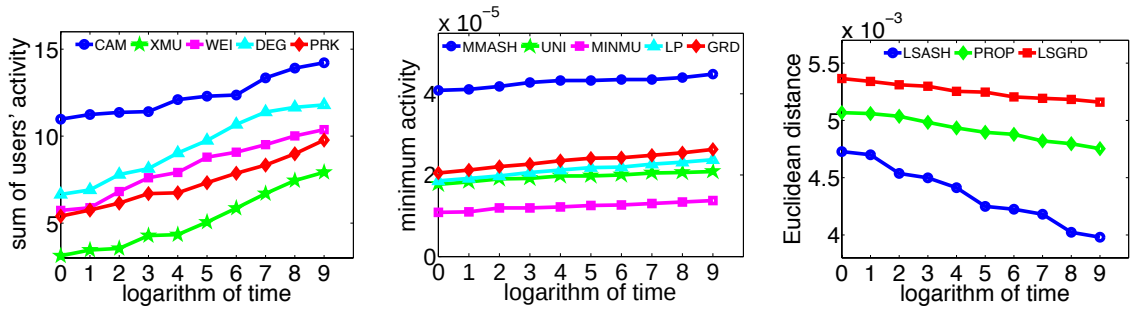
$$T_{naive} = T_{\mathbf{\Psi}} + kT_{prod}.$$

Here, $T_{\mathbf{\Psi}}$ is the time to compute $\mathbf{\Psi}(t)$, which itself comprised of three parts; matrix exponential computation, matrix inversion and matrix multiplications. $T_{prod}$ is the time for multiplication between the large non-sparse matrix and a vector plus the time to compute the inversion via solving linear systems of equation. Finally, $k$ is the number of gradient computations, or more generally, the number of iterations in any gradient-based iterative optimization. The dominant factor in the naive approach is the matrix exponential. It is computationally demanding and practically inefficient for more than 7000 users.

In contrast, the proposed framework benefits from the fact that the gradient depends on $\mathbf{\Psi}(t)$ only through matrix-vector products. Thus, the running time of our activity shaping framework will be written as

$$T_{our} = kT_{grad},$$

where $T_{grad}$ is the time to compute the gradient which itself comprises the time required to solve a couple of linear systems of equations and the time to compute a couple of exponential matrix-vector multiplication.




(a) Capped activity maximization. (b) Minimax activity shaping. (c) Least-square activity shaping.

Figure 3.5: Activity shaping on the 60K dataset.

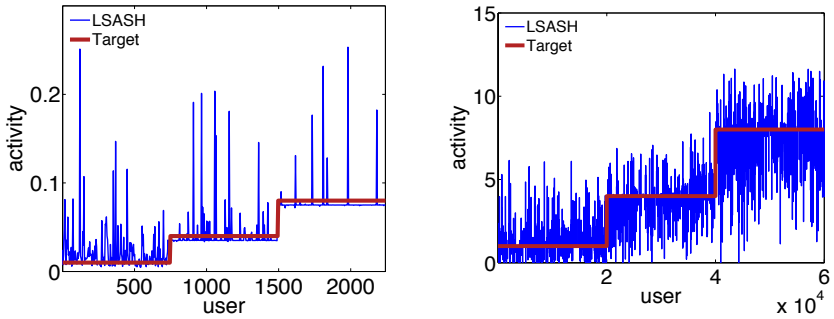
Figure 3.4 demonstrates $T_{our}$ and $T_{naive}$ with respect to the number of users. For better visualization we have provided two graphs for up to 10,000 and 50,000 users, respectively. We set $k$ equal to the number of users. Since the dominant factor in the naive computation method is matrix exponential, the choice of $k$ is not that determinant. The time for computing matrix exponential is interpolated for more than 7000 users; and the interpolated total time, $T_{naive}$, is shown in red dashed line. These experiments are done in a machine equipped with one 2.5 GHz AMD Opteron Processor. This graph clearly shows the significance of designing an scalable algorithm.

Figure 3.5 shows the results of running our large-scale algorithm on the 60K dataset evaluated via theoretical objective function. We observe the same patterns as 2K dataset. Especially, the proposed method consistently outperforms the heuristic baselines. Heuristic baselines provide similar performance as for the 2K dataset. DEG shows up again as a reasonable surrogate for influence, and the poor performance of XMU on activity maximization shows that high activity does not necessarily mean being more influential. For *minimax activity shaping* we observe MMASH is superior to others in $2 \times 10^{-5}$ actions per unit time, which means that the person with minimum activity uses the service $2 \times 10^{-5} \times 60 * 24 * 30 = 0.864$ times more compared to the best heuristic baseline. An increase in the activity per month of $0.864$ is not a big deal itself, however, if we consider the scale at which the network's activity is steered, we can deduce that now the service is guaranteeing, at least in theory, about $60000 \times 0.864 = 51840$ more adoptions monthly. As

shown by the experiments on real-world held-out data, our approach for activity shaping outperforms all the considered heuristic baselines.

### 3.6.6 Visualization of Least-squares Activity Shaping

To get a better insight on the the activity shaping problem we visualize the *least-squares activity shaping* results for the 2K and 60K datasets. Figure 3.6 shows the result of activity shaping at $t = 1$ targeting the same shape as in the experiments section. The red line is the target shape of the activity and the blue curve correspond to the activity profiles of users after incentivizing computed via theoretical objective. It is clear that the resulted activity behavior resembles the target shape.




(a) Capped activity maximization. (b) Minimax activity shaping.

Figure 3.6: Activity shaping results.

## 3.7 Summary and Conclusions

In this chapter, based on two categories of social events, exogenous and endogenous, we propose a framework for activity shaping. More especially, we derive an instantaneous activity measure which can be utilized in general networks to fulfill queries regarding to the activity at a particular time. Fortunately, the instantaneous activity linearly depends on exogenous activity which helps us build an intuitive activity shaping framework by incentivizing users to act more. Several convex instantiations of activity shaping problem is proposed along with an scalable algorithm to solve them. By running experiments on a

twitter dataset consisting of 60000 users we have shown that our model can shape activities to our desire better than some heuristics. It suggests that considering network structure will help shape the activity better. For future work, one can study the effect of manipulating the links between users on the their behavior. Exploring other possible kernel functions or learning it using non-parametric methods is an interesting problem. We have used point processes framework in intervention in other contexts as well. For example in [109] we find the best time to post in social networks in order to maximize the visibility of a post to followers. Furthermore, in [88] we tackle invasive species management problem in which a few cells of landscape are going to be chosen for complete removal of invasive plant.

# CHAPTER 4

# MULTI STAGE OPTIMIZATION IN POINT PROCESSES

We establish theoretical foundations of optimal campaigning over social networks where the user activities are modeled as a multivariate Hawkes process, and we derive a time dependent linear relation between the intensity of exogenous events and several commonly used objective functions of campaigning. We further develop a convex dynamic programming framework for determining the optimal intervention policy that prescribes the required level of external drive at each stage for the desired campaigning result. The dynamic programming framework is employed to balance the high present reward and large penalty on low future outcome in the presence of extensive uncertainties. We utilize the developed framework in optimizing campaigns over social networks. Experiments on both synthetic data and the real-world MemeTracker dataset show that our algorithm can steer the user activities for optimal campaigning much more accurately than baselines.

## 4.1 Introduction

The key factor differentiating social networks from traditional media is *peer influence*. In fact, events in an online social network can be categorized roughly into two types: endogenous events where users just respond to the actions of their neighbors within the network, and exogenous events where users take actions due to drives external to the network. Then it is natural to raise the following fundamental questions regarding optimal campaigning over social networks: can we model and exploit those event data to steer the online community to a desired exposure level? More specifically, can we drive the overall exposure to a campaign to a certain level (e.g., at least twice per week per user) by incentivizing a small number of users to take more initiatives? What about maximizing the overall exposure for a target group of people?

More importantly, those exposure shaping tasks are more effective when the interventions are implemented in multiple stages. Due to the inherent uncertainty in social behavior, the outcome of each intervention may not be fully predictable but can be anticipated to some extent before the next intervention happens. A key aspect of such situations is that interventions can't be viewed in isolation since one must balance the desire for high present reward with the penalty of low future outcome.

In this chapter, the *dynamic programming* framework [16] is employed to tackle the aforementioned issues. In particular, we first establish the fundamental theory of optimal campaigning over social networks where the user activities are modeled as a multivariate Hawkes process (MHP) [40, 228] since MHP can capture both endogenous and exogenous event intensities. We also derive a time dependent linear relation between the intensity of exogenous events and the overall exposure to the campaign. Exploiting this connection, we develop a convex dynamic programming framework for determining the optimal intervention policy that prescribes the required level of external drive at each stage in order for the campaign to reach a desired exposure profile. We propose several objective functions that are commonly considered as campaigning criteria in social networks.

We will use the proposed multi-stage intervention procedure to tackle campaigning problem. Obama was the first US president in history who successfully leveraged online social media in presidential campaigning, which has been popularized and become a ubiquitous approach to electoral politics (such as in the on-going 2016 US presidential election) in contrast to the decreasing relevance of traditional media such as TV and newspapers [209, 197, 162]. The power of campaigning via social media in modern politics is a consequence of online social networking being an important part of people's regular daily social lives. It has been quite common that individuals use social network sites to share their ideas and comment on other people's opinions. In recent years, large organizations, such as governments, public media, and business corporations, also start to announce news, spread ideas, and/or post advertisements in order to steer the public opinion through

social media platform. There has been extensive interest for these entities to influence the public's view and manipulate the trend by incentivizing influential users to endorse their ideas/merits/opinions at certain monetary expenses or credits. To obtain most cost-effective trend manipulations, one needs to design an optimal campaigning strategy or policy such that quantities of interests, such as influence of opinions, exposure of a campaign, adoption of new products, can be maximized or steered towards the target amount given realistic budget constraints.

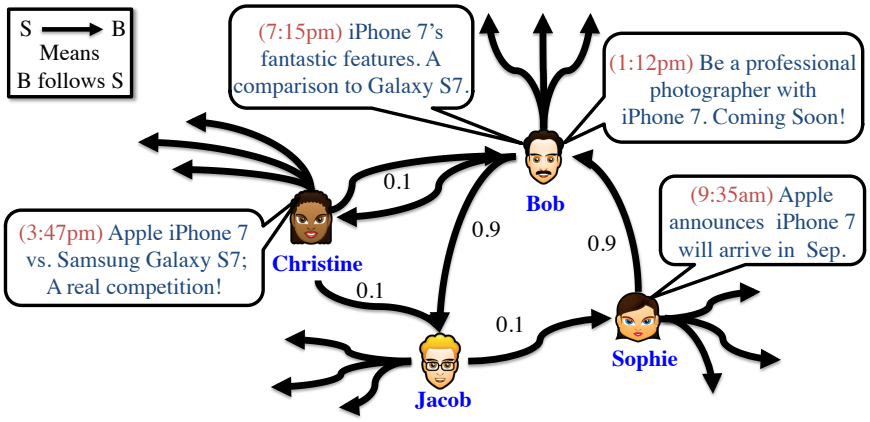
Experiments on both synthetic data and real world network of news websites in the MemeTracker dataset show that our algorithms can shape the exposure of campaigns much more accurately than baselines.

## 4.2 Illustrative Example

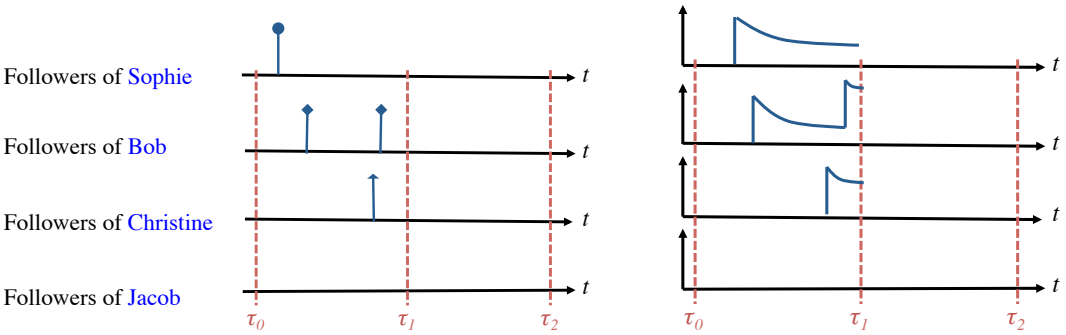
Next sections will define the campaigning problem formally. However, we found it beneficial to formulate it more intuitively. Next section will define all the concepts appeared here rigorously.

Figure 4.1-a shows a hypothetical social network highlighting 4 users and their influence on each other. They have their own set of followers which for simplicity are considered disjoint and being influenced equally. Our objective in this toy example is to maximize the minimum exposure on the followers. For the ease of exposition, we only consider 2 stages which in the beginning we can intervene. Hypothetically, we are going to advertise iPhone 7 on behalf of Apple Inc.

Now, we intuitively proceed to find the optimum intervention. At the first stage, where the state is zero (no exposure from the past), intervention through Jacob and Christine would not be part of the solution (or their share would be negligible), because the outcome would be small due to the low number of total followers and the small amount of influence. Between Bob and Sophie, we would select her. Note that Sophie has a high influence on Bob and if she is incentivized to make a blog post for the campaign, she can make Bob




a) The social network and events




b) exposure events to followers

c) intensity of exposure of followers

Figure 4.1: A social network and multi-stage campaigning to maximize the minimum exposure.

interested too. Therefore, the procedure will assign most budget to Sophie. Furthermore, Bob has a high influence on Jacob and will probably inspire him to do an activity. Let's follow a hypothetical scenario. Sophie makes her post at 9:35am about apple's new phone. Bob is notified about her post and will make an inspiring post about the phone's camera. However, Jacob will not respond to this campaign well, e.g., he may be traveling that day. This phenomenon is perfectly captured via the probabilistic framework for event analysis [1, 40]. On the other hand, Christine who is not a big fan of Bob's post compared to Jacob will respond to Bob's post by writing a comparison between Apple and Samsung's product at 3:47pm. This will reinforce Bob to make a separate post explaining the features of Apple's new product. These events are illustrated in Figure 4.1-b.

The cascade of events at the first stage will expose followers of Sophia, Bob, and Christine to the new product. However, all things do not go according to our expectation. Jacob happens to be out of his office that day and his followers are not aware of the news. Considering the MEM as the objective function, this is quite disappointing that there are people who are exposed to the campaign 0 times. Exposures of followers are demonstrated in Figure 4.1-b. This is the point where multi-stage dynamic campaigning helps. Considering the exposure intensity of followers in Figure 4.1-c, Christine's, Bob's, and even Sophia's followers are still under the influence. Furthermore, they may follow their initial posts by new ones. Then, in the next stage, it looks reasonable to invest on Jacob. At least his followers will notify about the campaign and also as Jacob has an influence on Sophie, his posts may inspire Sophie again and this may increase the exposure intensity of followers of Sophie as well. Therefore, thanks to a second chance in intervention, Jacob will get the most budget at the second stage to advertise the product.

## 4.3 Notations

Users in the social network are represented by a point process $\mathcal{N}(t) = (\mathcal{N}^1(t), \ldots, \mathcal{N}^n(t))^\top$. Here, $\mathcal{N}^i(t)$ records the number of events user $i$ performs before time $t$ for $1 \leq i \leq n$, and her intensity is:

$$\lambda^i(t)dt := \mathbb{P}\left\{\text{user } i \text{ performs event in } [t, t+dt) \mid \mathcal{H}(t)\right\} = \mathbb{E}[d\mathcal{N}^i(t) \mid \mathcal{H}(t)], \qquad (4.1)$$

where one typically assumes that only one event can happen in a small window of size $dt$. Figure 4.1-b and Figure 4.1-c show the events and intensity function of the activities of the 4 users in the social network respectively. Furthermore, these two can be seen as the exposure events and the exposure intensity function to their followers since for simplicity we are just considering the activities of these 4 users in the network. The functional form of the intensity $\lambda^i(t)$ is often designed to capture the phenomena of interests.

The Hawkes process [93] is a class of self and mutually exciting point process models,

$$\lambda^i(t) = \mu^i(t) + \sum_{k:t_k<t} \phi^{id_k}(t, t_k) = \mu^i(t) + \sum_{j=1}^{n} \int_0^t \phi^{ij}(t, s) d\mathcal{N}^j(s), \tag{4.2}$$

where the intensity is history dependent. $\phi^{ij}(t, s)$ is the impact function capturing the temporal influence of an event by user $j$ at time $s$ to the future events of user $j$ at time $t \geqslant s$. Here, the first term $\mu^i(t)$ is the exogenous event intensity modeling drive outside the network and indecent of the history, and the second term $\sum_{k:t_k<t} \phi^{id_k}(t, t_k)$ is the endogenous event intensity modeling interactions within the network [55]. Defining $\Phi(t, s) = [\phi^{ij}(t, s)]_{i,j=1\ldots n}$, and $\lambda(t) = (\lambda^1(t), \ldots, \lambda^n(t))^\top$, and $\mu(t) = (\mu^1(t), \ldots, \mu^n(t))^\top$ we can compactly rewrite Eq 4.2 in matrix form:

$$\lambda(t) = \mu(t) + \int_0^t \Phi(t, s) d\mathcal{N}(s). \tag{4.3}$$

In practice it is standard to employ shift-invariant impact function, *i.e.*, $\Phi(t, s) = \Phi(t - s)$. Then, by using notation of convolution $f(t) * g(t) = \int_0^t f(t - s)g(s)ds$ we have

$$\lambda(t) = \mu(t) + \Phi(t) * d\mathcal{N}(t). \tag{4.4}$$

## 4.4 From Intensity to Average Activity

In this section we will develop a closed form relation between the expected total intensity $\mathbb{E}[\lambda(t)]$ and the intensity $\mu(t)$ of exogenous events. This relation establish the basis of our campaigning framework. First, define the *mean function* as $\mathcal{M}(t) := \mathbb{E}[\mathcal{N}(t)] = \mathbb{E}_{\mathcal{H}(t)}[\mathbb{E}(\mathcal{N}(t)|\mathcal{H}(t))]$. Note that $\mathcal{M}(t)$ is history independent, and it gives the average number of events up to time $t$ for each of the dimension. Similarly, the *rate* function $\eta(t)$ is given by $\eta(t)dt := d\mathcal{M}(t)$. On the other hand,

$$d\mathcal{M}(t) = d\mathbb{E}[\mathcal{N}(t)] = \mathbb{E}_{\mathcal{H}(t)}[\mathbb{E}(d\mathcal{N}(t)|\mathcal{H}(t))] = \mathbb{E}_{\mathcal{H}(t)}[\lambda(t)|\mathcal{H}(t)]dt = \mathbb{E}[\lambda(t)]dt. \tag{4.5}$$

Therefore $\eta(t) = \mathbb{E}[\lambda(t)]$ which serves as a measure of activity in the network. In what follows we will find an analytical form for the average activity.

**Lemma 6.** *Suppose $\Psi : [0, T] \to \mathbb{R}^{n\times n}$ is a matrix function, then for every fixed constant*

*intensity* $\mu(t) = c \in \mathbb{R}^n_+$, $\eta_c(t) := \Psi(t)c$ *solves the semi-infinite integral equation*

$$\eta(t) = c + \int_0^t \Phi(t-s)\eta(s)ds, \quad \forall t \in [0,T], \tag{4.6}$$

*if and only if* $\Psi(t)$ *satisfies*

$$\Psi(t) = I + \int_0^t \Phi(t-s)\Psi(s)ds, \quad \forall t \in [0,T]. \tag{4.7}$$

*In particular, if* $\Phi(t) = Ae^{-\omega t}\mathbf{1}_{\geq 0}(t) = [a_{ij}e^{-\omega t}\mathbf{1}_{\geq 0}(t)]_{ij}$ *where* $0 \leq \omega \notin$ *Spectrum*$(A)$*, then*

$$\Psi(t) = I + A(A-\omega I)^{-1}(e^{(A-\omega I)t} - I) \tag{4.8}$$

*for* $t \in [0,T]$*, where,* $\mathbf{1}_{\geq 0}(t)$ *is an indicator function for* $t \geq 0$*.*

*Proof.* Suppose that $\Psi(t)c$ solves (4.6) for every $c$, then substituting $\eta(t)$ by $\eta_c(t) := \Psi(t)c$ in (4.6) we obtain $\left[\Psi(t) - I - \int_0^t \Phi(t-s)\Psi(s)ds\right] c = 0$. Since $c \in \mathbb{R}^n_+$ is arbitrary, we know that $\Psi(t) - I - \int_0^t \Phi(t-s)\Psi(s)ds = 0$ for all $t$, and hence (4.7) follows. The converse is trivial to verify. Furthermore, one can readily check that (4.8) satisfies (4.7) for $\Phi(t) = Ae^{-\omega t}\mathbf{1}_{\geq 0}(t)$. □

Let $\mu : [0,T] \to \mathbb{R}^n_+$ be a right-continuous piecewise constant function

$$\mu(t) = \sum_{m=1}^{M} c_m \mathbf{1}_{[\tau_{m-1},\tau_m)}(t), \tag{4.9}$$

where $0 = \tau_0 < \tau_1 < \cdots < \tau_M = T$ is a finite partition of time interval $[0,T]$ and function $\mathbf{1}_{[\tau_{m-1},\tau_m)}(t)$ indicates $\tau_{m-1} \leq t < \tau_m$. The next theorem shows that if $\Psi(t)$ satisfies (4.7), then one can calculate $\eta(t)$ for piecewise constant intensity $\mu : [0,T]$ of form (4.9).

**Theorem 7.** *Let* $\Psi(t)$ *satisfy* (4.7) *and* $\mu(t)$ *be a right-continuous piecewise constant intensity function of form* (4.9)*, then the rate function* $\eta(t)$ *is given by*

$$\eta(t) = \sum_{k=0}^{m} \Psi(t-\tau_k)(c_k - c_{k-1}), \tag{4.10}$$

*for all* $t \in (\tau_{m-1}, \tau_m]$ *and* $m = 1, \ldots, M$*, where* $c_{-1} := 0$ *by convention.*

*Proof.* We prove this result by induction on partition size $M$. The previous lemma shows (4.10) for constant $\mu(t) = c\mathbf{1}_{[0,T]}(t)$ (i.e., $M = 1$). Suppose (4.10) is true for any given

piecewise constant $\mu(t)$ of form (4.9) with $M$ partitions. If we impose a constant control $c \in \mathbb{R}^n_+$ (different from original $c_M$) since time $\tau \in (\tau_{M-1}, T]$, namely the piecewise constant intensity function is updated to $\hat{\mu}(t) := \mu(t) + (c - c_M)\mathbf{1}_{(\tau,T]}(t)$, then we need to show that the updated rate function $\hat{\eta}(t)$ is

$$\hat{\eta}(t) = \eta(t) + \Psi(t-\tau)(c - c_M)\mathbf{1}_{(\tau,T]}(t), \tag{4.11}$$

for all $t \in [0, T]$. This result can by verified easily for $t \in [0, \tau]$. If $t \in (\tau, T]$, then $\hat{\mu}(t) = \mu(t) + (c - c_M)\mathbf{1}_{(\tau,T]}(t) = \mu(t) + (c - c_M)$ and

$$\begin{aligned}
&\hat{\mu}(t) + \int_0^t \Phi(t-s)\hat{\eta}(s)ds \\
&= \mu(t) + (c - c_M) + \int_0^t \Phi(t-s)[\eta(s) + \Psi(s-\tau)(c - c_M)\mathbf{1}_{(\tau,T]}(s)]ds \\
&= \eta(t) + (c - c_M) + \int_0^t \Phi(t-s)\Psi(s-\tau)(c - c_M)\mathbf{1}_{(\tau,T]}(s)ds \\
&= \eta(t) + \left[I + \int_0^{t-\tau} \Phi(t-\tau-u)\Psi(u)du\right](c - c_M) \\
&= \eta(t) + \Psi(t-\tau)(c - c_M) = \hat{\eta}(t),
\end{aligned} \tag{4.12}$$

where we used the fact that $\eta(t)$ is the rate function for intensity $\mu(t)$ to get the second equality, applied change of variables $u = s - \tau$ to obtain the third equality, and the property (4.7) of $\Psi(t)$ to get the fourth equality. This implies that the rate function is $\hat{\eta}(t)$ given in (4.11) for the updated piecewise constant intensity $\hat{\mu}(t)$ with $M + 1$ partitions, and hence completes the proof. □

Using the above lemma, for the first time, we derive the average intensity for a general exogenous intensity. Section 4.7 includes a few experiments to investigate these results empirically.

**Theorem 8.** *If $\Psi \in C^1([0, T])$ and satisfies* (4.7)*, and exogenous intensity $\mu$ is bounded and piecewise absolutely continuous on $[0, T]$ where $\mu(t+) = \mu(t)$ at all discontinuous*

*points $t$, then $\mu$ is differentiable almost everywhere, and the semi-indefinite integral*

$$\eta(t) = \mu(t) + \int_0^t \Phi(t-s)\eta(s)ds, \quad \forall t \in [0,T], \tag{4.13}$$

*yields a rate function $\eta : [0,T] \to \mathbb{R}_+^n$ given by*

$$\eta(t) = \int_0^t \Psi(t-s)d\mu(s). \tag{4.14}$$

*Proof.* It suffices to show (4.14) for absolutely continuous $\mu(t)$ on $[0,T)$ since extending the proof to piecewise absolutely continuous function is straightforward. We first define $\mu(T) = \mu(T-)$ and obtain a continuous $\mu(t)$ on $[0,T]$. Since $[0,T]$ is compact, we know $\mu(t)$ is uniformly continuous, and hence there exists a sequence of piecewise constant functions $\{\mu_k\}_{k=1}^{\infty}$ such that $\mu_k \to \mu$ uniformly on $[0,T]$, i.e., $\lim_{k\to\infty} \sup_{0\le t\le T} |\mu_k(t)-\mu(t)| = 0$ [97, theorem 2.3.6] This also implies that $\{\mu_k\}$ is uniformly bounded. For every $k$, piecewise constant function $\mu_k$ has bounded variation, therefore we have by [61, theorem 3.36] that

$$\eta_k(t) := \int_0^t \Psi(t-s)d\mu_k(s) = \int_0^t \Psi'(t-s)\mu_k(s)ds + \Psi(0)\mu_k(t) - \Psi(t)\mu_k(0), \tag{4.15}$$

for all $t \in [0,T]$. Since $\Psi \in C^1$ we know $\Psi'$ is continuous and bounded on $[0,T]$. By Lebesgue's bounded convergence theorem we know

$$\int_0^t \Psi'(t-s)\mu_k(s)ds \to \int_0^t \Psi'(t-s)\mu(s)ds. \tag{4.16}$$

Furthermore, using the uniform convergence of $\{\mu_k\}$ to $\mu$, we know the right hand side (4.15) converges to $\int_0^t \Psi'(t-s)\mu(s)ds + \Psi(0)\mu(t) - \Psi(t)\mu(0)$. Then integration by parts for piecewise absolutely continuous function $\mu$ which has bounded variation implies that $\eta(t) = \int_0^t \Psi(t-s)d\mu(s)$ for all $t \in [0,T]$. □

**Corollary 9.** *Suppose $\Psi$ and $\mu$ satisfy the same conditions as in theorem 8, and define $\psi = \Psi'$, then the rate function is $\eta(t) = (\psi * \mu)(t)$. In particular, if $\Phi(t) = Ae^{-\omega t}\mathbf{1}_{\ge 0}(t) = [a_{ij}e^{-\omega t}\mathbf{1}_{\ge 0}(t)]_{ij}$ then the rate function $\eta(t) = \mu(t) + A\int_0^t e^{(A-wI)(t-s)}\mu(s)ds$.*

*Proof.* Note that both $\psi$ and $\mu$ have supports in $\mathbb{R}_+$, therefore integration by parts and the property of derivative of convolution [25, p. 126] imply that $\eta(t) = \int_0^t \Psi'(t-s)\mu(s)ds =$

$(\psi * \mu)(t)$. If $\Phi(t) = Ae^{-\omega t}\mathbf{1}_{\geq 0}(t) = [a_{ij}e^{-\omega t}\mathbf{1}_{\geq 0}(t)]_{ij}$, then $\Psi(t)$ is given in (4.8), and hence $\psi(t) = Ae^{(A-wI)t}$ and the closed form of $\eta(t)$ follows as in the claim. □

## 4.5 Multi-stage Closed-loop Control Problem

Given the analytical relation between exogenous intensity and expected overall intensity (rate function), one can solve a single one-stage campaigning problem to find the optimal constant intervention intensity [55]. Alternatively, the time window can be partitioned into multiple stages and one can impose different levels of interventions in these stages. This yields an open-loop optimization of the cost function where one selects all the intervention actions at initial time 0. More effectively, we tackle the campaigning problem in a dynamic and adaptive manner where we can postpone deciding the intervention by observing the process until the next stage begins. This is called the *closed-loop* optimization of the objective function.

In this section, we establish the foundation to formulate the problem as a multi-stage closed-loop optimal control problem. We assume that $n$ users are generating events according to multi-dimensional Hawkes process with exogenous intensity $\mu(t) \in \mathbb{R}^n$ and impact function $\Phi(t, s) \in \mathbb{R}^{n \times n}$.

### 4.5.1 Event Exposure

Event exposure is the quantity of major interests in campaigning. The exposure process is mathematically represented as a counting process, $\mathcal{E}(t) = (\mathcal{E}^1(t), \ldots, \mathcal{E}^n(t))^\top$: Here, $\mathcal{E}^i(t)$ records the number of times user $i$ is exposed (she or one of her neighbors performs an activity) to the campaign by time $t$. Let $B$ be the adjacency matrix of the user network, *i.e.*, $b_{ij} = 1$ if user $i$ follows user $j$ or equivalently user $j$ influences user $i$. We assume $b_{ii} = 1$ for all $i$. Then the exposure process is given by $\mathcal{E}(t) = B\,\mathcal{N}(t)$.

### 4.5.2 Stages and Interventions

Let $[0, T]$ be the time horizon and $0 = \tau_0 < \tau_1 < \ldots < \tau_{M-1} < \tau_M = T$ be a partition into the $M$ stages. In order to steer the activities of network towards a desired level (criteria given below) at these stages, we impose a constant intervention $u_m \in \mathbb{R}^n$ to the existing exogenous intensity $\mu$ during time $[\tau_m, \tau_{m+1})$ for each stage $m = 0, 1, \ldots, M-1$. The activity intensity at the $m$-th stage is $\lambda_m(t) = \mu + u_m + \int_0^t \Phi(t, s)\, d\mathcal{N}(s)$ for $\tau_m \leq t < \tau_{m+1}$ where $\mathcal{N}(t)$ tracks the counting process of activities since $t = 0$. Note that the intervention itself exhibits a stochastic nature: adding $u_m^i$ to $\mu^i$ is equivalent to incentivizing user $i$ to increase her activity rate but it is still uncertain when she will perform an activity, which appropriately mimics the randomness in real-world campaigning.

### 4.5.3 States and State Evolution

Note that the Hawkes process is non-Markov and one needs complete knowledge of the history to characterize the entire process. However, the conditional intensity $\lambda(t)$ only depends on the state of process at time $t$ when the standard exponential kernel $\Phi(t, s) = Ae^{-\omega(t-s)}\mathbf{1}_{\geq 0}(t-s)$ is employed. In this case, the activity rate at stage $m$ is

$$\lambda_m(t) = \mu + u_m + \underbrace{\int_0^{\tau_m} Ae^{-\omega(t-s)}\, d\mathcal{N}(s)}_{\text{from previous stages}} + \underbrace{\int_{\tau_m}^{t} Ae^{-\omega(t-s)}\, d\mathcal{N}(s)}_{\text{current stage}} \tag{4.17}$$

Define $x_m := \lambda_{m-1}(\tau_m) - u_{m-1} - \mu$ (and $x_0 = 0$ by convention) then the intensity due to events of all previous $m$ stages can be written as $\int_0^{\tau_m} Ae^{-\omega(t-s)}\, d\mathcal{N}(s) = x_m e^{-\omega(t-\tau_m)}$. In other words, $x_m$ is sufficient to encode the information of activity in the past $m$ stages that is relevant to future. This is in sharp contrast to the general case where the state space grows with the number of events.

### 4.5.4 Objective Function

For a sequence of controls $u(t) = \sum_{m=0}^{M-1} u_m \mathbf{1}_{[\tau_m, \tau_{m+1})}(t)$, the activity counting process $\mathcal{N}(t)$ is generated by intensity $\lambda(t) = \mu + u(t) + \int_0^t A e^{-\omega(t-s)}\, d\mathcal{N}(s)$. For each stage $m$ from 0 to $M-1$, $x_m$ encodes the effects from previous $m$ stages as above and $u_m$ is the current control imposed at this stage. Let $\mathcal{E}_m^i(t; x_m, u_m) := B \int_{\tau_m}^t d\mathcal{N}^i(s)$ be the number of times user $i$ is exposed to the campaign by time $t \in [\tau_m, \tau_{m+1})$ in stage $m$, then the goal is to steer the expected total number of exposure $\bar{\mathcal{E}}_m^i(x_m, u_m) := \mathbb{E}[\mathcal{E}_m^i(\tau_{m+1}; x_m, u_m)]$ to a desired level. In what follows, we introduce several instances of the objective function $g(x_m, u_m)$ in terms of $\{\bar{\mathcal{E}}_m^i(x_m, u_m)\}_{i=1}^n$ in each stage $m$ that characterize different *exposure shaping* tasks. Then the overall control problem is to find $u(t)$ that optimizes the total objective $\sum_{m=0}^{M-1} g_m(x_m, u_m)$.

- *Capped Exposure Maximization (CEM)*: In real networks, there is a cap on the exposure each user can tolerate due to the limited attention of a user. Suppose we know the upper bound $\beta_m^i$ , on user $i$'s exposure tolerance over which the extra exposure is not counted towards the objective. Then, we can form the following *capped exposure maximization*

$$g_m(x_m, u_m) = \frac{1}{n} \sum_{i=1}^{n} \min \left\{ \bar{\mathcal{E}}_m^i(x_m, u_m), \beta_m^i \right\} \tag{4.18}$$

- *Minimum Exposure Maximization (MEM)*: Suppose our goal is instead to maintain the exposure of campaign on each user above a certain minimum level, at each stage or, alternatively to make the user with the minimum exposure as exposed as possible, we can consider the following cost function:

$$g_m(x_m, u_m) = \min_i \bar{\mathcal{E}}_m^i(x_m, u_m) \tag{4.19}$$

- *Least-squares Exposure Shaping (LES)*: Sometimes we want to achieve a pre-specified target exposure levels, $\gamma_m \in \mathbb{R}^n$, for the users. For example, we may like to divide users into groups and desire a different level of exposure in each group. To this end, we can perform least-squares campaigning task with the following cost function where $D$

encodes potentially additional constraints (e.g., group partitions):

$$g_m(x_m, u_m) = -\frac{1}{n}\|D\bar{\mathcal{E}}_m(x_m, u_m) - \gamma_m\|^2 \tag{4.20}$$

### 4.5.5 Policy and Actions

By observing the counting process in previous stages (summarized in a sequence of $x_m$) and taking the future uncertainty into account, the control problem is to design a policy $\pi = \{\pi_m : \mathbb{R}^n \to \mathbb{R}^n : m = 0, \ldots, M-1\}$ such that the controls $u_m = \pi_m(x_m)$ can maximize the total objective $\sum_{m=0}^{M-1} g_m(x_m, u_m)$. In addition, we may have constraints on the amount of control. For example, a budget constraint on the sum of all interventions to users at each stage, or, a cap over the amount of intensity a user can handle. A feasible set or an action space over which we find the best intervention is represented as $\mathcal{U}_m := \left\{u_m \in \mathbb{R}^n | c_m^\top u_m \leq C_m, 0 \leqslant u_m \leqslant \alpha_m\right\}$. Here, $c_m \in \mathbb{R}_+^n$ contains the price of each person per unit increase of exogenous intensity and $C_m \in \mathbb{R}_+$ is the total budget at stage $m$. Also, $\alpha_m \in \mathbb{R}_+^n$ is the cap on the amount of activities of the users.

To summarize, the following problem is formulated to find the optimal policy $\pi$:

$$\begin{aligned} &\text{maximize}_\pi \textstyle\sum_{m=0}^{M-1} g_m(x_m, \pi_m(x_m)), \\ &\text{subject to } \pi_m(x_m) \in \mathcal{U}_m, \text{ for } m = 0, \ldots, M-1. \end{aligned} \tag{4.21}$$

## 4.6 Closed-loop Dynamic Programming Solution

We have formulated the control problem as an optimization in (4.21). However, when control policy $\pi_m$ is to be implemented, only $x_m$ is observed and there are still uncertainties in future $\{x_{m+1}, \ldots, x_{M-1}\}$. For instance, when $\pi_m$ is implemented according to $x_m$ starting from time $\tau_m$, the intensity $x_{m+1} := f(x_m, \pi_m(x_m))$ at time $\tau_{m+1}$ depends on $x_m$ and the control $\pi_m(x_m)$, but is also random due to the stochasticity of the process during time $[\tau_m, \tau_{m+1})$. Therefore, the design of $\pi$ needs to take future uncertainties into considerations.

Suppose we have arrived at stage $M$ at time $\tau_{M-1}$ with observation $x_{M-1}$, then the op-

timal policy $\pi_{M-1}$ satisfies $g_{M-1}(x_{M-1}, \pi_{M-1}(x_{M-1})) = \max_{u \in \mathcal{U}_{M-1}} g_{M-1}(x_{M-1}, u) =: J_{M-1}(x_{M-1})$. We then repeat this procedure for $m$ from $M-1$ to 0 backward to find the sequence of controls via dynamic programming such that the control $\pi_m(x_m) \in \mathcal{U}_m$ yields optimal objective value

$$J_m(x_m) = \max_{u_m \in \mathcal{U}_m} \mathbb{E}[g_m(x_m, u_m) + J_{m+1}(f(x_m, u_m))] \tag{4.22}$$

### 4.6.1 Approximate Dynamic Programming

Solving (4.22) for finding $J_m(x_m)$ analytically is intractable. Therefore, we will adopt an approximate dynamic programming scheme. In fact approximate control is as essential part of dynamic programming as the optimization is usually intractable due to curse of dimensionality except a few especial cases [16]. Here we adopt a suboptimal control scheme, *certainty equivalent control* (CEC), which applies at each stage the control that would be optimal if the uncertain quantities were fixed at some typical values like the average behavior. It results in an optimal control sequence, the first component of which is used at the current stage, while the remaining components are discarded. The procedure is repeated for the remaining stages. Algorithm 6 summarizes the dynamic programing steps. This algorithm has two parts: (i) certainty equivalence which the random behavior is replaced by its average; and (ii) the open-loop optimization. Let's assume we are at the beginning of stage $l$ of the algorithm 6 with state vector $x_l$ at $\tau_l$.

### 4.6.2 Certainty Equivalence

We use the machinery developed in Sec. 4.4 to compute the average of exposure at any stage $m = l, l+1, \ldots, M-1$.

$$\bar{\mathcal{E}}_m(x_m, u_m) = B\mathbb{E}[\mathcal{N}(\tau_{m+1}) - \mathcal{N}(\tau_m)] = B\mathbb{E}\left[\int_{\tau_m}^{\tau_{m+1}} d\mathcal{N}(s)\right] = B\int_{\tau_m}^{\tau_{m+1}} \eta_m(s)\, ds \tag{4.23}$$

**Algorithm 6** Closed-loop Multi-stage Dynamic Programming

**Input:** Intervention constraints: $c_0 \dots c_{M-1}$, $C_0 \dots C_{M-1}$, $\alpha_0 \dots \alpha_{M-1}$,
**Input:** Objective-specific constraints: $\beta_0 \dots \beta_{M-1}$ for CEM and $\gamma_0 \dots \gamma_{M-1}$ for LES
**Input:** Time: $T$, Hawkes parameters: $A$, $\omega$
**Output:** Optimal intervention $u_0 \dots u_{M-1}$, Optimal cost: $Cost$
Set $x_0 \leftarrow 0$
Set $Cost \leftarrow 0$
**for** $l \leftarrow 0 : M-1$ **do**
  $(v_l \dots v_{M-1})$ = $open\_loop(x_l)$ (Problems (4.33), (4.34), (4.35) for CEM, MEM, LES respectively)
  Set $u_l \leftarrow v_l$
  Drop $v_{l+1} \dots v_{M-1}$
  Update next state $x_{l+1} \leftarrow f_l(x_l, u_l)$
  Update $Cost \leftarrow Cost + g_l(x_l, u_l)$
**end for**

where $\eta_m(t) = \mathbb{E}[\lambda_m(t)]$ and $\lambda_m(t) = \mu + u_m + x_l e^{-\omega(t-\tau_l)} + \int_{\tau_l}^t A e^{-\omega(t-s)} d\mathcal{N}(s)$ for $t \in [\tau_m, \tau_{m+1})$. Now, we use the superposition property of point processes [40] to decompose the process as $\mathcal{N}(t) = \mathcal{N}^c(t) + \mathcal{N}^v(t)$ corresponding to $\lambda_m(t) = \lambda_m^c(t) + \lambda_m^v(t)$ where the first $\lambda_m^c(t) = \mu + u_m + \int_{\tau_l}^t A e^{-\omega(t-s)} d\mathcal{N}^c(s)$ consists of events caused by exogenous intensity at current stage $m$ and the second $\lambda_m^v(t) = x_l e^{-\omega(t-\tau_l)} + \int_{\tau_l}^t A e^{-\omega(t-s)} d\mathcal{N}^v(s)$ is due to activities in previous stages. According to theorem 7 we have

$$\eta_m^c(t) := \mathbb{E}[\lambda_m^c(t)] = \Psi(t-\tau_l)\mu + \Psi(t-\tau_l)u_l + \sum_{k=l+1}^{m-1} \Psi(t-\tau_k)(u_k - u_{k-1}), \quad (4.24)$$

and according to theorem 8 we have

$$\eta_m^v(t) := \mathbb{E}[\lambda_m^v(t)] = \int_{\tau_l}^t \Psi(t-s)\, d(x_l e^{-\omega(s-\tau_l)} \mathbf{1}_{[\tau_l,\infty)}(s)). \quad (4.25)$$

From now on, for simplicity, we assume stages are based on equal partition of $[0, T]$ to $M$ segments where each has length $\Delta_M$. Combining Eq. (4.23) and $\eta_m(t) = \eta_m^c(t) + \eta_m^v(t)$ yields:

$$\begin{aligned}\bar{\mathcal{E}}_m(x_m, u_m) =& \Gamma((m-l+1)\Delta_M)u_l + \Gamma((m-l)\Delta_M)(u_{l+1} - u_l) + \dots \\ &+ \Gamma(\Delta_M)(u_m - u_{m-1}) + \Gamma((m-l+1)\Delta_M)\mu + \Upsilon((m-l+1)\Delta_M)x_l\end{aligned} \quad (4.26)$$

where $\Gamma(t)$ and $\Upsilon(t)$ are matrices independent of $u_m$'s and are defined as:

$$\Upsilon(t) = B \int_0^t e^{(A-\omega I)s}\, ds = B(A-\omega I)^{-1}(e^{(A-\omega I)t} - I) \tag{4.27}$$

$$\Gamma(t) = B \int_0^t \Psi(s)\, ds = BIt + BA(A-\omega I)^{-1}(\Upsilon(t) - It); \tag{4.28}$$

Note the linear relation between average exposure $\bar{\mathcal{E}}_m(x_m, u_m)$ and interventions $u_l, \ldots, u_{m-1}$.

Let $\Gamma(k\Delta_M) = \Gamma_k$ and $\Upsilon(k\Delta_M) = \Upsilon_k$. Then for every $m \geq l$, Eq. (4.26) is rewritten:

$$\bar{\mathcal{E}}_m(x_m, u_m) = \sum_{k=l}^{m-1} (\Gamma_{m-k+1} - \Gamma_{m-k})u_k + \Gamma_1 u_m + \Gamma_{m-l+1}\mu + \Upsilon_{m-l+1}x_l. \tag{4.29}$$

### 4.6.3 Open-loop Optimization

Having found the average exposure at stages $m = l, \ldots, M-1$ we formulate an open-loop optimization to find optimal $u_l, u_{l+1}, \ldots, u_{M-1}$. Defining $\hat{u}_l = (u_l; \ldots; u_{M-1})$ and $\hat{\mathcal{E}}_l = (\bar{\mathcal{E}}_l(x_l, u_l); \ldots; \bar{\mathcal{E}}_{M-1}(x_{M-1}, u_{M-1}))$ we can aggregate Aggregating these linear forms for all $l \geq m$ yields to the following matrix equation for finding $\hat{\mathcal{E}}_l$:

$$\underbrace{\begin{bmatrix} \bar{\mathcal{E}}_l(x_l, u_l) \\ \bar{\mathcal{E}}_{l+1}(x_{l+1}, u_{l+1}) \\ \bar{\mathcal{E}}_{l+2}(x_{l+2}, u_{l+2}) \\ \vdots \\ \bar{\mathcal{E}}_{M-1}(x_{M-1}, u_{M-1}) \end{bmatrix}}_{\hat{\mathcal{E}}_l} = \underbrace{\begin{bmatrix} \Gamma_1 & 0 & 0 & \ldots & 0 \\ \Gamma_2 - \Gamma_1 & \Gamma_1 & 0 & \ldots & 0 \\ \Gamma_3 - \Gamma_2 & \Gamma_2 - \Gamma_1 & \Gamma_1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Gamma_{M-l} - \Gamma_{M-l-1} & \Gamma_{M-l-1} - \Gamma_{M-l-2} & \ldots & \Gamma_2 - \Gamma_1 & \Gamma_1 \end{bmatrix}}_{X_l} \underbrace{\begin{bmatrix} u_l \\ u_{l+1} \\ u_{l+2} \\ \vdots \\ u_{M-1} \end{bmatrix}}_{\hat{u}_l}$$

$$+ \underbrace{\begin{bmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \\ \vdots \\ \Gamma_{M-l} \end{bmatrix}}_{Y_l} \mu + \underbrace{\begin{bmatrix} \Upsilon_1 \\ \Upsilon_2 \\ \Upsilon_3 \\ \vdots \\ \Upsilon_{M-l} \end{bmatrix}}_{W_l} x_l \tag{4.30}$$

Defining the expanded form of constraint variables as $\hat{c}_l = (c_l; \ldots; c_{M-1})$, $\hat{C}_l = (C_l; \ldots; C_{M-1})$, and $\hat{\alpha}_l = (\alpha_l; \ldots; \alpha_{M-1})$ we have

$$\underbrace{\begin{bmatrix} c_l^\top & \ldots & 0^\top \\ \vdots & \ddots & \vdots \\ 0^\top & \ldots & c_{M-1}^\top \\ I & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & I \\ -I & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & -I \end{bmatrix}}_{Z_l} \underbrace{\begin{bmatrix} u_l \\ u_{l+1} \\ u_{l+2} \\ \vdots \\ u_{M-1} \end{bmatrix}}_{\hat{u}_l} \leq \underbrace{\begin{bmatrix} C_l \\ \vdots \\ C_{M-1} \\ \alpha_l \\ \vdots \\ \alpha_{M-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{z} . \tag{4.31}$$

In summary, the following equation states the relation between intervention intensities given the constrains:

$$X_l \hat{u}_l + Y_l \mu + W_l x_l = \hat{\mathcal{E}}_l \quad \text{where} \quad Z_l \hat{u}_l \leq z_l \tag{4.32}$$

Then, we provide the optimization from of the above exposure shaping tasks.

For CEM consider $\hat{\beta}_l = (\beta_l; \ldots, \beta_{M-1})$. Then the problem

$$\begin{aligned} &\text{maximize}_{\hat{h}, \hat{u}_l} \tfrac{1}{n} \mathbf{1}^\top \hat{h} \\ &\text{subject to } X_l \hat{u}_l + Y_l \mu + W_l x_l \geq \hat{h}, \;\; \hat{\beta}_l \geq \hat{h}, \;\; Z_l \hat{u}_l \leq z_l, \end{aligned} \tag{4.33}$$

solves CEM where $\hat{h}$ is an auxiliary vector of size $n(M-l)$.

For MEM consider the auxiliary $h$ as a vector of size $M-l$ and $\hat{h}$ a vector of size $n(M-1)$. $\hat{h} = (h(1); \ldots; h(1); h(2); \ldots, h(2); \ldots, h(M-l); \ldots; h(M-l))$ where each $h(k)$ is repeated $n$ times. Then MEM is equivalent to

$$\begin{aligned} &\text{maximize}_{\hat{h}, \hat{u}_l} \mathbf{1}^\top \hat{h} \\ &\text{subject to } X_l \hat{u}_l + Y_l \mu + W_l x_l \geq \hat{h}, \;\; \hat{\beta}_l \geq \hat{h}, \;\; Z_l \hat{u}_l \leq z_l \end{aligned} \tag{4.34}$$

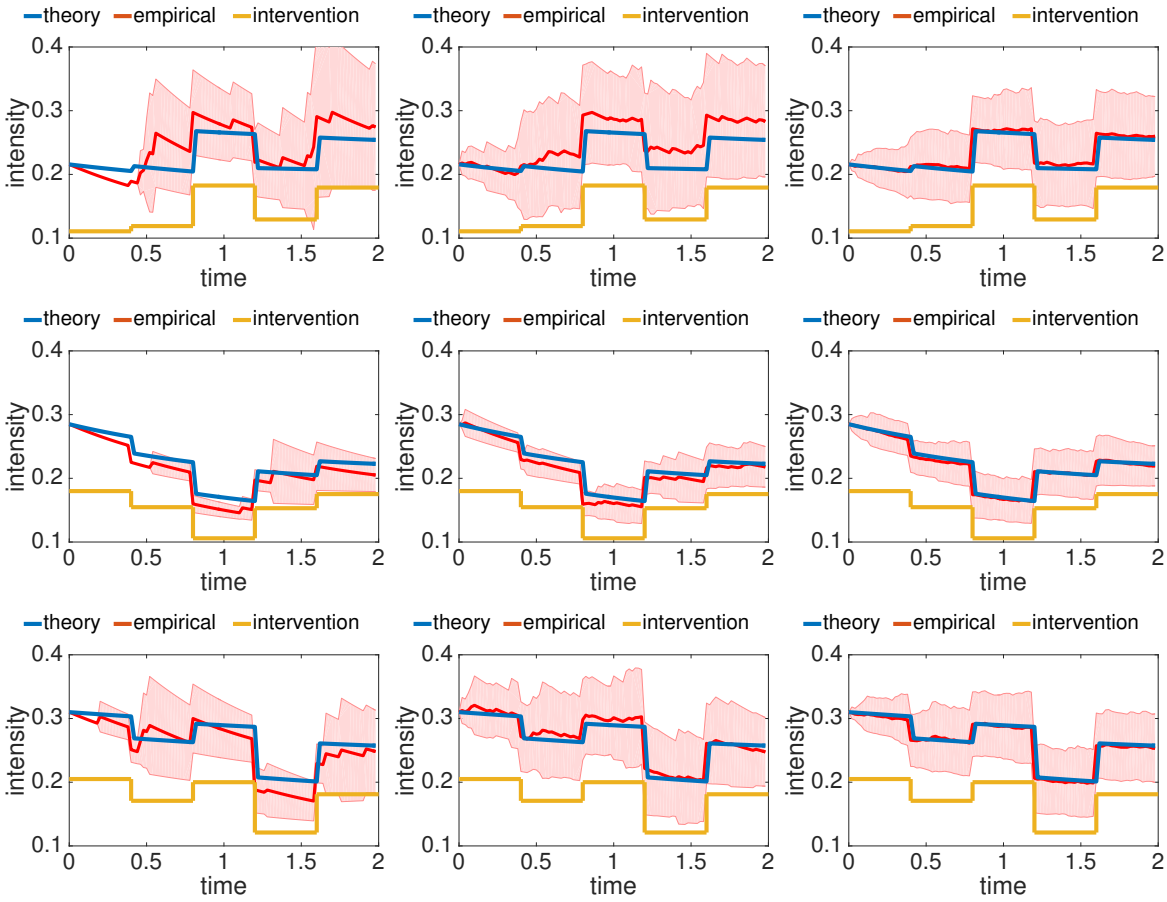
For LES let $\hat{\gamma}_l = (\gamma_l; \ldots; \gamma_{M-1})$ and $\hat{D}_l = diag(D, \ldots, D)$, then

$$\begin{aligned} &\text{minimize}_{\hat{u}_l} \tfrac{1}{n} \|\hat{D}_l(X_l \hat{u}_l + Y_l \mu + W_l x_l) - \hat{\gamma}_l\|^2 \\ &\text{subject to } Z_l \hat{u}_l \leq z_l \end{aligned} \tag{4.35}$$

All the three tasks involve convex (and linear) objective function with linear constraints which impose a convex feasible set. Therefore, one can use the rich and well-developed literature on convex optimization and linear programming to find the optimum intervention.

### 4.6.4 Scalable Optimization

All the exposure shaping problems defined above require an efficient evaluation of average intensity $\eta(t)$ at all stages, which entails computing matrices $X_l$, $Y_l$, $W_l$, and $Z_l$. This leads to work with matrix exponentials and inverse matrices to obtain $\Upsilon_m$, and $\Gamma_m$ for $m = 1, \ldots, M-1$. In small or medium networks, we can rely on well-known numerical methods to compute matrix exponentials and inverse. However, in large networks, the explicit computation of $X_l$, $Y_l$, $W_l$, and $Z_l$ becomes intractable. Fortunately, we can exploit the following key property of our convex campaigning framework: the average intensity itself and the gradient of the objective functions only depends on $X_l$, $Y_l$, $W_l$, and $Z_l$ (and consequently on $\Upsilon_m$, and $\Gamma_m$) through matrix-vector product operations. Similar to [55] for the computation of the product of matrix exponential with a vector, one can use the iterative algorithm by Al-Mohy et al. [4], which combines a scaling and squaring method with a truncated Taylor series approximation to the matrix exponential. For solving the sparse linear system of equation, we use the well-known GMRES method, which is an Arnoldi process for constructing an $l_2$-orthogonal basis of Krylov subspaces. The method solves the linear system by iteratively minimizing the norm of the residual vector over a Krylov subspace. For details please refer to [55]. Last but not least, we don't need to explicitly build $X_l$, $Y_l$, $W_l$, and $Z_l$. At each step of gradient computation all the operations involving them are multiplication of$\Upsilon_1, \ldots, \Upsilon_M$, and $\Gamma_1, \ldots, \Gamma_M$ to vectors such as $u_0, \ldots, u_{M-1}$ and $\mu$.

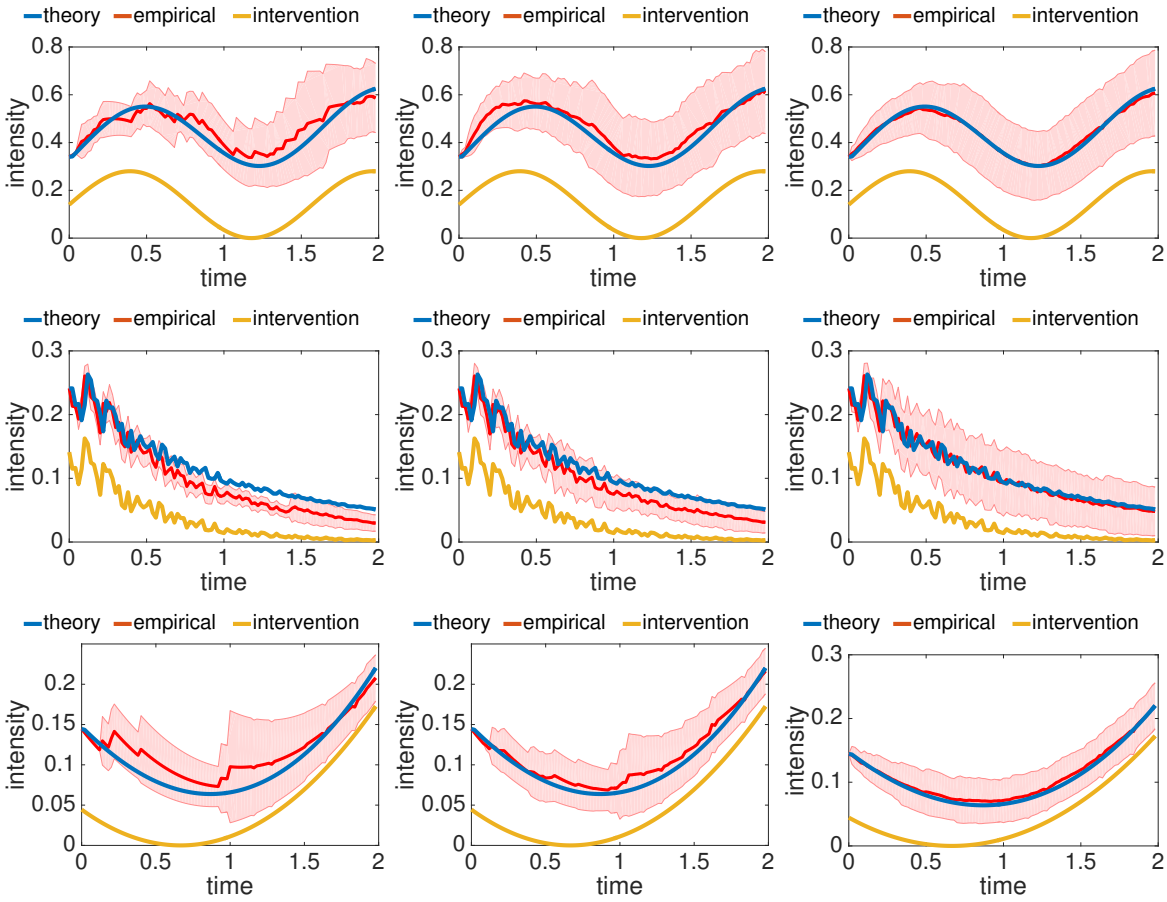



a) 5 runs b) 20 runs c) 100 runs

Figure 4.2: Empirical investigation of theoretical results in theorem 7. blue: theoretical average intensity; red: empirical average intensity and sample standard deviation; orang: piecewise-constant exogenous intensity (interventions)

## 4.7 Temporal Properties

In this section we empirically study the theoretical results of section 4.4. The empirical mean and standard deviation of the intensity averaged over multiple number of cascades is compared theoretical mean. Besides this, the other purpose of the experiment is to advocate verification process in the synthetic experiments when we used simulation to evaluate the merits of the proposed algorithm and compare to the baselines. In other words, we show that the empirical activity (and hence the average exposure) is very close to its theoretical value and it is justifiable to be used for the comparison.




a) 5 runs b) 20 runs c) 100 runs

Figure 4.3: Empirical investigation of theoretical results in theorem 8. blue: theoretical average intensity; red: empirical average intensity and sample standard deviation; orang: general time-varying intensity (interventions)

Figure 4.2 demonstrates the activity profile of 3 random users picked in a network of 300 ones simulated 100 times to investigate theorem 7. The setting is similar to main synthetic experiment. Piecewise exogenous intensity (interventions) are picked randomly in $[0.1, 0.2]$ with a slight noise. We consider 5 stages for changing the exogenous intensity. The empirical average and standard deviation is compared to theoretical average intensity for 3 different number of runs namely, 5, 20, and 100 times. Also, Figure 4.3 demonstrate the general case where the exogenous intensity is a time-varying function in theorem 8. We take 3 sample functions to investigate this case; a sinusoidal function; an exponential decaying function with added noise; and a quadratic function.

We observe a couple of interesting facts. Firstly, it's apparent that by increasing the number of averages the empirical intensity tends to theoretical one very fast. Secondly, as the mean becomes more accurate by increasing the number of cascades the standard deviation increases; e.g., compare the standard deviation in first and third column. Thirdly, the standard deviation is increasing with time. This is due to the fact that as time passes random elements are aggregated more and this increases variance.

## 4.8 Experiments

We evaluate our campaigning framework using both simulated and real world data and show that our approach significantly outperforms several baselines.

### 4.8.1 Synthetic Data Generation

The network is generated synthetically using varying number of nodes. Initial exogenous intensity is set uniformly at random, $u_m^i \sim \mathcal{U}[0, 0.1]$. Endogenous intensity coefficients (influence matrix elements) are set similarly, $a_{ij} \sim \mathcal{U}[0, 0.1]$. To mimic sparse real networks half of the elements are set to 0 randomly. The matrix is scaled appropriately such that the spectral radius of the coefficient matrix is a random number smaller than one and the stability of the process is ensured.

The upper bound for the intervention intensity is set randomly from interval $\alpha^i \sim \mathcal{U}[0, 0.1]$. The price of each person is set $c_m^i = 1$, and the total budget at stage $m$ is randomly generated as $C_m \sim (n/10)\mathcal{U}[0, 0.1]$. For the capped exposure maximization case the upper bound is set $\alpha_m^i \sim \mathcal{U}[0, 1]$ and target in least-squares exposure shaping is set similarly $v_m^i \sim (n/10)\mathcal{U}[0, 1]$. Furthermore, the shaping matrix $D$ is set to $I$. In all the synthetic experiments $\omega = 0.01$ which roughly means loosing 63 % of influence after 100 units of time (minutes, hours, etc). Furthermore, the exposing matrix is set to the unweighted adjacency matrix *i.e.*, $B_{ij} = 1$ if and only if $A_{ij} \geq 10^{-4}$. This way the results are reported in terms of the exact number of exposures and are easily interpretable. In general applications

any $B$ can be used for example using the influence matrix $A$ yields to a wighted exposure count. In all the synthetic and real experiments the above settings are assumed unless it is explicitly mentioned.

### 4.8.2 Real Data Description and Network Inference

In real data, we use a temporal resolution of one hour and selected the bandwidth $\omega = 0.001$ by cross validation. Roughly speaking, it corresponds to loosing almost 50 % of the initial influence after 1 month. The upper bound for intervention intensity is set uniformly at random with mean equal to empirical intensity learned from data. The upper bound for the cap and target exposure are set similarly. For the 10 pairs of cascades we used first 3 months of data to learn the network parameters. We then drop the exogenous intensity $\mu$, and keep the influence network parameters $A$. By fixing $A$ we use the next 6 months of data to learn the exogenous intensity of sites in the two cascades at each of the $M$ stages and name them $\mu_m^{c_1}$ and $\mu_m^{c_2}$. Given $A$ we find the optimal intervention intensity $u_m^{opt}$ stage by stage, for each of the three exposure shaping tasks assuming $\mu = 0$. Then, our prediction is: cascade $c1$ will reach a better objective value at stage $m$ if $dist(u_m^{opt}, \mu_m^{c_1}) < dist(u_m^{opt}, \mu_m^{c_2})$ and vice versa measured by cosine similarity. The prediction accuracy is then reported as a performance measure.

### 4.8.3 Baselines

In this section, we describe several baselines we compare our approach. Most often, these baseline methods utilize a property to prioritize users for budget assignment.

For the capped exposure maximization problem, we consider the following four baselines:

- **OPL**: It allocates the budget according to the solution to the dynamic programming in an *open loop* setting, *i.e.*, the decisions on the allocation policy are made once and for all at the initial intervention points at initial time $t = 0$. This is very important

baseline to which comparison quantify the so called *value of information* in the context of dynamic programming and optimal control. As the name suggests it indicates how much knowing what happened so far helps making decisions for future. For the minimum and capped exposure maximization creasing $n$ the objective function is normalized by the size of network.

- **RND**: It assigns a random point in the convex space of feasible solutions.

- **PRK**: At each stage it subtracts the previous state ($x_m^i$) from the cap ($\alpha_m^i$) and multiply by the page rank score of the the node ($r^i$) computed with damping factor $0.85$ and allocates the budget proportional to this value, *i.e.*, $u_m^i \propto \max\big((\alpha_m^i - x_m^i)r^i, 0\big)$. The proposed solution is then projected to the feasible set of actions in that stage and the extra amount is redistributed similarly. The process is iterated until all the budget are allocated. This baseline assumes that more central users can leverage the total activity, therefore, assigns the budget dynamically to the more connected users proportional to their page rank score.

- **WEI**: This baseline uses sum of out-going influence ( $q^i = \sum_j a_{ji}$) as a measure of centrality of users. Similar to the previous one it assigns budget dynamically to the users proportionally to $u_m^i \propto \max\big((\alpha_m^i - x_m^i)\, q^i, 0\big)$. This heuristic allows us to understand the effect of considering the whole network and the propagation layout with respect to only consider the direct (out-going) influence.

For the max-min exposure shaping problem, we implement the following four baselines:

- **OPL**: Similar to the previous objective it represents the open loop solution.

- **RND**: Similar to the previous objective it allocates the budget randomly within the feasible set.

- **WFL**: It takes a *water filling* approach. It sorts the users in ascending order of the exposure in the previous stage. Then allocates budget to the first users until the the summation of its previous exposure and the allocated budget reaches the second lowest value or it violates a constraint. Then, assigns the budget to these two until they reach the third user with lowest exposure or a constraint is violated. This process is continued until the budget is allocated.

- **PRP**: It allocates the budget inversely proportional to the the exposure at the previous stage.

For the least-square exposure shaping problem, we compare our method with four baselines:

- **OPL**: Similar to the previous objective it represents the open loop solution.

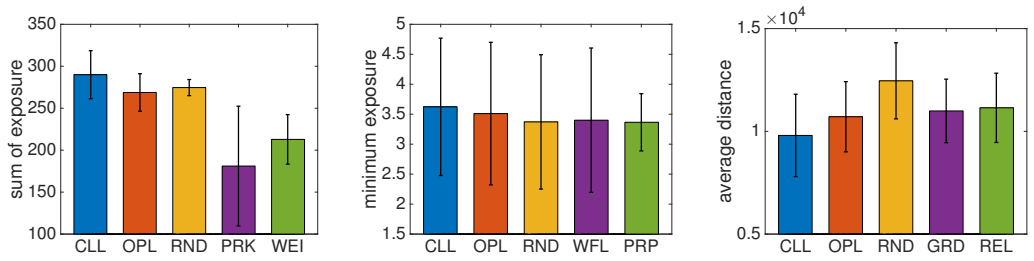
- **RND**: Similar to the previous objective it allocates the budget randomly within the feasible set.

- **GRD**: It finds the difference between the exposure at previous stage ($x_m^i$ and the target from $v$ and sorts them decreasingly. Then, allocates budget one at a time until a constraint is violated. It iterates over the users until the budget is fully allocated.

- **REL**: Similar to the above finds the difference from the target but allocates the budget proportionally, *i.e.*, $u_m^i \propto \max\left((v^i - x_m^i), 0\right)$ for all users. If one allocation violates a constraint the extra amount is reallocated in the same manner.

#### 4.8.4 Campaigning Results on Synthetic Networks

In this section, we experiment with a synthetic network of $300$ nodes. We focus on three tasks: capped exposure maximization, minimax exposure shaping, and least square exposure shaping. To compare the methods we simulate the network with the prescribed




a) Capped maximization b) Minimum maximization c) Least-squares shaping

Figure 4.4: The objective on simulated events and synthetic network; $n = 300$, $M = 6$, $T = 40$

intervention intensity and compute the objective function based on the events happened during the simulation. The mean and standard deviation of the objective function out of 10 runs are reported.

Figure 4.4 summarizes the performance of the proposed algorithm (CLL) and 4 other baselines on different campaigning tasks. For **CEM**, our approach consistently outperforms the others by at least 10. This means it exposes each user to the campaign at least 10 times more than the rest consuming the same budget and within the same constraints. The extra 20 units of exposures of over OPL or value of information shows how much we gain by incorporating a dynamic closed-loop solution as opposed to open-loop one-time optimization over all stages. For **MEM**, the proposed method outperforms the others by a smaller margin, however, the 0.1 exposure difference with the second best method is not trifling. This is expected as lifting the minimum exposure is a difficult task [55]. For **LES**, results demonstrate the superiority of CLL by a large margin. The $10^3$ difference with the second best algorithm aggregated over 6 stages roughly is translated to $\sqrt{10^3/6} \sim 13$ difference in the number of exposures per user. Given the heterogeneity of the network activity and target shape, this is a significant improvement over the baselines.

### 4.8.5 Campaigning Results on Real World Networks

We also evaluate the proposed framework on real world data. To this end, we utilize the MemeTracker dataset [122] which contains the information flows captured by hyperlinks

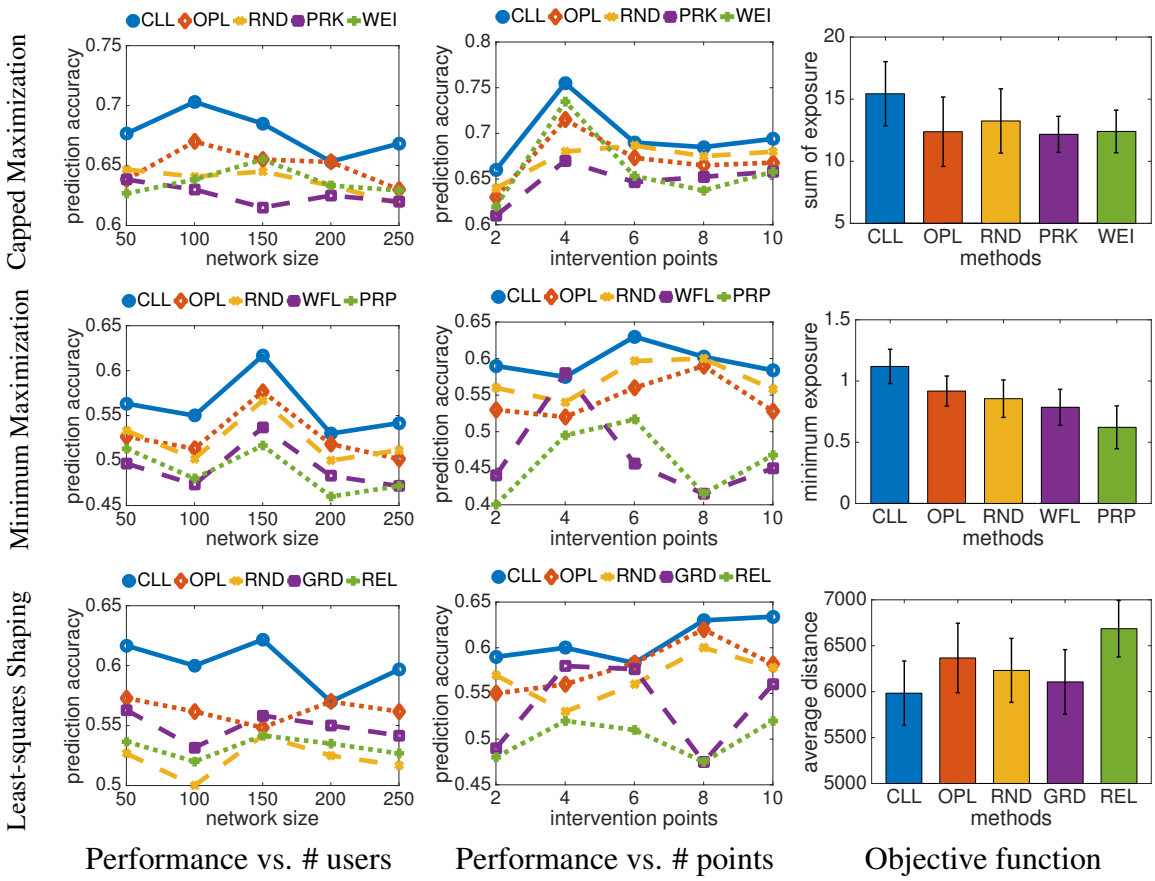



Figure 4.5: real world dataset results; $n = 300$, $M = 6$, $T = 40$

between different sites with timestamps during 9 months. This data has been previously used to validate Hawkes process models of social activity [228, 218]. For the real data, we utilize two evaluation procedures. First, similar to the synthetic case, we simulate the network, but now on a network based on the learned parameters from real data. However, the more interesting evaluation scheme would entail carrying out real intervention in a social media platform. Since this is very challenging to do, instead, in this evaluation scheme we used held-out data to mimic such procedure. Second, we form 10 pairs of clusters/cascades by selecting any 2 combinations of 5 largest clusters in the Memetracker data. Each is a cascade of events around a common subject. For any of these 10 pairs, the methods are faced to the question of predicting which cascade will reach the objective function better. They should be able to answer this by measuring how similar their prescription is to the real

exogenous intensity. The key point here is that the real events happened are used to evaluate the objective function of the methods. Then the results are reported on average prediction accuracy on all stages over 10 runs of random constraint and parameter initialization on 10 pairs of cascades.

Figure 4.5, left column illustrates the performance with respect to increasing the number of users in the network. The performance drops slightly with the network size. This means that prediction becomes more difficult as more random variables are involved. The middle panel shows the performance with respect to increasing the number of intervention points. Here, a slight increase in the performance is apparent. As the number of intervention points increases the algorithm has more control over the outcome and can reach the objective function better.

Figure 4.5 top row summarizes the results of **CEM**. The left panel demonstrates the predictive performance of the algorithms. CLL consistently outperforms the rest. With 65-70 % of accuracy in predicting the optimal cascade. The right panel shows the objective function simulated 10 times with the learned parameters for network of $n = 300$ users on $6$ intervention points. The extra 2.5 extra exposure per user compared to the second best method with the same budget and constraint would be a significant advertising achievement. Among the competitors OPL and RND seem to perform good. If there where no cap over the resultant exposure, all methods would perform comparably because of the linearity of sum of exposure. However, the successful method is the one who manage to maximize exposure considering the cap. Failure of PRK and WEI indicates that structural properties are not enough to capture the influence. Compared to these two, RND performs better in average, however exhibits a larger variance as expected.

Figure 4.5 middle row summarizes the results for **MEM** and shows CLL outperforms others consistently. CLL still is the best algorithm and OPL and RND are the significant baselines. Failure of WFL and PRP shows the network structure plays a significant role in the activity and exposure processes.

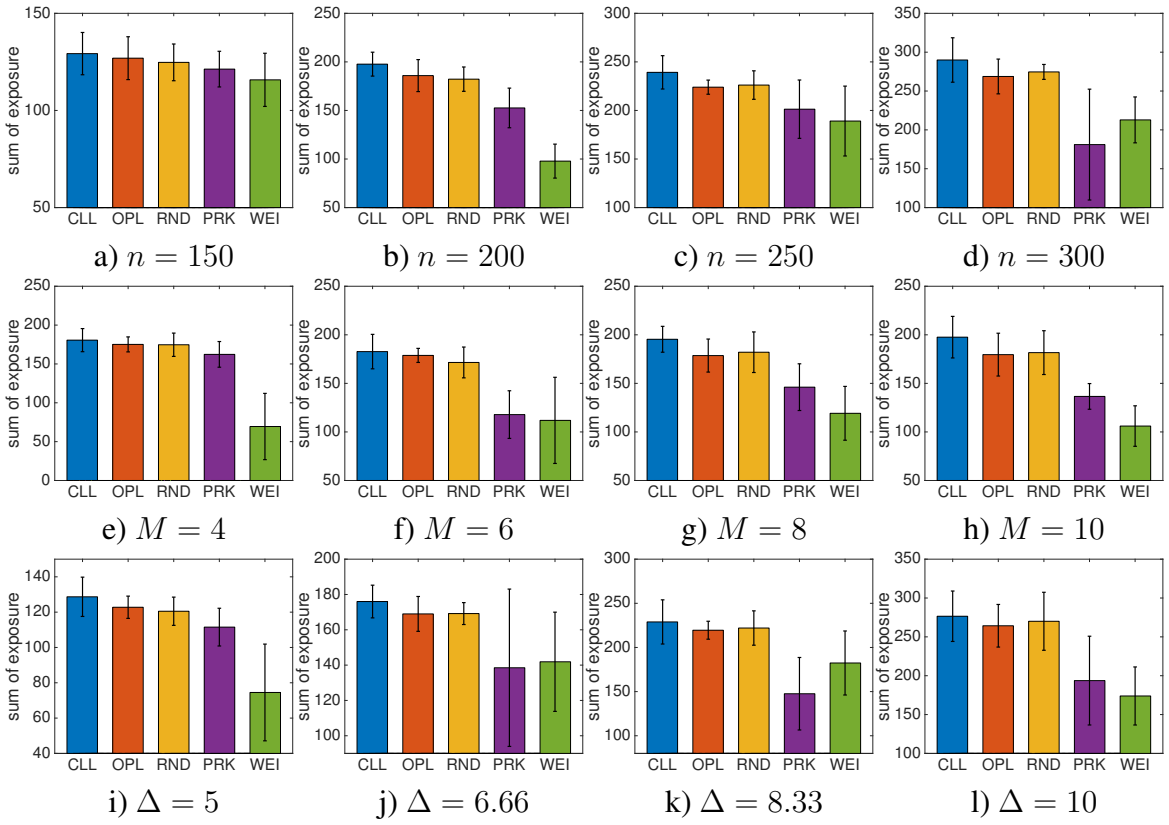



Figure 4.6: Caped exposure maximization results on synthetic data; top row: $n$ varies, $M = 6$, $T = 40$; middle row: $M$ varies, $T = 40$, $n = 200$; bottom row: $T$ varies, $n = 200$, $M = 6$

The bottom row in Figure 4.5 demonstrates the results of **LES**. CLL is still the best method. Among the competitors, OPL is still strong but RND is not performing well for this task. The objective function is summation of the square of the gap between target and current exposure. This explains why GRD is showing a comparable success, since, it starts with the highest gap in the exposure and greedily allocates the budget.

### 4.9 Extended Synthetic Results

For the synthetic case we can freely evaluate the properties of the proposed algorithm under several conditions. We assess the performance of the algorithm and compare to the baselines in three settings: i) increasing size of the network; ii) increasing number of intervention points; iii) increasing the time window (or equivalently the stage duration). The

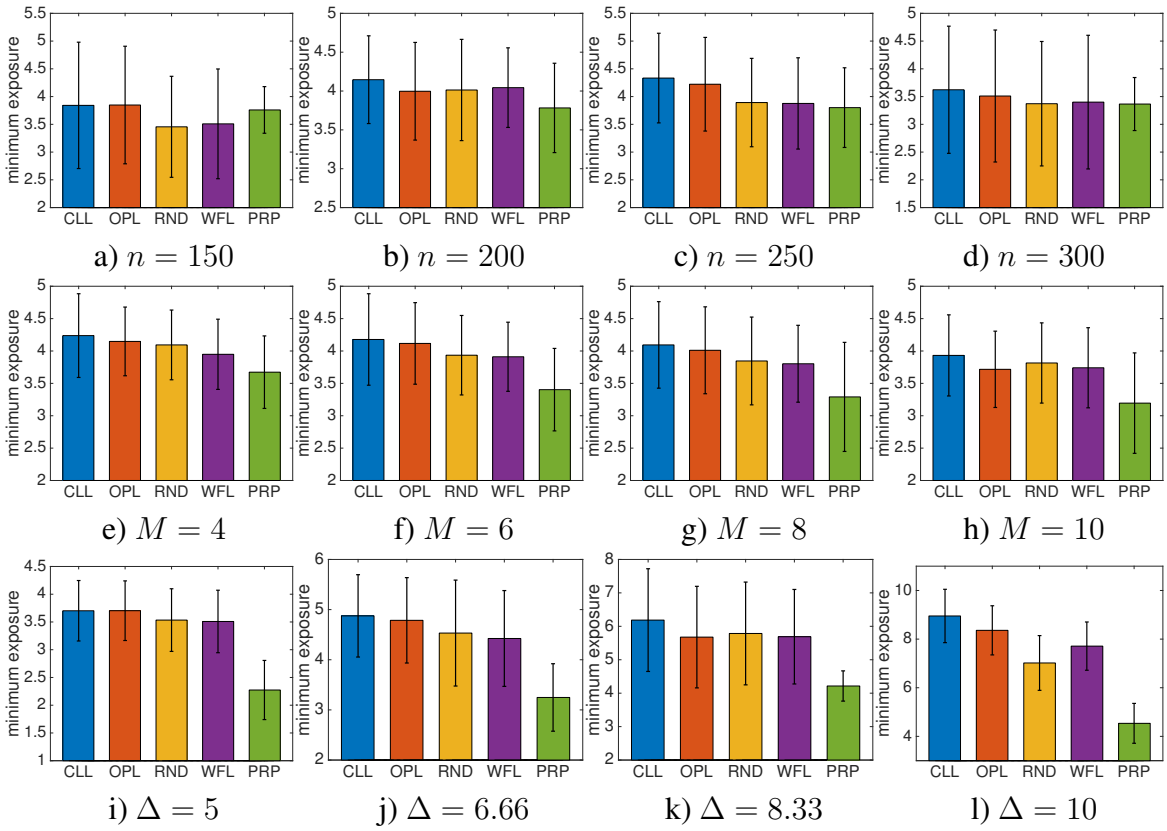



Figure 4.7: Minimum exposure maximization results on synthetic data; top row: $n$ varies, $M = 6$, $T = 40$; middle row: $M$ varies, $T = 40$, $n = 200$; bottom row: $T$ varies, $n = 200$, $M = 6$

results are reported while keeping other parameters fixed. To compare it to the others we simulate the network with the prescribed intervention intensity and compute the objective function. The mean and standard deviation of the objective function out of 10 runs are reported.

Figures 4.6, 4.7, and 4.8 shows the results for CEM, MEM, and LES respectively. In each figure, the first row is for varying number of nodes, the second row is for varying number of intervention points, and the third row is for varying duration of stages. The proposed method is consistently better than the baselines. The trends and facts reported before are observed in this extended experiment. Additionally, we want to refer the high variance of baseline methods especially RND and OPL which is what we expect.

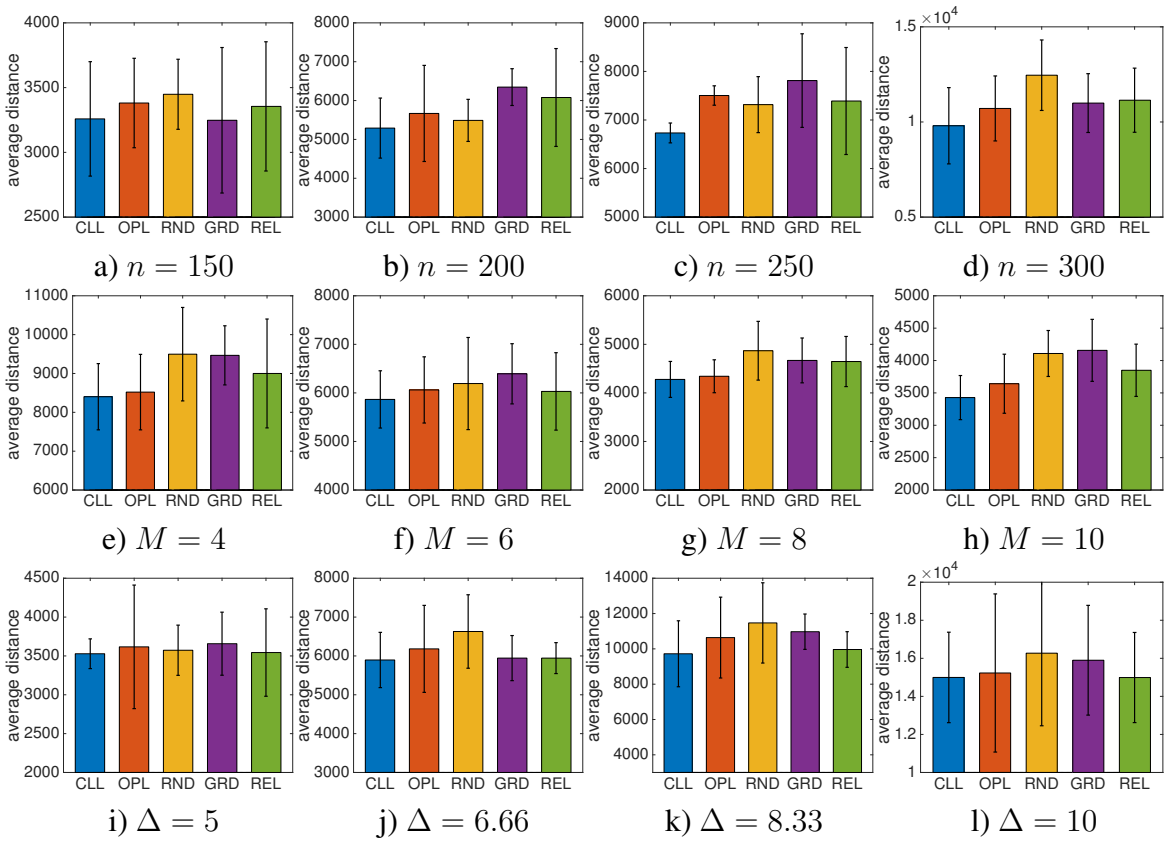



Figure 4.8: Least-squares exposure shaping results on synthetic data; top row: $n$ varies, $M = 6, T = 40$; middle row: $M$ varies, $T = 40$, $n = 200$; bottom row: $T$ varies, $n = 200$, $M = 6$

## 4.10 Related Work

Using Markov Decision Processes and Contextual Bandits in influence maximization, advertising, and campaigning in social network is becoming a hot trend in machine learning community [63, 196, 203, 92, 220, 118]. Exposure shaping problems are significantly more challenging than traditional influence maximization problems, which aim to identify a set of users who influence others in the network and trigger a large cascade of adoptions [161, 111]. First, in influence maximization, the state of each user is often assumed to be binary. However, such assumption does not capture the recurrent nature of social activity. Second, while influence maximization methods identify a set of users to provide incentives, they do not typically provide a quantitative prescription on how much incentive should be provided

to each user. Third, exposure shaping concerns about a larger variety of target states, such as minimum exposure requirement and homogeneity, not just maximization.

Existing work in stochastic optimal control includes jump diffusion stochastic differential equations (SDE) [91, 101] which focuses on controlling the SDEs with the jump term driven by Poisson processes not for Hawkes processes. Inspired by the opinion dynamics model proposed in [41], the authors in [204] proposes a multivariate jump diffusion process framework for modeling opinion dynamics over networks and determining the control over such networks.

In [21], a continuous action iterated prisoners' dilemma was used to model the interactions in a social network and extended by incorporating a mechanism for external influence on the behavior of individual nodes. Markov Decision Process (MDP) framework is proposed to develop several scheduling algorithms for optimal control of information epidemics with susceptible-infected (SI) model on Erdős-Rényi and scale-free networks [108]. In [33], the authors provided an analytically tractable model for information dissemination over networks and solved the optimal control signal distribution time for minimizing the accumulated network cost via dynamic programming. Furthermore, [110] formulated the maximization of spread of a given message in the population within the stipulated time as continuous-time deterministic optimal control problem.

In contrast, our work has been built on the well-developed theory of point processes [1, 40]. Their usage in modeling activity in social network is becoming increasingly popular [130, 150, 89, 56, 151]. More specifically, we utilizes the Hawkes process [93] which its self-exciting property has been proved to be an appropriate choice in modeling processes of and on the networks: [131, 228, 106, 218, 22] model and infer the social activity in networks Based on Hawkes process assumption of activity in social networks [58, 191, 37, 216, 95, 195] study one or several phenomena in the social network. In [109] authors proposed a broadcasting algorithm to maximize the visibility of posts in Twitter. It only consider the direct followers and does not involve peer influence in propagation process.

Our work is closely related to [55], which is extended in two significant directions here: First, we generalize their result on driving a time-dependent average intensity in the case where the exogenous intensity is not constant. Second, instead of one-shot optimization we pose the problem as a multi-stage optimal control problem which is more fit to real world applications. Then we propose a dynamic programming solution to the multi-stage optimization problem.

## 4.11 Summary and Conclusion

In this chapter, we introduced the optimal multistage campaigning problem, which is a generalization of the activity shaping and influence maximization problems, and it allows for more elaborate goal functions. Our model of social activity is based on multivariate Hawkes process, and for the first time, we manage to derive a linear connection between a time-varying exogenous intensity (*i.e.*, the part that can be easily manipulated via incentives) and the overall network exposure of the campaign. The multistage optimal control problem is introduced and an approximate closed-loop dynamic programming approach is proposed to find the optimal interventions. This linear connection between exogenous intensity and campaign's exposure enables developing a convex optimization framework for exposure shaping, deriving the necessary incentives to reach a global exposure pattern in the network. The method is evaluated on both synthetic and real-world held-out data and is shown to outperform several heuristics.

Experiments on synthetic and real world datasets reveal a couple of interesting facts:

- Most notable lesson is the presence of the so-called *value of information*. We have witnessed, both in synthetic and real dataset, it is possible to achieve lower cost, essentially by taking advantage of extra information. If the information was not available the controller couldn't adapt appropriately to the unexpected behavior and consequently the cost could have been adversely affected.

- What we have empirically observed is that the performance, measured in achieving the lower cost and accurate prediction, improves with increasing the number of intervention points. The more control over social network the better one can steer the campaign towards a goal.

- The performance slightly decreases with increasing the number of nodes. That might be due to the increased dimensionality of the optimization problem.

We acknowledge that our method has indeed limitations. For the networks at the scale of web or large social networks faster and scalable methods need to be explored and developed which remains as future works. There are many other interesting venues for future work too. For example, considering competing/collaborating campaigns and their equilibria and interactions, a continuous-time intervention scheme, and exploring other approximate dynamic programming approaches remain as future work.

# CHAPTER 5
# REINFORCEMENT LEARNING FOR OPTIMAL INTERVENTION

We propose a multistage intervention framework that combines reinforcement learning with a point process network activity model. We develop a policy iteration method unique to the multivariate networked point process, with the goal of optimizing the actions for maximal total reward under budget constraints. We then instantiate the proposed model in the task of mitigating fake news. The spread of fake news and mitigation events within the network is modeled by a multivariate Hawkes process with additional exogenous control terms. By choosing a feature representation of states, defining mitigation actions and constructing reward functions to measure the effectiveness of mitigation activities, we map the problem of fake news mitigation into the reinforcement learning framework. Our method shows promising performance in real-time intervention experiments on a Twitter network to mitigate a surrogate fake news campaign, and outperforms alternatives on synthetic datasets.

## 5.1 Introduction

In our path to develop an intervention framework for point processes we formulate a reinforcement learning approach to find the best policy that steers the network activities towards a desired profile. We give the first derivation of second-order statistics of random exposure counts in the non-stationary case, which is essential in policy evaluation and improvement. The optimal policy is learnt in an offline manner using the observed cascades and then deployed on the network.

In this chapter, we evaluate the intervention problem in the task of fake news mitigation. The recent proliferation of malicious fake news in social media has been a source of widespread concern. Given that more than $62\%$ of U.S. adults turn to social media for

news, with $18\%$ doing so often, fake news can have potential real-world consequences on a large scale [76]. For example, within the final three months of the 2016 U.S. presidential election, news stories that favored either of the two nominees–later proved to be fake–were shared over 37 million times on Facebook, and over half of those who recalled seeing fake news stories believed them [5]. An analysis by Buzzfeed News shows that the top 20 false election stories from hoax websites generated nearly 1.5 million more user engagement activities on Facebook than the top 20 stories from reputable major news outlets [177]. So, there is an urgent call to develop effective strategies to mitigate the impact of fake news.

Policies to counter fake news can be categorized by the level of manual oversight and the aggressiveness of action required. Aggressively acting on fake news has various drawbacks. For example, Facebook's strategy allows users to report stories as potential fake news, sends these stories to fact-checking organizations, and flags them as disputed in users' newsfeed [143]. Such direct action on the offending news requires a high degree of human oversight, which can be costly and slow, and also may violate civil rights. The report-and-flag mechanism is also open to abuse by adversaries who maliciously report real news. Given these disadvantages, we consider an alternative strategy: optimizing the performance of real news propagation over the network, ensuring that people who are exposed to fake news are also exposed to real news, so that they are less likely to be convinced by fake news.

We face several key modeling and computational issues. For example, how to quantify the uncertainty of user activities and news propagation within the network? How to measure the effect of mitigation incentives and activities? Is it possible to steer the spontaneous user mitigation activities by an intervention strategy? To address these questions, we model the temporal randomness of fake news and mitigation events ("valid news") as multivariate point processes with self and mutual excitations, in which the control incentivizes more spontaneous mitigation events by contributing to the exogenous activity of campaigner nodes. The influence of fake news and mitigation activities is quantified using

event exposure counts (i.e. the number of times that a user is exposed to fake or real news posts from other users whom she follows).

Our key contributions are as follows. We present the first formulation of fake news mitigation as the problem of optimal point process intervention in a network. The goal is to optimize the activity policy of a set of campaigner nodes to mitigate a fake news process stemming from another set of nodes. This framework enables one to design a variety of objectives to quantify the meaning of "mitigation", such as minimizing the number of users who see fake news but were not reached by real news. We give the first derivation of second-order statistics of random exposure counts in the non-stationary case, which is essential in policy evaluation and improvement. By defining a state space for the network, formulating actions as exogenous intensity, and defining reward functions, we map the fake news mitigation problem to an optimal policy problem in a Markov decision process (MDP), which is solved by model-based least-squares temporal difference learning (LSTD) specific to the context of point processes. Furthermore, to the best of our knowledge, we are the first to conduct a real-time point process intervention experiment. Figure 5.1 shows the overall architecture.

## 5.2 Problem Formulation

**Network activities**. We model both fake news and mitigation processes as Multivariate Hawkes Process (MHP) over the network. Conceptually, MHP is a networked point process model with dependent dimensions (nodes), and can capture the underlying network structure and node interactions [22, 217, 87]. For example, an event by one user (a node) can trigger more events at other connected users. Define $F(t) = \big(F_1(t), \ldots, F_n(t)\big)^\top \in \mathbb{N}_0^n$, where $F_i(t)$ counts the number of times user $i$ shares a piece of news from the fake campaign up to time $t$. Similarly, define $M(t) = \big(M_1(t), \ldots, M_n(t)\big)^\top \in \mathbb{N}_0^n$ for the mitigation process. Correspondingly, we have 2 intensity functions: $\lambda^M(t) = (\lambda_1^M(t), \ldots, \lambda_n^M(t))^\top$ and $\lambda^F(t) = (\lambda_1^F(t), \ldots, \lambda_n^F(t))^\top$ and two sets of exogenous intensities $\mu^M$ and $\mu^F$.

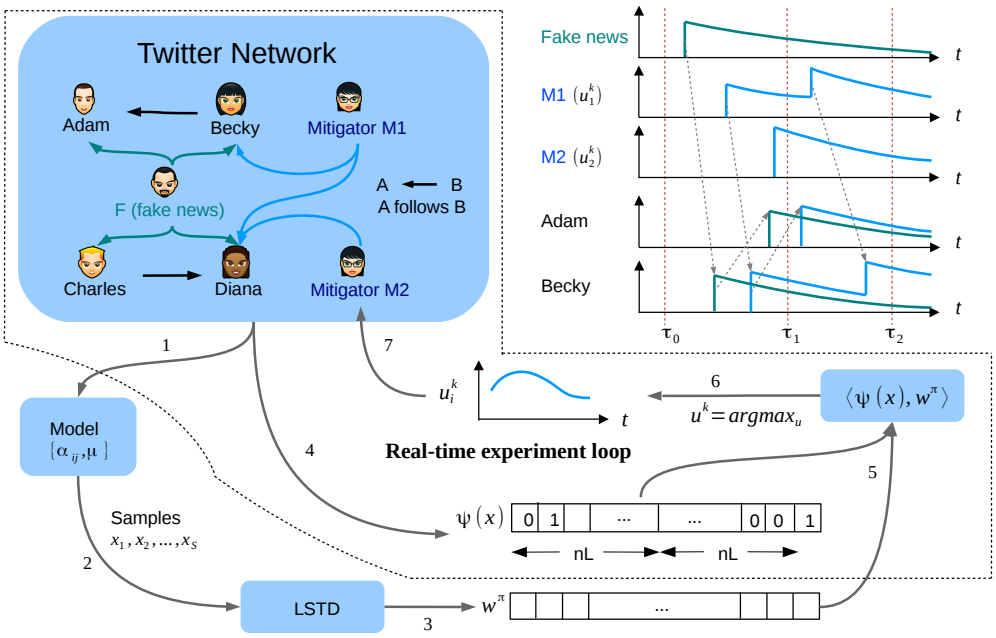



Figure 5.1: The framework of point process based intervention for countering fake news. (1-3) Offline learning of value function approximation weight vector using LSTD from transition samples generated from model. (4-7) Real-time intervention loop that uses feature representation of network state to choose optimal exogenous incentive for mitigator nodes.

**Goal**. Given that both $F(t)$ and $M(t)$ are modeled by the Hawkes processes, our goal is to find the optimal mitigation strategy that specifies how to adjust the exogenous intensity of a few mitigator nodes, such that an objective function (rigorously defined in sec. 5.3.1) can be maximized under budget constraints. To this end, we measure the influence of fake news and mitigation activities using event exposures, describe the mechanism of mitigation interventions, and quantify the effect of interventions mathematically.

**Event exposure**. Event exposure is a quantitative measure of campaign influence, and is represented as a counting process, $\mathcal{E}(t) = \big(\mathcal{E}_1(t), \ldots, \mathcal{E}_n(t)\big)^\top$. Here, $\mathcal{E}_i(t)$ records the number of times user $i$ is exposed to a campaign $N(t)$ by time $t$, where the exposure count increases whenever the user or a neighbor performs an activity. Let $B$ be the adjacency matrix of the user network, i.e., $b_{ij} = 1$ if user $i$ follows user $j$, and assume $b_{ii} = 1$ for all $i$. Then the exposure process is given by $\mathcal{E}(t) = BN(t)$. We define $\mathcal{F}(t) = BF(t)$

and $\mathcal{M}(t) = BM(t)$ as the fake news and mitigation exposure processes, respectively. Note that the MHP allows cascades of mutual excitations to occur among many nodes, so that non-adjacent users can also contribute to one another's exposure counts, if there is a directed path between them.

**Intervention**. To maximize objectives defined for fake news mitigation in section 5.3.1, suppose we can perform intervention by incentivizing a subset of users in the $k$-th stage during time $[\tau_k, \tau_{k+1})$ to trigger real news events. For simplicity, we consider uniform time duration $\tau_{k+1} - \tau_k = \Delta_T$ for $k = 0, 1, \ldots$, since generalization to nonuniform time durations is trivial. We model the incentive by a constant intervention $u_i^k \geq 0$ added to the exogenous intensity $\mu_i$ during time $[\tau_k, \tau_{k+1})$ for each stage $k = 0, 1, \ldots$. The mitigation activity intensity at the $k$-th stage is

$$\lambda^M(t) = \mu + u^k + \int_0^t \Phi(t-s)\,\mathrm{d}M(s), \tag{5.1}$$

for $t \in [\tau_k, \tau_{k+1})$. Note that the intervention itself exhibits a stochastic nature: adding $u_i^k$ to $\mu_i$ is equivalent to incentivizing user $i$ to increase her activity rate, but it is still uncertain when she will perform an activity, which appropriately mimics the randomness of the real world.

**Reward function**. For each stage $k$, let $x^k$ (defined in section 5.3.3) be the state of the whole MDP that encodes all the information from previous stages, and let $u^k$ be the current control imposed at this stage. Let $\mathcal{M}_i^k(t; x^k, u^k) := \sum_j b_{ij} \int_{\tau_k}^t \mathrm{d}M_j(s)$ be the number of times user $i$ is exposed to the mitigation campaign by time $t \in [\tau_k, \tau_{k+1})$ within stage $k$, and define $\mathcal{F}_i^k(t; x^k, u^k)$ similarly for the fake news exposure process. The reward function $R(x^k, u^k)$ can then be designed as a composite function of $\mathcal{M}$ and $\mathcal{F}$ (section 5.3.1).

**Problem statement**. By observing the counting process in previous stages (summarized in a sequence of $x^k$) and taking the future uncertainty into account, the control problem is to design a policy $\pi$ such that the controls $u^k = \pi(x^k)$ can maximize the total

discounted objective

$$\mathbb{E}[\sum_{k=0}^{\infty} \gamma^k R^k], \tag{5.2}$$

where $\gamma \in (0, 1]$ is the discount rate and $R^k$ is the observed reward at stage $k$. In addition, we may have constraints on the amount of control, such as a budget constraint on the sum of all interventions to users at each stage, or a cap over the amount of intensity a user can handle. A feasible set or an action space over which we find the best intervention is represented as

$$U_k := \left\{ u \in \mathbb{R}^n | u^\top c^k \leq C_k, 0 \leq u \leq \alpha^k \right\}. \tag{5.3}$$

Here, $c_i^k$ is the price per unit increase of exogenous intensity of user $i$ and $C_k \in \mathbb{R}_+$ is the total budget at stage $k$. Also, $\alpha_i^k$ is the cap on the amount of activities of the user $i$.

## 5.3 Proposed Method

In this section, we present the formulation of reward functions in terms of event exposures of fake news and mitigation activities. Then we derive the key statistics of the MHP required for reward function evaluation, followed by the policy iteration scheme to find the optimal intervention.

### 5.3.1 Fake News Mitigation

As we discussed above, the total reward of policy $\pi$ is defined by the value function

$$V^\pi(x^0) = \mathbb{E}\left[ \sum_{k=0}^{\infty} \gamma^k R^k \,\middle|\, x^0 \right] \tag{5.4}$$

for the initial state $x^0$ of fake and mitigation processes, where the observed reward $R$ quantifies the effect of mitigation activities $M(t)$ in each stage and $\gamma \in (0, 1]$ is the discount rate. We consider two types of reward functions $R(x, u)$:

1) *Correlation Maximization*: One possible way is to require correlation between mitigation exposures and fake news exposures: people exposed more to fake news should also be

exposed more to the true news, so that they are less likely to believe completely in fake news. Therefore, we can design the reward function $R$ in stage $k$ to be:

$$R(x^k, u^k) = \frac{1}{n}\mathcal{M}^k(\tau_{k+1}; x^k, u^k)^\top \mathcal{F}^k(\tau_{k+1}; x^k, u^k).$$

2) *Difference Minimization*: Suppose the goal is to minimize the number of unmitigated fake news events, then we can form a reward function $R$ in stage $k$ as the least squares of unmitigated numbers:

$$R(x^k, u^k) = \frac{-1}{n}\left\|\mathcal{M}^k(\tau_{k+1}; x^k, u^k) - \mathcal{F}^k(\tau_{k+1}; x^k, u^k)\right\|^2$$

These are two candidate reward functions in the MHP-MDP context, among others. To solve the policy optimization problem $\operatorname{argmax}_\pi V^\pi(x^0)$ for $V^\pi$ defined in (5.4), we need to evaluate the value function $V^\pi$ for any given policy $\pi$, which requires the first and second order statistics (moments) of any multivariate Hawkes processes $N(t)$, as we derive next.

### 5.3.2 Second Order Statistics of Non-stationary Process

For an $n$-dim MHP $N(t)$ with standard exponential kernel $\Phi(t)$, the following proposition provides closed-form solution of the mean intensity $\eta(t) := \mathbb{E}[\lambda(t)]$ for both constant and time-varying exogenous intensity $\mu(t)$:

**Proposition 10** (lemma 6 and theorem 4 )**.** *Let $N(t)$ be an $n$-dimensional MHP defined in sec. 5.2 with exogenous intensity $\mu(t)$ and Hawkes kernel $\Phi(t) = A\mathrm{e}^{-\omega t}h(t)$, then the mean intensity $\eta(t)$ is given by*

$$\eta(t) = \left[I + A(A - \omega I)^{-1}\left(\mathrm{e}^{(A-\omega I)t} - I\right)\right]\mu(t). \tag{5.5}$$

Let $\Lambda(t) = \int_0^t \lambda(s)\,\mathrm{d}s$ be the compensator of $N(t)$, then by the Doob-Meyer decomposition theorem, $N(t)-\Lambda(t)$ is a zero mean martingale. This implies that the first order statistics $\mathbb{E}[N(t)]$ can be obtained by $\mathbb{E}[N(t)] = \mathbb{E}[\Lambda(t)] = \mathbb{E}[\int_0^t \lambda(s)\,\mathrm{d}s] = \int_0^t \mathbb{E}[\lambda(s)]\,\mathrm{d}s = \int_0^t \eta(s)\,\mathrm{d}s$ using eq. (5.5).

To evaluate the reward function $R$ defined previously, we need to derive second order

statistics of multivariate Hawkes process $N(t)$ in its non-stationary stage. The following theorem states the key ingredients for the second order statistics.

**Theorem 11.** *Let $N(t)$ be an $n$-dim MHP with exogenous intensity $\mu$ and Hawkes kernel $\Phi$ defined in sec. 5.2, then the second order statistics of $N(t)$ for $t, t' \geq 0$ is given by*

$$\begin{aligned}\mathbb{E}\left[\mathrm{d}N(t)\,\mathrm{d}N(t')^\top\right] = G(t',t)^\top \Sigma(t')\,\mathrm{d}t\,\mathrm{d}t' + \\ \delta(t-t')\Sigma(t')\,\mathrm{d}t\,\mathrm{d}t' + \eta(t)\eta(t')^\top\,\mathrm{d}t\,\mathrm{d}t'\end{aligned} \tag{5.6}$$

*where $\eta(t) = \mathbb{E}[\lambda(t)]$ is given in* (5.5)*, $\Sigma(t) = \mathrm{diag}([\eta_i(t)])$ is diagonal, and $G$ is the unique solution of*

$$G(t',t) = G(t',t) * \Phi(t) + \Phi(t-t') - \delta(t-t')I. \tag{5.7}$$

*Moreover $G(t',t)^\top \Sigma(t') = \Sigma(t) G(t,t')$ for all $t, t' \geq 0$.*

*Proof.* Fix node index $j$ and $t' \geq 0$, define $g_{ji}(t',t)$ for all node $i$ and $t$ such that

$$g_{ji}(t',t)\,\mathrm{d}t = \mathbb{E}\left[\mathrm{d}N_i(t) \middle|\, \mathrm{d}N_j(t') = 1\right] - \delta_{ij}\delta(t-t')\,\mathrm{d}t - \eta_i(t)\,\mathrm{d}t \tag{5.8}$$

Since the conditional intensity of $N_i(t)$ is $\lambda_i(t)$, we have

$$\begin{aligned} g_{ji}(t',t)\,\mathrm{d}t &= \mathbb{E}\left[\mathrm{d}N_i(t) \middle|\, \mathrm{d}N_j(t') = 1\right] - \delta_{ij}\delta(t-t')\,\mathrm{d}t - \eta_i(t)\,\mathrm{d}t \\ &= \mathbb{E}\left[\lambda_i(t) \middle|\, \mathrm{d}N_j(t') = 1\right]\mathrm{d}t - \delta_{ij}\delta(t-t')\,\mathrm{d}t - \eta_i(t)\,\mathrm{d}t \end{aligned}$$

Furthermore, we have $\lambda_i(t) = \mu_i(t) + \sum_{k=1}^n \int_0^t \phi_{ki}(t-s)\,\mathrm{d}N_k(s)$ and hence

$$\begin{aligned} \mathbb{E}\left[\lambda_i(t) \middle|\, \mathrm{d}N_j(t') = 1\right] &= \mu_i(t) + \sum_{k=1}^n \int_0^t \phi_{ki}(t-s)\mathbb{E}[\mathrm{d}N_k(s) \mid \mathrm{d}N_j(t') = 1] \\ &= \mu_i(t) + \sum_{k=1}^n \int_0^t \phi_{ki}(t-s)\left[g_{jk}(t',s)\,\mathrm{d}s + \delta_{kj}\delta(s-t')\,\mathrm{d}s + \eta_k(s)\,\mathrm{d}s\right] \\ &= \mu_i(t) + \sum_{k=1}^n \int_0^t \phi_{ki}(t-s) g_{jk}(t',s)\,\mathrm{d}s + \phi_{ji}(t-t') \\ &\quad + \sum_{k=1}^n \int_0^t \phi_{ki}(t-s)\eta_k(s)\,\mathrm{d}s \end{aligned}$$

where we applied the definition of $g_{jk}$ in (5.8) to obtain the second equality. Combining the two equations above and using the fact that $\eta_i(t) = \mu_i(t) + \sum_{k=1}^n \int_0^t \phi_{ki}(t-s)\eta_k(s)\,\mathrm{d}s$,

we obtain that

$$g_{ji}(t',t) = \sum_{k=1}^{n} \int_0^t \phi_{ki}(t-s) g_{jk}(t',s)\,\mathrm{d}s + \phi_{ji}(t-t') - \delta_{ij}\delta(t-t')$$

Since $j$ and $t'$ are arbitrary, we let $G(t',t)$ be the matrix such that the $(j,i)$-th entry of $G(t',t)$ is $g_{ji}(t',t)$, then we have

$$G(t',t) = G(t',t) * \Phi(t) + \Phi(t-t') - \delta(t-t')I \tag{5.9}$$

Note that the Wiener-Hopf equation (5.9) determines the unique solution $G(t',t)$ for all $t \geq t'$. Moreover, since MHP is simple and that $\mathrm{d}N_i(t) = 0$ or $1$ a.s. for all $i$, we have

$$\begin{aligned}
\mathbb{E}[\mathrm{d}N_i(t)\,\mathrm{d}N_j(t')] &= \Pr\left(\mathrm{d}N_i(t) = 1, \mathrm{d}N_j(t') = 1\right) \\
&= \Pr\left(\mathrm{d}N_i(t) | \,\mathrm{d}N_j(t') = 1\right) \Pr\left(\mathrm{d}N_j(t') = 1\right) \\
&= \mathbb{E}[\mathrm{d}N_i(t) | \,\mathrm{d}N_j(t) = 1]\mathbb{E}[\mathrm{d}N_j(t')] \\
&= \mathbb{E}[\mathrm{d}N_i(t) | \,\mathrm{d}N_j(t) = 1]\mathbb{E}[\lambda_j(t')]\,\mathrm{d}t' \\
&= \mathbb{E}[\mathrm{d}N_i(t) | \,\mathrm{d}N_j(t) = 1]\eta_j(t')\,\mathrm{d}t' \\
&= g_{ji}(t',t)\eta_j(t')\,\mathrm{d}t\,\mathrm{d}t' + \delta_{ij}\delta(t-t')\eta_j(t')\,\mathrm{d}t\,\mathrm{d}t' + \eta_i(t)\eta_j(t')\,\mathrm{d}t\,\mathrm{d}t'
\end{aligned} \tag{5.10}$$

Similarly, we can switch $i$ and $j$, and $t$ and $t'$ to obtain

$$\mathbb{E}[\mathrm{d}N_i(t)\,\mathrm{d}N_j(t')] = g_{ij}(t,t')\eta_i(t)\,\mathrm{d}t\,\mathrm{d}t' + \delta_{ij}\delta(t-t')\eta_i(t)\,\mathrm{d}t\,\mathrm{d}t' + \eta_i(t)\eta_j(t')\,\mathrm{d}t\,\mathrm{d}t' \tag{5.11}$$

Combining (5.10) and (5.11) we have that

$$g_{ji}(t',t)\eta_j(t') = g_{ij}(t,t')\eta_i(t)$$

i.e., $G(t',t)^\top \Sigma(t') = \Sigma(t)G(t,t')$, from which $G(t',t)$ for $t < t'$ is also uniquely determined. We therefore have

$$\mathbb{E}\left[\mathrm{d}N(t)\,\mathrm{d}N(t')^\top\right] = G(t',t)^\top \Sigma(t')\,\mathrm{d}t\,\mathrm{d}t' + \delta(t-t')\Sigma(t')\,\mathrm{d}t\,\mathrm{d}t' + \eta(t)\eta(t')^\top\,\mathrm{d}t\,\mathrm{d}t' \tag{5.12}$$

This completes the proof. □

Based on Theorem 11, we can compute second order statistics such as $\mathbb{E}[N_i(t)N_j(t')]$ for all $i, j$ and $t, t' \geq 0$.

### 5.3.3 State Representation

Hawkes process is non-Markovian and one needs complete knowledge of the history to characterize the entire process. However, when the standard exponential kernel $\Phi(t,s) = Ae^{-\omega(t-s)}h(t-s)$ is employed, the effect of history up to time $\tau_k$ on the future $t > \tau_k$ can be cleverly summarized by one scalar per dimension [178, 60]. For $1 \leq i \leq n$, define $y_i^k := \lambda_i^{k-1}(\tau_k) - u_i^{k-1} - \mu_i$, (and $y_0^i = 0$ by convention), then the intensity due to events of all previous $k$ stages can be written as $\int_0^{\tau_k} Ae^{-\omega(t-s)}\,\mathrm{d}N(s) = y^k e^{-\omega(t-\tau_k)}$. In other words, $y^k$ is sufficient to encode the information of activities in the past $k$ stages that are relevant to future. Note that we have two separate $y_M^k$ and $y_F^k$ to track the dynamics of both mitigation and fake processes.

In order to tackle objectives over multiple stages, we add aggregated number of events at $L$ previous $\Delta_f$-time intervals over all dimensions. Define a vector $z^k \in \mathbb{R}^{nL}$ where $z_{(l-1)n+i}^k = \int_{\tau_k - l\Delta_f}^{\tau_k - (l-1)\Delta_f} \mathrm{d}N_i(s)$ for $1 \leq i \leq n$ and $1 \leq l \leq L$. In other words, $z_{(l-1)n+i}^k$ records the number of events of $i$-th dimension in the $l$-th interval of length $\Delta_f$ prior to time $\tau_k$. For example, choosing $\Delta_f = \Delta_T$ and setting $L = 2$ means that events from the two most recent stages are counted. Similarly, we have two separate $z_M^k$ and $z_F^k$ corresponding to the two processes. Now, the state vector $x^k \in \mathbb{R}^{2nL+2n}$ is the concatenation of the above four vectors $x^k = [y_M^k; y_F^k; z_M^k; z_F^k]$.

### 5.3.4 Least Squares Temporal Difference

The optimal value function satisfies the Bellman equation:

$$V^\pi(x) = \mathbb{E}[R(x, \pi(x))] + \gamma \mathbb{E}[V^\pi(x')], \tag{5.13}$$

where $x'$ is the next state after taking action based on policy $\pi$ at state $x$. Least squares temporal difference learning (LSTD) is a sample-efficient procedure for policy evaluation, which subsequently facilitates policy improvement. The value function is approximated by $\hat{V}^\pi(x) = \sum_{d=1}^D w_d^\pi \psi_d(x)$, where $\psi_d$ is the $d$-th feature of state $x$ and $w_d^\pi$ is its coef-

**Algorithm 7** LSTD policy iteration in point processes

**Input:** set of samples $\mathcal{S}$, feature $\psi(\cdot)$, discount $\gamma$
**repeat**
  Initialize $A^\pi = 0$ and $b^\pi = 0$.
  **for each** state $x \in \mathcal{S}$ **do**
    $A^\pi \leftarrow A^\pi + \psi(x)(\psi(x) - \gamma\psi(x'))^\top$
    $b^\pi \leftarrow b^\pi + \psi(x)r^\pi$
  **end for**
  $w^\pi \leftarrow (A^\pi)^{-1}b^\pi$
  **for each** state $x \in \mathcal{S}$ **do**
    $\pi(x) \leftarrow \operatorname{argmax}_u \{\mathbb{E}[R(x,u)] + \gamma\mathbb{E}[V^\pi(x')|u, w^\pi]\}$
  **end for**
**until** $\|\Delta w^\pi\| < 0.1$
**return** $w^\pi$

ficient for policy $\pi$. This can be compactly represented as $\hat{V}^\pi(x) = \psi(x)^\top w^\pi$, where $\psi(x) = (\psi_1(x), \ldots, \psi_D(x))^\top$. The following presents our choice of features and the policy evaluation and improvement steps of LSTD(0) [185].

**Features**. The number of events in a few recent consecutive intervals of point processes have been used as a reliable feature to parameterize point processes [150, 156, 130]. Following their work we take $L$ prior intervals of length $\Delta_f$ for each dimension of the fake news process and record the number of events in that period as one feature. $\psi^k_{(l-1)n+i} = z^k_{(l-1)n+i}$ for $1 \le i \le n$ and $1 \le l \le L$. This will count for $nL$ features. Similarly we take $nL$ features from the mitigation process. Finally, we add a last feature $\psi^k_{2nL+1} = 1$ as the bias term. Therefore, $\psi^k = [z^k_M; z^k_F; 1]$ and the feature space has dimension $D = 2nL + 1$.

**Policy Evaluation.** Substituting the approximation into the Bellman equation, we have:

$$\psi(x)^\top w^\pi = \mathbb{E}[R(x, \pi(x))] + \gamma\mathbb{E}[\psi(x')^\top]w^\pi. \tag{5.14}$$

To find the best fit of $w^\pi$ we have to consider all possible $x$; however, since the state space is infinite-dimensional, enumerating all states is impossible and we utilize a set $\mathcal{S}$ of samples $\mathcal{S} = \{x_1, \ldots, x_S\}$.

Let $\psi(x_s) = \psi_s \in \mathbb{R}^D$, $\mathbb{E}[\psi(x'_s)] = \psi'_s \in \mathbb{R}^D$, and $r^\pi_s = \mathbb{E}[R(x_s, \pi(x_s))] \in \mathbb{R}$.

Then define matrices of current features $\Psi = [\psi_1^\top; \ldots; \psi_S^\top]^\top \in \mathbb{R}^{S \times D}$ and next features $\Psi' = [\psi'^\top_1; \ldots; \psi'^\top_S]^\top \in \mathbb{R}^{S \times D}$, the rewards $r^\pi = [r_1^\pi, \ldots, r_S^\pi]^\top \in \mathbb{R}^S$, and the sample value functions as $v^\pi = [V^\pi(x_1), \ldots, V^\pi(x_S)]^\top \in \mathbb{R}^S$. Given the above definition, the Bellman optimality of eq. (5.14) can be written in matrix format:

$$v^\pi = \Psi w^\pi = r^\pi + \gamma \Psi' w^\pi \triangleq T^\pi v^\pi, \tag{5.15}$$

where $T^\pi$ is the Bellman optimality operator. A way to find a good estimate is to force the approximate value function to be a fixed point of the optimality equation under the Bellman operator, *i.e.*, $T^\pi \hat{v}^\pi \approx \hat{v}^\pi$. [117]. For that, the fixed point has to lie in the space of approximate value functions, spanned by the basis functions $\Psi$. $\hat{v}^\pi$ lies in that space by definition, but $T^\pi \hat{v}^\pi$ may have an orthogonal component and must be projected. This is achieved by the orthogonal projection operator $(\Psi(\Psi^\top \Psi)^{-1} \Psi^\top)$. Therefore the approximate value function $\hat{v}^\pi$ must be invariant under one application of the Bellman operator $T^\pi$ followed by orthogonal projection:

$$\hat{v}^\pi = \Psi(\Psi^\top \Psi)^{-1} \Psi^\top (T^\pi \hat{v}^\pi). \tag{5.16}$$

By substituting the linear approximation $\Psi w^\pi = v^\pi$ into the above equation and some manipulations, we get a $D \times D$ linear systems of equations $A^\pi \omega^\pi = b^\pi$, where $A^\pi = \Psi^\top(\Psi - \gamma \Psi')$ and $b^\pi = \Psi^\top r^\pi$, and whose solution is the fitted coefficients $w^\pi$. It has been shown that the estimated $w^\pi$ converges to the best $w^*$ as the available number of samples tends to infinity [26]. Details are as follows.

We seek an approximate value function $\hat{v}^\pi$ that is invariant under one application of the Bellman operator $T^\pi$ followed by orthogonal projection:

$$\hat{v}^\pi = \Psi(\Psi^\top \Psi)^{-1} \Psi^\top (T^\pi \hat{v}^\pi) \tag{5.17}$$

By replacing the linear approximation, $\Psi w^\pi = v^\pi$ , and some manipulations we get:

$$\begin{aligned}
\Psi(\Psi^\top\Psi)^{-1}\Psi^\top(r^\pi + \gamma\Psi' w^\pi) &= \Psi w^\pi \\
\Psi\left((\Psi^\top\Psi)^{-1}\Psi^\top(r^\pi + \gamma\Psi' w^\pi) - w^\pi\right) &= 0 \\
(\Psi^\top\Psi)^{-1}\Psi^\top(r^\pi + \gamma\Psi' w^\pi) - w^\pi &= 0 \\
(\Psi^\top\Psi)^{-1}\Psi^\top(r^\pi + \gamma\Psi' w^\pi) &= w^\pi \\
\Psi^\top(r^\pi + \gamma\Psi' w^\pi) &= (\Psi^\top\Psi)w^\pi \\
\underbrace{\Psi^\top(\Psi - \gamma\Psi')}_{D\times D} w^\pi &= \underbrace{\Psi^\top r^\pi}_{D\times 1}
\end{aligned}$$

Defining $A^\pi = \Psi^\top(\Psi - \gamma\Psi')$ and $b^\pi = \Psi^\top r^\pi$ the estimated coefficients are the solution of a $D \times D$ linear systems of equation: $A^\pi\omega^\pi = b^\pi$.

**Policy Improvement.** The second part of the algorithm implements policy improvement, i.e., getting an improved policy $\pi'$ via one-step look-ahead as follows:

$$\pi'(x) = \underset{u}{\operatorname{argmax}}\, \mathbb{E}[R(x,u) + \gamma V^\pi(x')]. \tag{5.18}$$

LSTD(0) alternates between the policy improvement and policy evaluation iteratively until $w^\pi$ converges [26]. Algorithm 7 summarizes this procedure.

**LSTD in Hawkes context**. LSTD is particularly suitable to the problem we are interested in. It learns the value function $V^\pi(x)$, and as such, policy improvement can be challenging without knowing the model. Because of this, methods that aim to learn the Q-function $Q^\pi(x,u)$, such as LSPI [117], are widely applied. The downside of Q-function based methods is that they typically require more samples than value-function based methods, because they require a large collection of state-action pairs $\{(s,a)\}$ for sufficient exploration of both state and action spaces. Yet, in our setup, learning the value function is sufficient, by writing the action-value function as $Q^\pi(x,u) = \mathbb{E}[R(x,u) + V^\pi(x')]$, and observing that the learned model of the multivariate Hawkes process enables analytical

computation of the expectation

$$\mathbb{E}[V^{\pi}(x')] = \sum_{i=1}^{n}\sum_{l=1}^{L-1} w^{\pi}_{ln+i} z^{k-1}_{M,(l-1)n+i} + w^{\pi}_{nL+ln+i} z^{k-1}_{F,(l-1)n+i}$$
$$+ \sum_{i=1}^{n} w_i \mathbb{E}[z^k_{M,i}] + w_{nL+i}\mathbb{E}[z^k_{F,i}] + w^{\pi}_{2nL+1},$$
$$\mathbb{E}[R(x,u)] = \frac{1}{n}\mathbb{E}[z^k_M]^{\top} B^{\top} B\, \mathbb{E}[z^k_F], \qquad \% \text{ correlation}$$
$$\mathbb{E}[R(x,u)] = -\frac{1}{n}\mathbb{E}[{z^k_M}^{\top} B^{\top} B\, z^k_M] - \frac{1}{n}\mathbb{E}[{z^k_F}^{\top} B^{\top} B\, z^k_F]$$
$$+ \frac{2}{n}\mathbb{E}[z^k_M]^{\top} B^{\top} B\, \mathbb{E}[z^k_F]. \qquad \% \text{ difference}$$

The above are derived as a result of the following computations. Assume we are at the beginning of stage $k$. The expected feature vector for the next state $x'$ is comprised of $L$ intervals per process, out of which $L-1$ are observed. Only the most recent interval is not observed and needs to be re-evaluated in expectation sense. To compute $\mathbb{E}[V^{\pi}(x')]$ have:

$$\mathbb{E}[V^{\pi}(x')] = \mathbb{E}[\sum_{d=1}^{D} w^{\pi}_d \psi_d(x')] \tag{5.19}$$
$$= \mathbb{E}[\sum_{i=1\ldots n, l=1\ldots L} w^{\pi}_{(l-1)n+i} z^k_{M,(l-1)n+i} \tag{5.20}$$
$$+ w^{\pi}_{nL+(l-1)n+i} z^k_{F,(l-1)n+i} + w^{\pi}_{2nL+1}] \tag{5.21}$$
$$= \sum_{i=1\ldots n, l=1\ldots L-1} w^{\pi}_{ln+i} z^{k-1}_{M,(l-1)n+i} + w^{\pi}_{nL+ln+i} z^{k-1}_{F,(l-1)n+i} \tag{5.22}$$
$$+ \sum_{i=1\ldots n} w_i \mathbb{E}[z^k_{M,i}] + w_{nL+i}\mathbb{E}[z^k_{F,i}] + w^{\pi}_{2nL+1} \tag{5.23}$$

Then, following chapter 4, we obtain

$$\mathbb{E}[z^k_M] = \Gamma\,(\mu^M + u^k) + \Upsilon\, y^k_M \tag{5.24}$$
$$\mathbb{E}[z^k_F] = \Gamma\,\mu^F + \Upsilon\, y^k_M \tag{5.25}$$

where

$$\Upsilon = (A - \omega I)^{-1}(e^{(A-\omega I)\Delta} - I) \tag{5.26}$$
$$\Gamma = I\Delta + A(A-\omega I)^{-1}(\Upsilon - I\Delta); \tag{5.27}$$

To find $\mathbb{E}[R(x,u)]$ for the two different reward functions we have defined

- Correlation Maximization

$$\begin{aligned}
\mathbb{E}[R(x^k,u^k)] =& \frac{1}{n}\mathbb{E}[\mathcal{M}^k(\tau_{k+1};x^k,u^k)^\top \mathcal{F}^k(\tau_{k+1};x^k,u^k) && (5.28)\\
=& \frac{1}{n}\mathbb{E}[{z_M^k}^\top B^\top B\, z_F^k] && (5.29)\\
=& \frac{1}{n}\mathbb{E}[z_M^k]^\top B^\top B\, \mathbb{E}[z_F^k] && (5.30)\\
=& \frac{1}{n}\left(\Gamma(\mu^M+u^k)+\Upsilon y_M^k\right)^\top B^\top B\left(\Gamma\mu^F+\Upsilon y_F^k\right) && (5.31)
\end{aligned}$$

  Here the second line is due to the fact that mitigation campaign and fake news process are independent of each other (given the network model). Note the linear dependence of the the objective on our intervention $u^k$ which combined with linear constraints result in a convex optimization problem.

- Difference Minimization

$$\begin{aligned}
\mathbb{E}[R(x^k,u^k)] =& -\frac{1}{n}\mathbb{E}[\left\|\mathcal{M}^k(\tau_{k+1};x^k,u^k)-\mathcal{F}^k(\tau_{k+1};x^k,u^k)\right\|^2] && (5.32)\\
=& -\frac{1}{n}\mathbb{E}[(Bz_M^k - B\,z_F^k)^\top (Bz_M^k - B\,z_F^k)] && (5.33)\\
=& -\frac{1}{n}\underbrace{\mathbb{E}[{z_M^k}^\top B^\top B\, z_M^k]}_{\text{Second order moments}} + \frac{2}{n}\underbrace{\mathbb{E}[z_M^k]^\top B^\top B\,\mathbb{E}[z_F^k]}_{\text{First order moments}} - \frac{1}{n}\underbrace{\mathbb{E}[{z_F^k}^\top B^\top B\, z_F^k]}_{\text{Second order moments}} && (5.34)
\end{aligned}$$

  The first and second order moments are computed by [55] and Theorem 11, respectively.

We require much fewer samples to learn $V^\pi(x)$ compared to learning an approximate $Q^\pi(x,u)$, and in particular compared to LSPI, we avoid explicitly discretizing the continuous action space from which the action $u$ is chosen.

We further remark that the policy improvement step finds the optimal action $u$ at any state $x$ by computing $\operatorname{argmax}_u \mathbb{E}[R(x,u)+V^\pi(x')]$, where the action $u$ to be optimized appears in the calculation of both the expected current reward and the expected value at the

**Algorithm 8** Real-time fake news mitigation

**Input:** network $A$, learned $w^\pi$, feature $\psi(\cdot)$, discount $\gamma$
**repeat**
  Observe state $x$ of the network activities
  $u = \text{argmax}_a\{\mathbb{E}[R(x,a)] + \gamma\mathbb{E}[V^\pi(x')|a, w^\pi]\}$
  Add $u$ to base exogenous intensity $\mu$ and generate mitigation event times $\{t_i\}$ using point process model
  Create posts at times $\{t_i\}$ using campaigner accounts
**until** end of campaign

next state. This optimization problem is convex under our choice of reward functions and the form of the Hawkes conditional intensity.

After learning the optimal policy (implicitly defined by $w^\pi$ of the linearly-approximated value function) we start at the real-time intervention part. Given a state observation, we find the optimal intervention intensity by solving eq. (5.18). Algorithm 8 summarizes the real-time mitigation procedure.

### 5.4 Experiments

We evaluated our fake news mitigation framework by both simulated and real-time real-world experiments and show that our approach significantly outperforms several state-of-the-art methods and alternatives. First we verify the theoretical second order statistics in Figure 5.2. Then we introduce the baseline methods against which we compare our proposed approach, and present the results of synthetic and real intervention experiments. The measure of performance for all methods was how much total reward could be accumulated by each method, where the reward function is defined via the objective functions in section 5.3.1. We conclude by examining convergence properties and representative power of the chosen linear features in section 5.3.4.

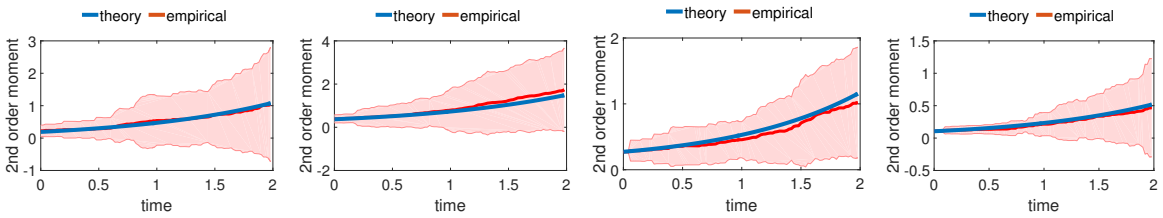



Figure 5.2: Empirical and theoretical second order moments of a Hawkes process, $\mathbb{E}[\mathrm{d}N_i(t)\,\mathrm{d}N_j(t')]$ for 4 random pairs $(i, j)$ and $t' = 0$ and varying $t$ from 0 to 2.

### 5.4.1 Empirical Validation of Second Order Statistics

In this section, we empirically study the theoretical results of section 5.3.2. The mean and standard deviation of the empirical second order statistics averaged over 100 simulations was compared to the theoretical mean. This experiment helps to verify that it can be used in simulations to evaluate the merits of our proposed algorithm and versus the baselines. Figure 5.2 demonstrates the second order correlation profile of 4 random pairs of users simulated 100 times. We see that the empirical average almost matches the theoretical average. Furthermore, it is interesting to see that the standard deviation increases with time. This is due to the aggregation of more random elements as time passes. Therefore, one should be careful with using the empirical mean without a sufficient number of random runs when the time interval is large.

### 5.4.2 Baselines

We compared our Least-squares Temporal Difference (**LTD**) intervention procedure with the following baselines. Sections 5.4.3 and 5.4.4 present experimental procedures and results.

1) **CEC** [60]: This is a recent network intervention algorithm based on point processes. It formulates a dynamic programming problem,

$$V^k(x) = \max_{u^k} \mathbb{E}[R(x^k, u^k) + \gamma V^{\pi}(x^{k+1})], \tag{5.35}$$

and uses approximate look-ahead dynamic programming implemented via Certainly Equivalence Control (CEC) [16] to find the optimum intervention.

2) **OPL** [55]: An open loop dynamic programming control based on convex optimization that finds the best intervention for all stages in one shot:

$$\max_{u^1,u^2,\ldots,u^K} \mathbb{E}[\sum_k \gamma^k R(x^k, u^k)]. \tag{5.36}$$

Open Loop (OPL) is an important baseline, because comparing it against closed-loop strategies like CEC and LTD indicates how much feedback information helps improve future decisions. It quantifies the *value of information* in the context of dynamic programming and optimal control.

3) **CLS**: For each node $i$ belonging to the mitigation campaign, we compute its closeness centrality $cent_i = \frac{1}{\sum_j dis(i,j)}$, where $dis$ is the shortest distance from $i$ to $j$. Then, we assign the budget such that $u_i \propto cent_i$, meaning that budget is distributed to mitigation nodes based on their proximity to nodes according to network structure. Closeness Centrality (CLS) has been widely used in the literature as a baseline for finding influential nodes [32, 42, 62].

4) **EXP**: The CLS baseline is only structural and does not use the fake news infection data. EXP augments it by computing an Exposure-based Closeness Centrality, $cent_i^k = \sum_j \frac{\sum_{l=1}^{L} \mathcal{F}_j^{k-l}}{dis(i,j)}$, where the numerator is the total number of times node $j$ has been exposed to the fake news campaign in the $L$ intervals before stage $k$. The more times node $j$ has been exposed to the fake news, the more important it is for the mitigation campaign to reach it. EXP assigns the budget according to $u_i^k \propto cent_i^k$.

5) **RND**: This policy assigns a random solution in the convex space of feasible interventions. It serves as a baseline and improvement over this random policy makes comparison feasible across different settings.

### 5.4.3 Synthetic Experiments

**Setup.** For all except the experiment over network size, the networks were generated synthetically with $n = 300$ nodes. Endogenous intensity coefficients were set as $a_{ij} \sim \mathcal{U}[0, 0.5]$. To mimic real world networks, sparsity was set to $0.02$, *i.e.*, each edge was kept with probability 0.02. The influence matrix was scaled appropriately such that the spectral radius is a random number smaller than one to ensure the stability of the process. The Hawkes kernel parameter was set to $\omega = 1$, which means loosing roughly 63 % of influence after 1 time unit (minutes, hours, etc). Both fake news and mitigation processes obey these network settings. Among $n$ nodes, we assume 20 nodes create fake news and another 20 nodes can be incentivized (via the exogenous intensity) to spread true news. Each stage has length of $\Delta_T = 1$. The discount factor was set to $\gamma = 0.7$. For determining features, we set $L = 2$ and we choose $\Delta_f = \Delta_T$ for simplicity. The upper bound for the intervention intensity was chosen by $\alpha_i \sim \mathcal{U}[0, 0.5]$. The price of each person was $c_i^k = 1$, and the total budget at stage $k$ was randomly generated as $C_k \sim (n \times \mathcal{U}[0, 0.5])$. 1000 randomly sampled states were used for the LSTD algorithm. To evaluate a policy (learned by an algorithm) we simulated the network under that policy 50 times and took the discounted total reward averaged over these 50 runs as an empirical valuation of the policy. Furthermore, each single run was simulated for 10 consecutive stages; from the eleventh stage onward, the objectives contribute 0.02 of the total reward and can be discarded. For all experiments, the above settings are assumed unless it is explicitly mentioned otherwise.

**Intervention results.** Figure 5.3 demonstrates the performance of different methods. Performance of a policy is quantified as the ratio of the total reward achieved by running the policy, over the total reward achieved by the random policy (RND). This allows us to compare the effectiveness of the algorithms over a variety of settings. All the results reported are averages over 10 runs with random networks generated according to the above setup. Overall, it is clear that LTD is almost consistently the best. It improves over the

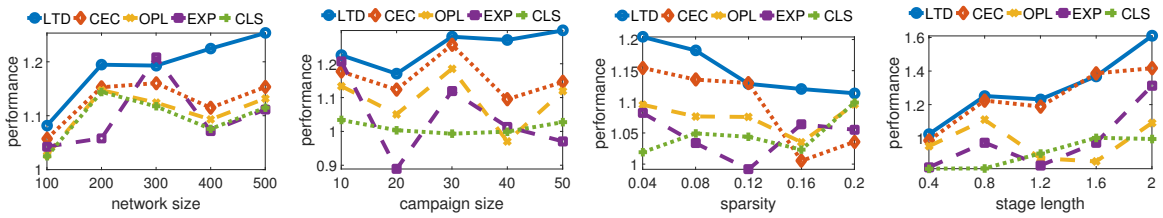



Figure 5.3: Performance improvement of different methods over the random policy on synthetic networks for correlation maximization

random policy by roughly 20 percent. CEC is the second best and shows the effectiveness of multi-stage and closed loop intervention. This validates our intuition that although CEC computes the reward from both fake news and mitigation processes, the lack of explicit features corresponding to previous events in its value function prevents it from learning the reason for the reward. Roughly, OPL is the third best algorithm, due to its negligence of the state and the actual events that occurred. Next, comes the EXP algorithm followed by the CLS. The poor performance of these (compared to others) shows that structural properties are not sufficient to tackle the fake news mitigation problem. EXP is roughly better than CEC because it heuristically takes into account the fake news exposure.

Figure 5.3-a shows the performance with respect to increasing network size. The difference between alternative methods and the gap between LTD and others increase with the network size. Furthermore, the performance of all methods show an increase over random policy when the problem size gets larger. This illustrates the fact that efficient distribution of budget matters more when confronted with problems of increasing complexity and size.

Figure 5.3-b shows the performance with respect to increasing the mitigation campaign size. Larger campaigns imply greater flexibility of intervention, which can be exploited by clever algorithms to achieve higher performance.

Figure 5.3-c shows the performance with respect to increasing sparsity of the network. Interestingly, the performance of all the algorithms move towards to the random policy as the network becomes denser. This can be understood by considering a complete graph,

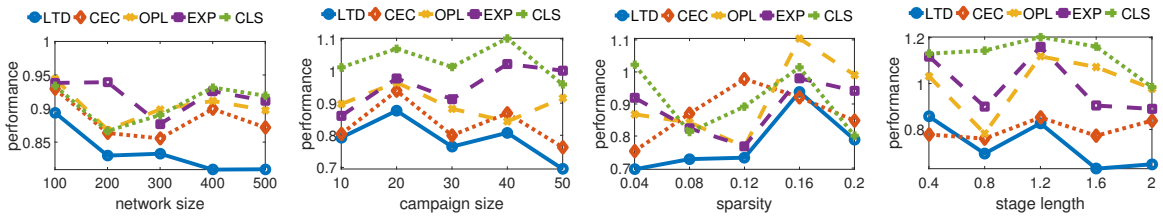



Figure 5.4: Performance improvement of different methods over the random policy on synthetic networks for distance minimization

so that no matter how and to whom we distribute the mitigation budget, all the nodes are exposed to the mitigation campaign almost equally. However, since real social networks are usually sparse, the effectiveness of the proposed method stands out.

Finally, Figure 5.3-d shows the performance with respect to the length of an stage. Longer stage lengths increase the potential for a good policy to attain higher reward than a random policy, and this is reflected by the sharp increase and larger performance gap between LTD and others for longer lengths. We observe the same patterns for the distance minimization in Figure 5.4 problem and avoid repeating them.

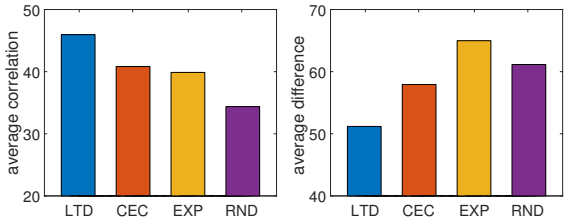
### 5.4.4 Real Experiments

In this section we explain our real-time intervention results. To the best of our knowledge, we are the first to employ a real-time experiment to evaluate a point process based social network intervention strategy.

**Setup.** Using five Twitter accounts, each of which made five posts on machine learning topics at random times per day for a span of two months (Nov.-Dec. 2016), we accumulated a network of 1894 real users with 23407 directed edges in total. We used this historical data to learn the network parameters $\{\alpha_{ij}, \mu^i\}$ using maximum likelihood (similar to related work [228, 55]) with one hour as the time resolution and the kernel decay parameter $\omega$ set to 0.1. As illustrated in Figure 5.1 the optimal policy was learned using LSTD and policy improvement. Then the real-time experiment starts: Two of the accounts, interpreted as

the source of fake news, continued to behave using the same randomized policy as they did in the data collection stage, while the posting times of the other three accounts were generated from $(u_1, u_2, u_3)^T$, produced by our LTD strategy or a competitor strategy. Each policy was run for 10 stages of length 12 hours. Therefore, $\Delta_T = \Delta_f = 12$. Since both fake news and mitigation accounts were tweeting random posts on machine learning, we assume negligible bias in the content that can confound the performance. At the end of each stage, all retweets–by users within the network–of the posts made during the two most recent stages were used to construct the feature vector and compute the value function, which was used to find the optimal intervention for the next stage. The methods CEC and OPL belong to the same category, and it has been shown that CEC outperforms OPL in [60]. Furthermore, EXP and CLS also belong to similar families and our synthetic experiments confirm the superiority of the former. So, to save time in real interventions, we only test CEC from the first and EXP from the second pair, and compare them with the random policy (RND) and with our algorithm (LTD).

**Real-time intervention results.** Figure 5.5 shows the performance of our results compared to competitors. The results show that our approach outperforms the other three baselines by a reasonable margin. As expected CEC is the second best algorithm with a margin of 5 for the correlation maximization objective. It translates to increase in amount of correlation equal to 5, which is a noticeable amount. Furthermore, in the difference minimization task, our approach reached around 7 in difference. This means that we decreased the difference in exposure to the two processes to less 2.6 per user, which is considerable improvement. For both tasks, LTD made more mitigation posts over all daytime phases than it did over all nighttime phases, whereas the competitor strategies did the opposite. This could be a reason for its better performance. One surprising fact is that the number of retweets by users outside the network, which was not used for our features, can exceed the number of retweets by users within the network. This is because the "hashtag" feature on Twitter allows posts to be seen by a much larger set of users, who do not necessarily follow the

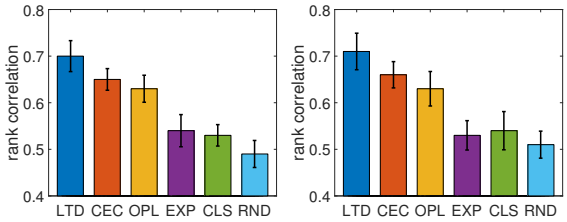



(a) Correlation (b) Difference

Figure 5.5: Results of fake news mitigation on Twitter network

source accounts. In addition to retweets, users can also "like" a post, indicating that they were exposed to fake or real news; while we measured this, we did not include it in the reward. Future experiments can use these two observations to widen the experimental scope and more accurately measure the effectiveness of a mitigation strategy. Despite having these limitations, our experiment serves as a proof-of-concept for the applicability of point process based intervention in networks, and–to the best of our knowledge–is the first to verify the superiority of a method in a real-time, real-world intervention setting.

**Prediction evaluation results.** The previous part described the evaluation scheme of real-time intervention in a social media platform. In this part, we used historical real data to mimic this procedure. We extracted 12 full 10-stage trajectories of events from the 2-month historical data under the random policy. For any of these 10 pairs, the methods were evaluated according to how well they predict the relative ordering among these 12 trajectories (with respect to the objective function). To evaluate each method, we created a sorted list of these 12 trajectories according to increasing objective, and created a second list sorted by increasing closeness to the intervention method. This closeness is the mean squared error between the prescribed intervention and actual intensity, which we inferred using maximum likelihood. Then, by computing the rank correlation of the two sorted lists, and repeating for each of the five methods, we can find out how well they perform on the prediction task. A better predictor is expected to be a better mitigation strategy. Figure 5.6 shows the performance.

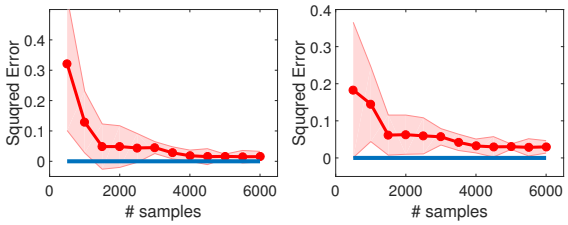



(a) Correlation (b) Difference

Figure 5.6: Rank correlation for prediction

### 5.4.5 Linear Approximation Accuracy

In our LSTD algorithm we used a linear approximation for the value-function. One might ask how accurately can linear features approximate the value-function. To this end, we take a sample state $x$ as a state with no prior activity and intensity $\psi(x) = (0, \ldots, 0, 1)$. This is the the initial state assumed in our experiment runs. First we empirically found $V^{\pi}(x)$ under the learned policy by simulating the process 100 times each with 10 stages. We compare the empirical average and the standard deviation of the total reward with the one estimated by the linear approximation $\psi^{\top} w^{\pi}$. Figure 5.7 shows the results for correlation maximization and difference minimization. In both cases, by increasing the number of samples (used in LSTD), the estimated $w$ leads to better estimation of $V^{\pi}(x)$. First, the figure shows that we can achieve a reasonable accuracy with a fair amount of samples. Secondly, although it appears that the approximation is converging to the empirical value, we notice that increasing the number of samples beyond 4000 does not improve the error, which maintains a constant distance from 0. We believe this is because the optimal value function does not lie in the linear span of the feature space. Employing more complex features, such as polynomial features and deep neural network based representations, remain as interesting avenues for future work.




(a) Correlation (b) Difference

Figure 5.7: Convergence of linear approximated value function

## 5.5 Related Work

The emergence of social media as a prominent news source in the past few years raises concomitant concerns about the quality, truthfulness, and credibility of information presented [139, 198, 154]. Many efforts have been made to study and analyze the dynamics of fake news [168, 210], detect and recognize them [173, 176, 115], or assessing the nodes vulnerability [171] over social networks. To reduce the amount of labor-intensive manual fact-checking, there have been research efforts devoted to building classifiers to detect factuality of information, predicting credibility level of posts, and detecting controversial information from inquiry phrases [139, 224, 226]. These works mainly focused on extracting linguistic features from texts to determine the credibility of news and posts. Our focus in this chapter, however, is to design an incentive strategy so that users can spontaneously take action to promote real news, in opposition to a real-world fake news epidemic.

Point process models have been recently used to model activities in networks [58, 150, 100, 109, 211]. More especially Hawkes process [93] is a class of self- and mutually exciting point processes that has been applied to variety of problems in social networks including cascade modeling [223], reliability of crowd generated data [186], social media popularity [163], community detection [191], causal inference [216], linguistic influence [87], and change point detection in social networks [128].

Steering user activities by adding external incentives to the exogenous intensity of Hawkes processes was first considered in [55]. In [55], a multistage campaigning method to

optimally distribute incentive resources based on dynamic programming was developed. In these previous works, objective functions were designed using expected values of exposure counts rather than the stochastic exposure process, which may reduce the accuracy of solutions. Furthermore, it faced the demanding problem of computing the cost-to-go using the Hawkes model, while we address this using linear function approximation. For stationary Hawkes processes, second order statistics was derived in [12, 13]; however, it is essential to compute both first and second order statistics for Hawkes processes in the *non-stationary* stages due to time sensitivity of the fake news mitigation task, and we derive it for the first time in this chapter. Recent work has also applied methods in stochastic differential equations to the context of point processes, to find the best intensity for information guiding [204] and achieving highest visibility [222]. While these works consider networks with only a single process, our work focuses on optimizing a mitigation process with respect to a second competing process. Finally, although our goal is related to influence maximization problems [111, 18], our point process approach is much more general and has greater temporal resolution, as it models continuous-time recurrent activity in networks, in contrast to binary discrete-time infection states in traditional influence maximization approaches. Moreover, our framework enables one to consider a variety of objectives (not only maximization) and incorporate budget constraints.

Reinforcement learning tackles the problem of finding good policies for actions to take in MDP where exact solutions are intractable, either due to size or lack of complete knowledge. Large-scale policy evaluation and iteration problems can be tackled by function approximation, which reduces the solution dimension using feature vector basis [185]. By adding control terms to a multivariate Hawkes process model of random network activities, fake news mitigation can be formulated as a policy optimization problem in an MDP. To address the randomness of Hawkes processes, batch reinforcement learning using samples collected from the trajectory of a fixed behavior policy can be applied [7]. In particular, linear Least Squares Temporal Difference (LSTD) uses a batch of samples to learn a linear

approximation of the value function under a policy with provable convergence [26]. This *policy evaluation* step alternates with a model-based *policy improvement* step in a policy iteration to arrive at successively improved policies.

## 5.6 Summary and Discussion

The use of point process based intervention comes with a tradeoff: on one hand, the stochastic nature of multivariate point processes allows it to model the uncertainty of event occurrences in real-world networks; however, by adopting this stochastic model, the intervention policy can only set the optimal conditional intensity, rather than the precise best times, of fake news mitigation events. This can be improved in future experiments by choosing shorter time intervals for stages in the mitigation campaign. An assumption made in the real-world experiment is that all events by user $u$ are seen by user $v$ if $v$ follows $u$, meaning that fake or real news events at $u$ are seen by $v$. While it is true for Twitter that all tweets by $u$ will appear on the home timeline of $v$ who follows $u$ [193], it is not necessarily true that a follower $v$ will see these tweets (suppose they did not access Twitter that day). Therefore, future experiments can improve the accuracy of reward and performance measurements by estimating the probability of users being online during certain time intervals and seeing tweets from accounts they follow.

It is important to note that mitigating fake news is a vague and qualitative goal, and it does not necessarily imply a reduction of fake news events. Matching the exposure of users to real and fake news is one of many possible objectives for arriving at a precise quantitative realization of mitigating fake news. One can define any objective function that accounts for the number of exposures or events from fake and real news cascades. Furthermore, the exposure itself can be interpreted in many different ways depending on the application. However, whenever there are multiple dependent event sequences and the rate function of one or many can be controlled, the point process framework naturally allows one to define objective functions based on events and to find an optimal policy with respect to the desired

goal.

We acknowledge that the experiments conducted in this present work do not directly test for a reduction of fake news events, and that the content-neutral real-time experiment does not perfectly represent fake news processes, as semantic content may also contribute to propagation dynamics. From a modeling standpoint, introducing interactions between the fake and real news processes may allow one to design objectives that specifically reward the reduction of fake news events. Aside from subjecting real users to actual fake news, more complex real-world experiments that are not content-neutral, involving two competing opinions, may allow one to better measure the effectiveness of various methods in reducing the spread of one opinion. Furthermore, for future work we would like to incorporate more complex features, such as quadratic and nonlinear features. Utilizing deep neural nets is also an interesting future work for modeling complex feature set.

# CHAPTER 6
# GENERATIVE POINT PROCESS MODELS VIA DEEP WASSERSTEIN LEARNING

Point processes are often characterized via intensity function which limits model's expressiveness due to unrealistic assumptions on its parametric form used in practice. Furthermore, they are learned via maximum likelihood approach which is prone to failure in multi-modal distributions of sequences. In this chapter, we propose an intensity-free approach for point processes modeling that transforms nuisance processes to a target one. Furthermore, we train the model using a likelihood-free approach leveraging Wasserstein distance between point processes. Experiments on various synthetic and real-world data substantiate the superiority of the proposed point process model over conventional ones.

## 6.1 Introduction

Conventional point process models often make strong unrealistic assumptions about the generative processes of the event sequences. In fact, a point process is characterized by its *conditional intensity function* – a stochastic model for the time of the next event given all the times of previous events. The functional form of the intensity is often designed to capture the phenomena of interests [58]. Some examples are homogeneous and non-homogeneous Poisson processes [113], self-exciting point processes [93], self-correcting point process models [105], and survival processes [1]. Unfortunately, they make various parametric assumptions about the latent dynamics governing the generation of the observed point patterns. As a consequence, model misspecification can cause significantly degraded performance using point process models, which is also shown by our experimental results later.

To address the aforementioned problem, the authors in [47, 213, 137] propose to learn a

general representation of the underlying dynamics from the event history without assuming a fixed parametric form in advance. The intensity function of the temporal point process is viewed as a nonlinear function of the history of the process and is parameterized using a recurrent neural network. Attentional mechanism is explored to discover the underlying structure [212]. Apparently this line of work still relies on explicit modeling of the intensity function. However, in many tasks such as data generation or event prediction, knowledge of the whole intensity function is unnecessary. On the other hand, sampling sequences from intensity-based models is usually performed via a thinning algorithm [148], which is computationally expensive; many sample events might be rejected because of the rejection step, especially when the intensity exhibits high variation. More importantly, most of the methods based on intensity function are trained by maximizing log likelihood or a lower bound on it. They are asymptotically equivalent to minimizing the Kullback-Leibler (KL) divergence between the data and model distributions, which suffers serious issues such as mode dropping [8, 74]. Alternatively, Generative Adversarial Networks (GAN) [75] have seen to be a promising alternative to traditional maximum likelihood approaches [104, 187].

In this chapter, for the first time, we bypass the intensity-based modeling and likelihood-based estimation of temporal point processes and propose a neural network-based model with a generative adversarial learning scheme for point processes. In GANs, two models are used to solve a minimax game: a generator which samples synthetic data from the model, and a discriminator which classifies the data as real or synthetic. Theoretically speaking, these models are capable of modeling an arbitrarily complex probability distribution, including distributions over discrete events. They achieve state-of-the-art results on a variety of generative tasks such as image generation, image super-resolution, 3D object generation, and video prediction [147, 157].

The original GAN in [75] minimizes the Jensen-Shannon (JS) and is regarded as highly unstable and prone to miss modes. Recently, Wasserstein GAN (WGAN) [9] is proposed to use the Earth Moving distance (EM) as an objective for training GANs. Furthermore

it is shown that the EM objective, as a metric between probability distributions [227] has many advantages as the loss function correlates with the quality of the generated samples and reduces mode dropping [84]. Moreover, it leverages the geometry of the space of event sequences in terms of their distance, which is not the case for an MLE-based approach. In this chapter we extend the notion of WGAN for temporal point processes and adopt a Recurrent Neural Network (RNN) for training. Importantly, we are able to demonstrate that Wasserstein distance training of RNN point process models outperforms the same architecture trained using MLE.

In a nutshell, we make the following contributions: i) We propose the first intensity-free generative model for point processes and introduce the first (to our best knowledge) likelihood-free corresponding learning methods; ii) We extend WGAN for point processes with Recurrent Neural Network architecture for sequence generation learning; iii) In contrast to the usual subjective measures of evaluating GANs we use a statistical and a quantitative measure to compare the performance of the model to the conventional ones. iv) Extensive experiments involving various types of point processes on both synthetic and real datasets show the promising performance of our approach.

## 6.2 Proposed Framework

In this section, we define Point Processes in a way that is suitable to be combined with the WGANs.

### 6.2.1 Point Processes

Let $S$ be a compact space equipped with a Borel $\sigma$-algebra $\mathcal{B}$. Take $\Xi$ as the set of counting measures on $S$ with $\mathcal{C}$ as the smallest $\sigma$-algebra on it. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. A point process on $S$ is a measurable map $\xi : \Omega \to \Xi$ from the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ to the measurable space $(\Xi, \mathcal{C})$. Figure 6.1-a illustrates this mapping.

Every *realization* of a point process $\xi$ can be written as $\xi = \sum_{i=1}^{n} \delta_{X_i}$ where $\delta$ is

the Dirac measure, $n$ is an integer-valued random variable and $X_i$'s are random elements of $S$ or *events*. A point process can be equivalently represented by a *counting process*: $N(B) := \int_B \xi(x)dx$, which basically is the number of events in each Borel subset $B \in \mathcal{B}$ of $S$. The *mean measure* $M$ of a point process $\xi$ is a measure on $S$ that assigns to every $B \in \mathcal{B}$ the expected number of events of $\xi$ in $B$, *i.e.*, $M(B) := \mathbb{E}[N(B)]$ for all $B \in \mathcal{B}$.

For *inhomogeneous Poisson process*, $M(B) = \int_B \lambda(x)dx$, where the intensity function $\lambda(x)$ yields a positive measurable function on $S$. Intuitively speaking, $\lambda(x)dx$ is the expected number of events in the infinitesimal $dx$. For the most common type of point process, a *Homogeneous Poisson process*, $\lambda(x) = \lambda$ and $M(B) = \lambda|B|$, where $|\cdot|$ is the Lebesgue measure on $(S, \mathcal{B})$. More generally, in *Cox point processes*, $\lambda(x)$ can be a random density possibly depending on the history of the process. For any point process, given $\lambda(\cdot)$, $N(B) \sim$ Poisson$(\int_B \lambda(x)dx)$. In addition, if $B_1, \ldots, B_k \in \mathcal{B}$ are disjoint, then $N(B_1), \ldots, N(B_k)$ are independent conditioning on $\lambda(\cdot)$.

For the ease of exposition, we will present the framework for the case where the events are happening in the real half-line of time. But the framework is easily extensible to the general space.

### 6.2.2 Temporal Point Processes

A particularly interesting case of point processes is given when $S$ is the time interval $[0, T)$, which we will call a *temporal point process*. Here, a realization is simply a set of time points: $\xi = \sum_{i=1}^{n} \delta_{t_i}$. With a slight notation abuse we will write $\xi = \{t_1, \ldots, t_n\}$ where each $t_i$ is a random time before $T$. The conditional intensity (rate) function is the usual way to characterize temporal point processes.

For Inhomogeneous Poisson process (**IP**), the intensity $\lambda(t)$ is a fixed non-negative function supported in $[0, T)$. For example, it can be a multi-modal function comprised of $k$ Gaussian kernels: $\lambda(t) = \sum_{i=1}^{k} \alpha_i (2\pi\sigma_i^2)^{-1/2} \exp\left(-(t - c_i)^2/\sigma_i^2\right)$, for $t \in [0, T)$, where $c_i$ and $\sigma_i$ are fixed center and standard deviation, respectively, and $\alpha_i$ is the weight (or

importance) for kernel $i$.

A self-exciting (Hawkes) process (**SE**) is a cox process where the intensity is determined by previous (random) events in a special parametric form: $\lambda(t) = \mu + \beta \sum_{t_i<t} g(t - t_i)$, where $g$ is a nonnegative kernel function, *e.g.*, $g(t) = \exp(-\omega t)$ for some $\omega > 0$. This process has an implication that the occurrence of an event will increase the probability of near future events and its influence will (usually) decrease over time, as captured by (the usually) decaying fixed kernel $g$. $\mu$ is the exogenous rate of firing events and $\alpha$ is the coefficient for the endogenous rate.

In contrast, in self-correcting processes (**SC**), an event will decrease the probability of an event: $\lambda(t) = \exp(\eta t - \sum_{t_i<t} \gamma)$. The $\exp$ ensures that the intensity is positive, while $\eta$ and $\gamma$ are exogenous and endogenous rates.

We can utilize more flexible ways to model the intensity, *e.g.*, by a Recurrent Neural Network (**RNN**): $\lambda(t) = g_w(t, h_{t_i})$, where $h_{t_i}$ is the feedback loop capturing the influence of previous events (last updated at the latest event) and is updated by $h_{t_i} = h_v(t_i, h_{t_{i-1}})$. Here $w$ and $v$ are network weights.

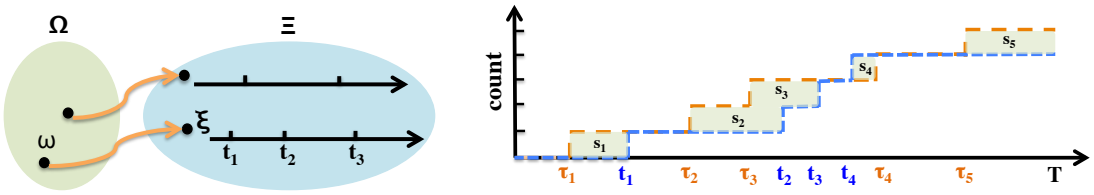
### 6.2.3 Wasserstein-Distance for Temporal Point Processes

Given samples from a point process, one way to estimate the process is to find a model $(\Omega_g, \mathcal{F}_g, \mathbb{P}_g) \to (\Xi, \mathcal{C})$ that is *close enough* to the real data $(\Omega_r, \mathcal{F}_r, \mathbb{P}_r) \to (\Xi, \mathcal{C})$. As mentioned in the introduction, Wasserstein distance [9] is our choice as the proximity measure. The Wasserstein distance between distribution of two point processes is:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\psi \in \Psi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(\xi,\rho)\sim\psi}[\|\xi - \rho\|_\star], \tag{6.1}$$

where $\Psi(\mathbb{P}_r, \mathbb{P}_g)$ denotes the set of all joint distributions $\psi(\xi, \rho)$ whose marginals are $\mathbb{P}_r$ and $\mathbb{P}_g$.

The distance between two sequences $\|\xi - \rho\|_\star$, is tricky and need further attention. Take $\xi = \{x_1, x_2, \ldots, x_n\}$ and $\rho = \{y_1, \ldots, y_m\}$, where for simplicity we first consider the case $m = n$. The two sequences can be thought as discrete distributions $\mu^\xi = \sum_{i=1}^n \frac{1}{n} \delta_{x_i}$ and




a) Point process probability space b) $\|\cdot\|_\star$ distance between sequences

Figure 6.1: a) The outcome of the random experiment $\omega$ is mapped to a point in space of count measures $\xi$; b) Distance between two sequences $\xi = \{t_1, t_2, \ldots\}$ and $\rho = \{\tau_1, \tau_2, \ldots\}$

$\mu^\rho = \sum_{i=1}^n \frac{1}{n}\delta_{y_i}$. Then, the distance between these two is an optimal transport problem $\operatorname{argmin}_{\pi\in\Sigma}\langle\pi, C\rangle$, where $\Sigma$ is the set of doubly stochastic matrices (rows and columns sum up to one), $\langle\cdot,\cdot\rangle$ is the Frobenius dot product, and $C$ is the cost matrix. $C_{ij}$ captures the energy needed to move a probability mass from $x_i$ to $y_j$. We take $C_{ij} = \|x_i - y_j\|_\circ$ where $\|\cdot\|_\circ$ is the norm in $S$. It can be seen that the optimal solution is attained at extreme points and, by Birkhoff's theorem, the extreme points of the set of doubly stochastic matrices is a permutation [199]. In other words, the mass is transferred from a unique source event to a unique target event. Therefore, we have: $\|\xi - \rho\|_\star = \min_\sigma \sum_{i=1}^n \|x_i - y_{\sigma(i)}\|_\circ$, where the minimum is taken among all $n!$ permutations of $1 \ldots n$. For the case $m \neq n$, without loss of generality we assume $n \leq m$ and define the distance as follows:

$$\|\xi - \rho\|_\star = \min_\sigma \sum_{i=1}^n \|x_i - y_{\sigma(i)}\|_\circ + \sum_{i=n+1}^m \|s - y_{\sigma(i)}\|, \tag{6.2}$$

where $s$ is a fixed limiting point in border of the compact space $S$ and the minimum is over all permutations of $1 \ldots m$. The second term penalizes unmatched points in a very special way which will be clarified later. The following lemma proves that it is indeed a valid distance measure.

**Lemma 12.** *The defined $\|\cdot\|_\star$ is a norm.*

*Proof.* It is obvious that $\|\cdot\|_\star$ is nonnegative and symmetric. If $\|\xi - \rho\|_\star = 0$, then $m = n$ and there is a assignment $\sigma$ such that $x_i = y_{\sigma(i)}$ for all $i = 1, \ldots, n$.

Now we prove that $\|\cdot\|_\star$ has triangle inequality. WLOG, assume that $\xi = \{x_1, \ldots, x_n\}$, $\rho = \{y_1, \ldots, y_k\}$ and $\zeta = \{z_1, \ldots, z_m\}$ where $n \leq k \leq m$. Define the permutation $\hat{\sigma}$ on

$\{1, \ldots, k\}$ by

$$\hat{\sigma} := \arg\min_{\sigma} \sum_{i=1}^{n} \|x_i - y_{\sigma(i)}\| + \sum_{i=n+1}^{k} \|s - y_{\sigma(i)}\| \tag{6.3}$$

Then we know that

$$\|\xi - \rho\|_\star = \sum_{i=1}^{n} \|x_i - y_{\hat{\sigma}(i)}\| + \sum_{i=n+1}^{k} \|s - y_{\hat{\sigma}(i)}\| \tag{6.4}$$

Therefore, we have that

$$\begin{aligned}
\|\xi - \zeta\|_\star &= \min_{\sigma} \sum_{i=1}^{n} \|x_i - z_{\sigma(i)}\| + \sum_{i=n+1}^{m} \|s - z_{\sigma(i)}\| \\
&\leq \min_{\sigma} \sum_{i=1}^{n} \left(\|x_i - y_{\hat{\sigma}(i)}\| + \|y_{\hat{\sigma}(i)} - z_{\sigma(i)}\|\right) + \sum_{i=n+1}^{k} \left(\|s - y_{\hat{\sigma}(i)}\| + \|y_{\hat{\sigma}(t)} - z_{\sigma(i)}\|\right) \\
&\qquad + \sum_{i=k+1}^{m} \|s - z_{\sigma(i)}\| \qquad (6.5) \\
&= \|\xi - \rho\|_\star + \min_{\sigma} \sum_{i=1}^{k} \|y_{\hat{\sigma}(i)} - z_{\sigma(i)}\| + \sum_{i=k+1}^{m} \|s - z_{\sigma(i)}\| \\
&= \|\xi - \rho\|_\star + \min_{\sigma} \sum_{i=1}^{k} \|y_i - z_{\sigma(\hat{\sigma}^{-1}(i))}\| + \sum_{i=k+1}^{m} \|s - z_{\sigma(i)}\| \\
&= \|\xi - \rho\|_\star + \|\rho - \zeta\|_\star
\end{aligned}$$

where the last equality is due to the fact that the minimization is taken over all permutations $\sigma$ of $\{1, \ldots, m\}$, and $\hat{\sigma}$ is a fixed permutation of $\{1, \ldots, k\}$ where $k \leq m$. This completes the proof. □

Interestingly, in the case of temporal point process in $[0, T)$ the distance between $\xi = \{t_1, \ldots, t_n\}$ and $\rho = \{\tau_1, \ldots, \tau_m\}$ is reduced to

$$\|\xi - \rho\|_\star = \sum\nolimits_{i=1}^{n} |t_i - \tau_i| + (m - n) \times T - \sum\nolimits_{i=n+1}^{m} \tau_i, \tag{6.6}$$

where the time points are ordered increasingly, $s = T$ is chosen as the anchor point, and $|\cdot|$ is the Lebesgue measure in the real line.

**Lemma 13.** *Finding the distance between sequences $\xi$ and $\rho$,*

$$\|\xi - \rho\|_\star = \min_\sigma \sum_{i=1}^{n} \|x_i - y_{\sigma(i)}\|_\circ + \sum_{i=n+1}^{m} \|s - y_{\sigma(i)}\|, \tag{6.7}$$

*in the case of temporal point process in* $[0, T)$, i.e., $\xi = \{t_1 < t_2 < \ldots < t_n\}$ *and* $\rho = \{\tau_1 < \tau_2 < \ldots < \tau_m\}$, *reduces to*

$$\|\xi - \rho\|_\star = \sum_{i=1}^{n} |t_i - \tau_i| + \sum_{i=n+1}^{m} (T - y_i). \tag{6.8}$$

*Proof.* Here, without loss of generality $n \leq m$ is assumed. The choice of $s = T$ is basically padding the shorter sequences with $T$. Given, the sequences have the same length now, we claim that the identity permutation *i.e.*, $\sigma(i) = i$ is the minimizer in (6.7). We proceed by a proof by contradiction. Assume that the minimizer is NOT the identity permutation. Then, find the first $i$ such that $\sigma(i) \neq i$. Then, $\Sigma(i) = j$ where $j > i$. Therefore, there should be a $k > i$ such that $\sigma(k) = i$. Then, if you change the permutation according to $\sigma(i) = i$ and $\sigma(k) = j$ the cost will change by

$$\Delta = \underbrace{(|t_i - \tau_j| + |t_k - \tau_i|)}_{\text{for the old permutation}} - \underbrace{(|t_i - \tau_i| + |t_k - \tau_j|)}_{\text{for the new permutation}} \tag{6.9}$$

Given $i < j$ and $i < k$, it is easy to see that $\Delta > 0$. This means that we've found a better permutation which contradicts our assumption. Therefore, the optimal permutation will match the event points in an increasing order one by one. □

This choice of distance is significant in two senses. First, it is computationally efficient and no excessive computation is involved. Secondly, in terms of point processes, it is interpreted as the volume by which the two counting measures differ. Figure 6.1-b demonstrates this intuition and justifies our choice of metric in $\Xi$ and the following lemma contains the proof.

**Lemma 14.** *Equivalence of the $\|\cdot\|_\star$ distance and difference in count measures.*

*Proof.* The count measure of a temporal point process is a special case of the one defined for point processes in general space in Section 6.2.1. For a Borel subset $B \subset S = [0, T)$

we have $N(B) = \int_{t\in B} \xi(t)dt$. With a little abuse of notation we write $N(t) := N([0,t)) = \int_0^t \xi(t)dt$. Figure 6.1 is a good guidance through this paragraph. Starting from time 0 the first gap in count measure starts from $\min(t_1, \tau_1)$ and ends in $\max(t_1, \tau_1)$. Therefore, there is difference equal to $s_1 = \max(t_1, \tau_1) - \min(t_1, \tau_1) = |t_1 - \tau_1|$ in the count measure. Similarly, the second block of difference has volume of $s_2 = |t_2 - \tau_2|$, and so on. Finally, for $m > n$ the $(n+i)$-th block make a difference of $s_{n+i} = T - \tau_{n+i}$. Therefore, the area ($L_1$ distance) between the two sequences is a equal to $S = \sum_{i=1}^m s_i$. On the other hand by looking (6.8) we observe that $\|\xi - \rho\|_\star = \sum_{i=1}^m s_i$. Therefore, by choice of $s = T$ as an anchor point, the distance we have is exactly the area between the two count measures. □

The distance used in our current work is the simplest yet effective distance that exhibits high interpretability and efficient computability. More robust distance like local alignment distance and dynamic time warping [39] should be more robust and are great venues for future work.

Equation (6.1) is computationally highly intractable and its dual form is usually utilized [9]:

$$W(\mathbb{P}_r, \mathbb{P}_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{\xi\sim\mathbb{P}_r}[f(\xi)] - \mathbb{E}_{\rho\sim\mathbb{P}_g}[f(\rho)], \tag{6.10}$$

where the supremum is taken over all Lipschitz functions $f : \Xi \to \mathbb{R}$, *i.e.*, functions that assign a value to a sequence of events (points) and satisfy $|f(\xi) - f(\rho)| \leq \|\xi - \rho\|_\star$ for all $\xi$ and $\rho$.

However, solving the dual form is still highly nontrivial. Enumerating all Lipschitz functions over point process realizations is impossible. Instead, we choose a parametric family of functions to approximate the search space $f_w$ and consider solving the problem

$$\max_{w\in\mathcal{W}, \|f_w\|_L \leq 1} \mathbb{E}_{\xi\sim\mathbb{P}_r}[f_w(\xi)] - \mathbb{E}_{\rho\sim\mathbb{P}_g}[f_w(\rho)] \tag{6.11}$$

where $w \in \mathcal{W}$ is the parameter. The more flexible $f_w$, the more accurate will be the approximation.

It is notable that W-distance leverages the geometry of the space of event sequences in

terms of their distance, which is not the case for MLE-based approach. It in turn requires functions of event sequences $f(x_1, x_2, ...)$, rather than functions of the time stamps $f(x_i)$. Furthermore, Stein's method to approximate Poisson processes [175, 43] is also relevant as they are defining distances between a Poisson process and an arbitrary point process.

### 6.2.4 WGAN for Temporal Point Processes

Equipped with a way to approximately compute the Wasserstein distance, we will look for a model $\mathbb{P}_r$ that is close to the distribution of real sequences. Again, we choose a sufficiently flexible parametric family of models, $g_\theta$ parameterized by $\theta$. Inspired by GAN [75], this generator takes a noise and turns it into a sample to mimic the real samples. In conventional GAN or WGAN, Gaussian or uniform distribution is chosen. In point processes, a homogeneous Poisson process plays the role of a non-informative and uniform-like distribution: the probability of events in every region is independent of the rest and is proportional to its volume. Define the noise process as $(\Omega_z, \mathcal{F}_z, \mathbb{P}_z) \to (\Xi, \mathcal{C})$, then $\zeta \sim \mathbb{P}_z$ is a sample from a Poisson process on $S = [0, T)$ with constant rate $\lambda_z > 0$. Therefore, $g_\theta : \Xi \to \Xi$ is a transformation in the space of counting measures. Note that $\lambda_z$ is part of the prior knowledge and belief about the problem domain. Therefore, the objective of learning the generative model can be written as $\min W(\mathbb{P}_r, \mathbb{P}_g)$ or equivalently:

$$\min_{\theta} \max_{w \in \mathcal{W}, \|f_w\|_L \leq 1} \mathbb{E}_{\xi \sim \mathbb{P}_r}[f_w(\xi)] - \mathbb{E}_{\zeta \sim \mathbb{P}_z}[f_w(g_\theta(\zeta))] \tag{6.12}$$

In GAN terminology $f_w$ is called the *discriminator* and $g_\theta$ is known as the *generator* model. We estimate the generative model by enforcing that the sample sequences from the model have the same distribution as training sequences. Given $L$ samples sequences from real data $\mathcal{D}_r = \{\xi_1, \ldots, \xi_L\}$ and from the noise $\mathcal{D}_z = \{\zeta_1, \ldots, \zeta_L\}$ the two expectations are estimated empirically: $\mathbb{E}_{\xi \sim \mathbb{P}_r}[f_w(\xi)] = \frac{1}{L} \sum_{l=1}^{L} f_w(\xi_l)$ and $\mathbb{E}_{\zeta \sim \mathbb{P}_z}[f_w(g_\theta(\zeta))] = \frac{1}{L} \sum_{l=1}^{L} f_w(g_\theta(\zeta_l))$.

To proceed with our point process based WGAN, we need the generator function $g_\theta : \Xi \to \Xi$, the discriminator function $f_w : \Xi \to \mathbb{R}$, and enforce Lipschitz constraint on $f_w$.

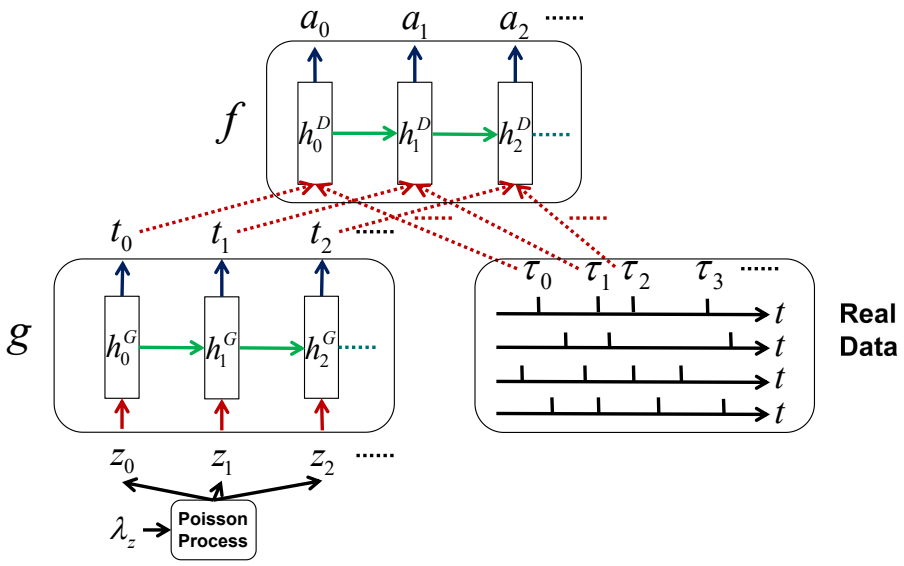



Figure 6.2: The input and output sequences are $\zeta = \{z_1, \ldots, z_n\}$ and $\rho = \{t_1, \ldots, t_n\}$ for generator $g_\theta(\zeta) = \rho$, where $\zeta \sim Poisson(\lambda_z)$ process and $\lambda_z$ is a prior parameter estimated from real data. Discriminator computes the Wasserstein distance between the two distributions of sequences $\rho = \{t_1, t_2, \ldots\}$ and $\xi = \{\tau_1, \tau_2, \ldots\}$

Figure 6.2 illustrates the data flow for WGANTPP.

The generator transforms a given sequence to another sequence. Similar to [140, 64] we use Recurrent Neural Networks (RNN) to model the generator. For clarity, we use the vanilla RNN to illustrate the computational process as below. The LSTM is used in our experiments for its capacity to capture long-range dependency. If the input and output sequences are $\zeta = \{z_1, \ldots, z_n\}$ and $\rho = \{t_1, \ldots, t_n\}$ then the generator $g_\theta(\zeta) = \rho$ works according to

$$h_i = \phi_g^h(A_g^h z_i + B_g^h h_{i-1} + b_g^h), \qquad t_i = \phi_g^x(B_g^x h_i + b_g^x) \tag{6.13}$$

Here $h_i$ is the $k$-dimensional history embedding vector and $\phi_g^h$ and $\phi_g^x$ are the activation functions. The parameter set of the generator is

$$\theta = \left\{ \left(A_g^h\right)_{k\times 1}, \left(B_g^h\right)_{k\times k}, \left(b_g^h\right)_{k\times 1}, \left(B_g^x\right)_{1\times k}, \left(b_g^x\right)_{1\times 1} \right\}.$$

Similarly, we define the discriminator function who assigns a scalar value $f_w(\rho) = \sum_{i=1}^n a_i$

to the sequence $\rho = \{t_1, \ldots, t_n\}$ according to

$$h_i = \phi_d^h(A_d^h t_i + B_g^h h_{i-1} + b_g^h) \qquad a_i = \phi_d^a(B_d^a h_i + b_d^a) \tag{6.14}$$

where the parameters are $w = \left\{ \left(A_d^h\right)_{k\times 1}, \left(B_d^h\right)_{k\times k}, \left(b_d^h\right)_{k\times 1}, \left(B_d^a\right)_{1\times k}, \left(b_d^a\right)_{1\times 1} \right\}$. Note that both generator and discriminator RNNs are causal networks. Each event is only influenced by the previous events. To enforce the Lipschitz constraints the original WGAN paper [8] adopts weight clipping. However, our initial experiments shows an inferior performance by using weight clipping. This is also reported by the same authors in their follow-up paper [84] to the original work. The poor performance of weight clipping for enforcing 1-Lipschitz can be seen theoretically as well: just consider a simple neural network with one input, one neuron, and one output: $f(x) = \sigma(wx + b)$ and the weight clipping $w < c$. Then,

$$|f'(x)| \leq 1 \Longleftrightarrow |w\sigma'(wx + b)| \leq 1 \Longleftrightarrow |w| \leq 1/|\sigma'(wx + b)| \tag{6.15}$$

It is clear that when $1/|\sigma'(wx + b)| < c$, which is quite likely to happen, the Lipschitz constraint is not necessarily satisfied. In our work, we use a novel approach for enforcing the Lipschitz constraints, avoiding the computation of the gradient which can be costly and difficult for point processes. We add the Lipschitz constraint as a regularization term to the empirical loss of RNN.

$$\min_{\theta} \max_{w\in\mathcal{W}, \|f_w\|_L \leq 1} \frac{1}{L}\sum_{l=1}^{L} f_w(\xi_l) - \sum_{l=1}^{L} f_w(g_\theta(\zeta_l)) - \nu \sum_{l,m=1}^{L} \left|\frac{|f_w(\xi_l) - f_w(g_\theta(\zeta_m))|}{|\xi_l - g_\theta(\zeta_m)|_\star} - 1\right| \tag{6.16}$$

We can take each of the $\binom{2L}{2}$ pairs of real and generator sequences, and regularize based on them; however, we have seen that only a small portion of pairs ($O(L)$), randomly selected, is sufficient.

**Remark**. The significance of Lipschitz constraint and regularization (or more generally any capacity control) is more apparent when we consider the connection of W-distance and optimal transport problem [199]. Basically, minimizing the W-distance between the

empirical distribution and the model distribution is equivalent to a semidiscrete optimal transport [199]. Without capacity control for the generator and discriminator, the optimal solution simply maps a partition of the sample space to the set of data points, in effect, *memorizing* the data points. We would like to highlight some recent efforts on defining distances and optimal transport for point process and sequential dynamic data [107, 39, 205, 138]. Interested reader is referred to [114, 182, 152] too for computational aspects of optimal transport and their application.

## 6.3 Experiments

In this section we provide experimental results [1]. The current work aims at exploring the feasibility of modeling point process without prior knowledge of its underlying generating mechanism. To this end, most widely-used parametrized point processes, e.g., self-exciting and self-correcting, and inhomogeneous Poisson processes and one flexible neural network model, neural point process are compared. In this work we use the most general forms for simpler and clear exposition to the reader and propose the very first model in adversarial training of point processes in contrast to likelihood based models.

### 6.3.1 Datasets and Protocol

**Synthetic datasets.** We simulate 20,000 sequences over time $[0, T)$ where $T = 15$, for inhomogeneous process (IP), self-exciting (SE), and self-correcting process (SC), recurrent neural point process (NN). We also create another 4 ($= C_4^3$) datasets from the above 4 synthetic data by a uniform mixture from the triplets. The new datasets IP+SE+SC, IP+SE+NN, IP+SC+NN, SE+SC+NN are created to testify the mode dropping problem of learning a generative model. The parameter setting follows:

i) **Inhomogeneous process.** The intensity function is independent from history and given in Sec. 6.2.2, where $k = 3, \alpha = [3, 7, 11], c = [1, 1, 1], \sigma = [2, 3, 2]$.

[1] The code is available https://github.com/xiaoshuai09/Wasserstein-Learning-For-Point-Process

ii) **Self-exciting process.** The past events increase the rate of future events. The conditional intensity function is given in Sec. 6.2.2 where $\mu = 1.0, \beta = 0.8$ and the decaying kernel $g(t - t_i) = e^{-(t-t_i)}$.

iii) **Self-correcting process.** The conditional intensity function is defined in Sec. 6.2.2. It increases with time and decreases by events occurrence. We set $\eta = 1.0, \gamma = 0.2$.

iv) **Recurrent Neural Network process.** The conditional intensity is given in Sec. 6.2.2, where the neural network's parameters are set randomly and fixed. We first feed random variable from [0,1] uniform distribution, and then iteratively sample events from the intensity and feed the output into the RNN to get the new intensity for the next step.

**Real datasets.** We collect sequences separately from four public available datasets, namely, health-care MIMIC-III, public media MemeTracker, NYSE stock exchanges, and publications citations. The time scale for all real data are scaled to [0,15], and the details are as follows:

i) **MIMIC.** MIMIC-III (Medical Information Mart for Intensive Care III) is a large, publicly available dataset, which contains de-identified health-related data during 2001 to 2012 for more than 40,000 patients. We worked with patients who appear at least 3 times, which renders 2246 patients. Their visiting timestamps are collected as the sequences.

ii) **Meme.** MemeTracker tracks the *meme* diffusion over public media, which contains more than 172 million news articles or blog posts. The memes are sentences, such as ideas, proverbs, and the time is recorded when it spreads to certain websites. We randomly sample 22,000 cascades.

iii) **MAS.** Microsoft Academic Search provides access to its data, including publication venues, time, citations, etc. We collect citations records for 50,000 papers.

iv) **NYSE.** We use 0.7 million high-frequency transaction records from NYSE for a stock in one day. The transactions are evenly divided into 3,200 sequences with equal durations.

### 6.3.2 Experimental Setup

**Details.** We can feed the temporal sequences to generator and discriminator directly. In practice, all temporal sequences are transformed into time duration between two consecutive events, i.e., transforming the sequence $\xi = \{t_1, \ldots, t_n\}$ into $\{\tau_1, \ldots, \tau_{n-1}\}$, where $\tau_i = t_{i+1} - t_i$. This approach makes the model train easily and perform robustly. The transformed sequences are statistically identical to the original sequences, which can be used as the inputs of our neural network. To make sure we that the times are increasing we use elu + 1 activation function to produce positive inter arrival times for the generator and accumulate the intervals to get the sequence. The Adam optimization method with learning rate 1e-4, $\beta_1 = 0.5, \beta_2 = 0.9$, is applied.

**Baselines.** We compare the proposed method of learning point processes (*i.e.*, minimizing sample distance) with maximum likelihood based methods for point process. To use MLE inference for point process, we have to specify its parametric model. The used parametric model are inhomogeneous Poisson process (mixture of Gaussian), self-exciting process, self-correcting process, and RNN. For each data, we use all the above solvers to learn the model and generate new sequences, and then we compare the generated sequences with real ones.

**Evaluation metrics.** Although our model is an intensity-free approach we will evaluate the performance by metrics that are computed via intensity. For all models, we work with the empirical intensity instead. Note that our objective measures are in sharp contrast with the best practices in GANs in which the performance is usually evaluated subjectively, *e.g.*, by visual quality assessment. We evaluate the performance of different methods to learn the underlying processes via two measures: 1) The first one is the well-known QQ plot of sequences generated from learned model. The quantile-quantile (q-q) plot is the graphical representation of the quantiles of the first data set against the quantiles of the second data set. From the time change property [113] of point processes, if the sequences come from the point process $\lambda(t)$ , then the integral $\Lambda = \int_{t_i}^{t_{t+1}} \lambda(s)ds$ between consecutive events

should be exponential distribution with parameter 1. Therefore, the QQ plot of $\Lambda$ against exponential distribution with rate 1 should fall approximately along a 45-degree reference line. The evaluation procedure is as follows: i) The ground-truth data is generated from a model, say IP; ii) All 5 methods are used to learn the unknown process using the ground-truth data; iii) The learned model is used to generate a sequence; iv) The sequence is used against the theoretical quantiles from the model to see if the sequence is really coming from the ground-truth generator or not; v) The deviation from slope 1 is visualized or reported as a performance measure. 2) The second metric is the deviation between empirical intensity from the learned model and the ground truth intensity. We can estimate empirical intensity $\lambda'(t) = E(N(t + \delta t) - N(t))/\delta t$ from sufficient number of realizations of point process through counting the average number of events during $[t, t + \delta t]$, where $N(t)$ is the count process for $\lambda(t)$. The $L_1$ distance between the ground-truth empirical intensity and the learned empirical intensity is reported as a performance measure.

### 6.3.3 Results and Discussion

**Synthetic data.** Figure 6.3 presents the learning ability of WGANTPP when the ground-truth data is generated via different types of point process. We first compare the QQ plots in the top row from the micro perspective view, where QQ plot describes the dependency between events. Red dots legend-ed with *Real* are the optimal QQ distribution, where the intensity function generates the sequences are known. We can observe that even though WGANTPP has no prior information about the ground-truth point process, it can estimate the model better except for the estimator that knows the parametric form of data. This is quite expected: When we are training a model and we know the parametric form of the generating model we can find it better. However, whenever the model is misspecified (*i.e.*, we don't know the parametric from *a priori*) WGANTPP outperforms other parametric forms and RNN approach. The middle row of figure 6.3 compares the empirical intensity. The *Real* line is the optimal empirical intensity estimated from the real data. The estimator

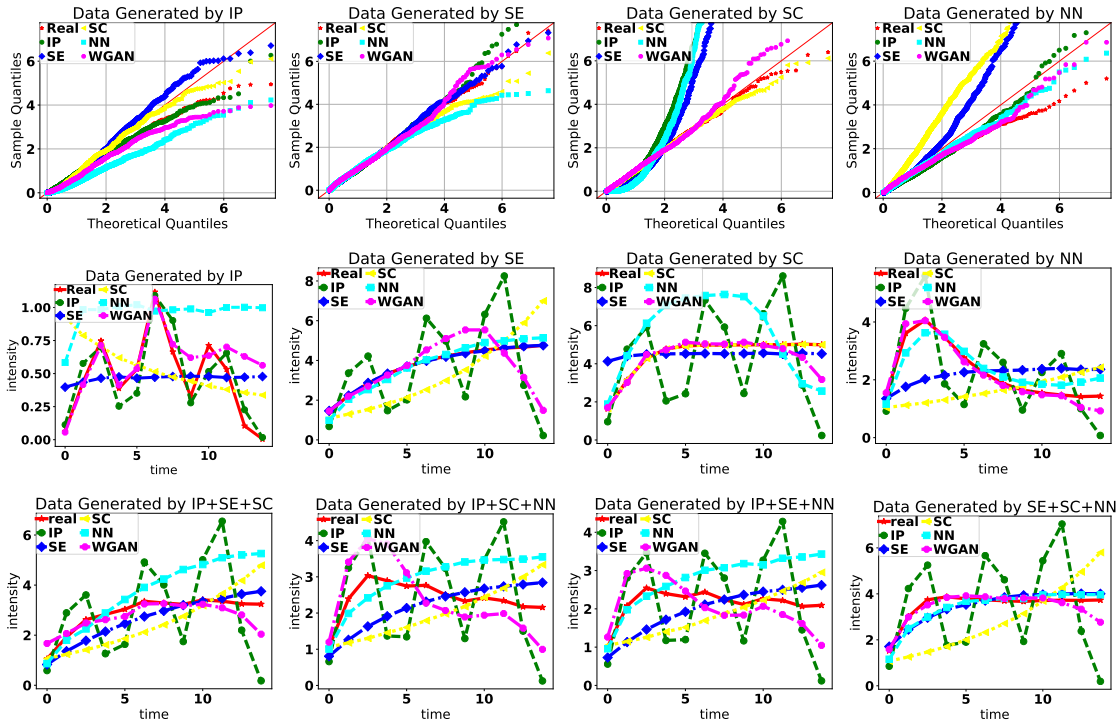



Figure 6.3: Performance of different methods on various synthetic data. Top row: QQ plot slope deviation; middle row: intensity deviation in basic conventional models; bottom row: intensity deviation in mixture of conventional processes.

can recover the empirical intensity well in the case that we know the parametric form where the data comes from. Otherwise, estimated intensity degrades considerably when the model is misspecified. We can observe our WGANTPP produces the empirical intensity better and performs robustly across different types of point process data. For MLE-IP, different number of kernels are tested and the empirical intensity results improves but QQ plot results degrade when the number of kernels increases, so only result of 3 kernels is shown mainly for clarity of presentation. The fact that the empirical intensity estimated from MLE-IP method are good and QQ plots are very bad indicates the inhomogeneous Poisson process can capture the average intensity (Macro dynamics) accurately but incapable of capturing the dependency between events (Micro dynamics). To testify that WGANTPP can cope with mode dropping, we generate mixtures of data from three different point processes and use this data to train different models. Models with specified form can handle limited types of data and fail to learn from diverse data sources. The last row of figure 6.3 shows

Table 6.1: Deviation of QQ plot slope and empirical intensity for ground-truth and learned model

| | Data | Estimator | | | | |
|---|---|---|---|---|---|---|
| | | MLE-IP | MLE-SE | MLE-SC | MLE-NN | WGAN |
| QQP. Dev. | IP | 0.035 (8.0e-4) | 0.284 (7.0e-5) | 0.159 (3.8e-5) | 0.216 (3.3e-2) | **0.033 (3.3e-3)** |
| | SE | 0.055 (6.5e-5) | **0.001 (1.3e-6)** | 0.086 (1.1e-6) | 0.104 (6.7e-3) | 0.051 (1.8e-3) |
| | SC | 3.510 (4.9e-5) | 2.778 (7.4e-5) | **0.002 (8.8e-6)** | 4.523 (2.6e-3) | 0.070 (6.4e-3) |
| | NN | 0.182 (1.6e-5) | 0.687 (5.0e-6) | 1.004 (2.5e-6) | 0.065 (1.2e-2) | **0.012 (4.7e-3)** |
| Int. Dev. | IP | **0.110 (1.9e-4)** | 0.241 (1.0e-4) | 0.289 (2.8e-5) | 0.511 (1.8e-1) | 0.136 (8.7e-3) |
| | SE | 1.950 (4.8e-4) | **0.019 (1.84e-5)** | 1.112 (3.1e-6) | 0.414 (1.6e-1) | 0.860 (6.2e-2) |
| | SC | 2.208 (7.0e-5) | 0.653 (1.2e-4) | **0.006 (9.9e-5)** | 1.384 (1.7e-1) | 0.302 (2.2e-3) |
| | NN | 1.044 (2.4e-4) | 0.889 (1.2e-5) | 1.101 (1.3e-4) | 0.341 (3.4e-1) | **0.144 (4.28e-2)** |
| Int. Dev. | IP+SE+SC | 1.505 (3.3e-4) | 0.410 (1.8e-5) | 0.823 (3.1e-6) | 0.929 (1.6e-1) | **0.305 (6.1e-2)** |
| | IP+SC+NN | 1.178 (7.0e-5) | 0.588 (1.3e-4) | 0.795 (9.9e-5) | 0.713 (1.7e-1) | **0.525 (2.2e-3)** |
| | IP+SE+NN | 1.052 (2.4e-4) | 0.453 (1.2e-4) | 0.583 (1.0e-4) | 0.678 (3.4e-1) | **0.419 (4.2e-2)** |
| | SE+SC+NN | 1.825 (2.8e-4) | 0.324 (1.1e-4) | 1.269 (1.1e-4) | 0.286 (3.6e-1) | **0.200 (3.8e-2)** |

the learned intensity from mixtures of data. WGANTPP produces better empirical intensity than alternatives, which fail to capture the heterogeneity in data. To verify the robustness of WGANTPP, we randomly initialize the generator parameters and run 10 rounds to get the mean and std of deviations for both empirical intensity and QQ plot from ground truth. For empirical intensity, we compute the integral of difference of learned intensity and ground-truth intensity. Table 6.1 reports the mean and std of deviations for intensity deviation. For each estimators, we obtain the slope from the regression line for its QQ plot. Table 6.1 reports the mean and std of deviations for slope of the QQ plot. Compared to the MLE-estimators, WGANTPP consistently outperforms even without prior knowledge about the parametric form of the true underlying generative point process. Note that for mixture models QQ-plot is not feasible.

**Real-world data.** We evaluate WGANTPP on a diverse real-world data process from health-care, public media, scientific activities and stock exchange. For those real world data, the underlying generative process is unknown, previous works usually assume that they are certain types of point process from their domain knowledge. Figure 6.4 shows the intensity learned from different models, where *Real* is estimated from the real-world data itself. Table 6.2 reports the intensity deviation. When all models have no prior knowledge

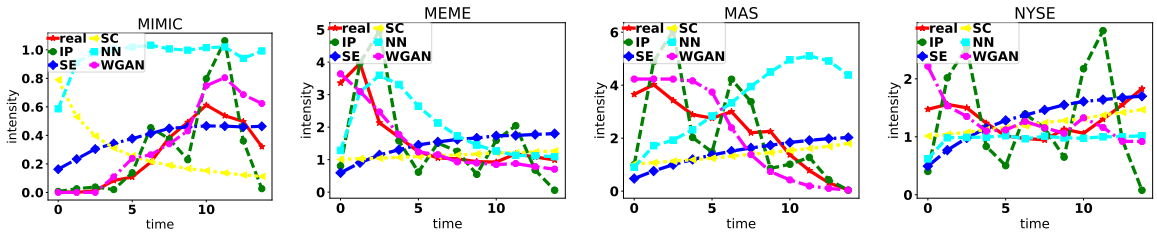



Figure 6.4: Performance of different methods on various real-world datasets.

Table 6.2: Deviation of empirical intensity for real-world data.

| Data | Estimator | | | | |
|---|---|---|---|---|---|
| | MLE-IP | MLE-SE | MLE-SC | MLE-NN | WGAN |
| MIMIC | 0.150 | 0.160 | 0.339 | 0.686 | **0.122** |
| Meme | 0.839 | 1.008 | 0.701 | 0.920 | **0.351** |
| MAS | 1.089 | 1.693 | 1.592 | 2.712 | **0.849** |
| NYSE | 0.799 | 0.426 | 0.361 | 0.347 | **0.303** |

about the true generative process, WGANTPP recovers intensity better than all the other models across the data sets.

**Analysis.** We have observed that when the generating model is misspecified WGANTPP outperforms the other methods without leveraging the a priori knowledge of the parametric form. However, when the exact parametric form of data is known and when it is utilized to learn the parameters, MLE with this full knowledge performs better. However, this is generally a strong assumption. As we have observed from the real-world experiments WGANTPP is superior in terms of performance. Somewhat surprising is the observation that WGANTPP tends to outperform the MLE-NN approach which basically uses the same RNN architecture but trained using MLE. The superior performance of our approach compared to MLE-NN is another witness of the the benefits of using W-distance in finding a generator that fits the observed sequences well. Even though the expressive power of the estimators is the same for WGANTPP and MLE-NN, MLE-NN may suffer from mode dropping or get stuck in an inferior local minimum since maximizing likelihood is asymptotically equivalent to minimizing the Kullback-Leibler (KL) divergence between the data and model distribution. The inherent weakness of KL divergence [9] renders MLE-NN

perform unstably, and the large variances of deviations empirically demonstrate this point.

## 6.4 Summary and Conclusion

We have presented a novel approach for Wasserstein learning of deep generative point processes which requires no prior knowledge about the underlying true process and can estimate it accurately across a wide scope of theoretical and real-world processes. For the future work, we would like to explore the connection of the WGAN with the optimal transport problem. We will also explore other possible distance metrics over the realizations of point processes, and more sophisticated transforms of point processes, particularly those that are causal. Extending the current work to marked point processes and processes over structured spaces is another interesting venue for future work.

# CHAPTER 7

# CONCLUSION

In this chapter, we conclude the thesis with extensions, discussions, and future works.

## 7.1 Point Process Modeling

The basic model presented in chapter 2 is just a show-case of the potential of point processes in modeling networks and processes over them. In this section, we extend point process models of information diffusion and network evolution 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 dataset does not contain information regarding to link removal and node birth and death. Curating a comprehensive dataset that can be used in modeling all these aspects of networks is left as interesting future work. The more our models are reflective of the real social networks, the better we can utilize them in prediction and optimization tasks. To do so, a model could incorporate the followings:

### 7.1.1 Link Deletion

We can generalize our model to support link deletion by introducing an intensity matrix $\boldsymbol{\Xi}^*(t) = \big(\boldsymbol{\xi}^*_{us}(t)\big)_{u,s\in[m]}$ 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 $\boldsymbol{A}^-(t) = \big(A^-_{us}(t)\big)_{u,s\in[m]}$, which indicates the lack of an edge at time $t$, and we define it via its intensity function as

$$\mathbb{E}[d\boldsymbol{A}^-(t) \mid \mathcal{H}^r(t) \cup \mathcal{H}^l(t)] = \boldsymbol{\Xi}^*(t)\, dt, \tag{7.1}$$

Then, we define the intensity as

$$\boldsymbol{\xi}^*_{us}(t) = A^+_{us}(t)(\zeta_u + \nu_s \sum_{v \in \mathcal{F}u} \kappa_{\omega_3}(t) \star dA^-_{vs}(t)), \tag{7.2}$$

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 (2.10), the parameter $\zeta_u$ 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.

### 7.1.2 Node Birth and Death

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

$$m(t) = m_b(t) - m_d(t) \tag{7.3}$$

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 $\phi^*(t)$:

$$\mathbb{E}[dm_b(t) \mid \mathcal{H}^r(t) \cup \mathcal{H}^l(t)] = \phi^*(t)\, dt, \tag{7.4}$$

where

$$\phi^*(t) = \epsilon + \theta \sum_{u,s \in [m(t)]} \kappa_{\omega_4}(t) \star dN_{us}(t), \tag{7.5}$$

Here, $\epsilon$ 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, $\boldsymbol{l}(t) = \big(l_u(t)\big)_{u\in[m]}$, characterized by a multidimensional conditional intensity function $\boldsymbol{\sigma}^*(t) = \big(\sigma_u(t)\big)_{u\in[m_b(t)]}$:

$$\mathbb{E}[d\boldsymbol{l}(t)|\mathcal{H}^r(t) \cup \mathcal{H}^l(t)] = \boldsymbol{\sigma}^*(t)\, dt, \tag{7.6}$$

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:

$$\sigma_u^*(t) = (1 - l_u(t)) \left[ \sum_{j=1}^{J} \pi_j g_j(t) + \left( h(t) - \sum_{s\in[m(t)]} \kappa_{\omega_5}(t) \star dN_{us}(t) \right)_+ \right], \tag{7.7}$$

where the term $\big(1 - l_u(t)\big)$ 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, $\big\{g_j(t)\big\}$ with unknown coefficients, $\big\{\pi_j\big\}$, 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

$$m_d(t) = \sum_{u=1}^{m_b(t)} l_u(t), \tag{7.8}$$

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

### 7.1.3 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. 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, etc.

Then, 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 $\eta_u$ 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 $\beta_s$ by $\boldsymbol{\beta}_s \boldsymbol{x}_s$. Therefore, one can rewrite the information diffusion intensity as:

$$\gamma^*_{us}(t) = \mathbb{I}[u = s]\, \boldsymbol{\eta}_u^\top \boldsymbol{x}_u + \mathbb{I}[u \neq s]\, \boldsymbol{\beta}_s^\top \boldsymbol{x}_s \sum\nolimits_{v \in \mathcal{F}_u(t)} \kappa_{\omega_1}(t) \star \big(A_{uv}(t)\ dN_{vs}(t)\big)\,, \quad (7.9)$$

Similarly, we can parameterize the coefficients of the link creation intensity by a $K$-dimensional vector and write it with incorporating features of the node for computing the intensity:

$$\lambda^*_{us}(t) = (1 - A_{us}(t))(\boldsymbol{\mu}_u^\top \boldsymbol{x}_u + \boldsymbol{\alpha}_u^\top \boldsymbol{x}_u \sum_{v \in \mathcal{F}_u(t)} \kappa_{\omega_2}(t) \star dN_{vs}(t)). \quad (7.10)$$

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.

### 7.1.4 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 $\beta_s$ and $\eta_u$ in the information diffusion intensity given by Equation (2.8) only depend on the source and the actor, respectively. However, proceeding with this example, parameters can be made connection specific, *i.e.*, Equation (2.8) can be restated as

$$\gamma^*_{us}(t) = \mathbb{I}[u = s]\, \eta_{us} + \mathbb{I}[u \neq s]\, \beta_{us} \sum\nolimits_{v \in \mathcal{F}_u(t)} \kappa_{\omega_1}(t) \star \big(A_{uv}(t)\ dN_{vs}(t)\big)\,, \quad (7.11)$$

where $\eta_{us}$ is the base intensity of $u$ retweeting a tweet by $s$ and $\beta_{us}$ 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 $\beta_{svu}$ 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.

### 7.1.5 Future Works

In this thesis, 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 [228].

- Alternatively, one can use different triggering kernels for the Hawkes processes and learn them to capture finer details of temporal dynamics.

## 7.2 Point Process Intervention

Intervention and control is a well-studied problem in many application areas, however, three things set our work apart from most of the existing literature. First, although point processes have shown success in modeling social networks and health studies, however, less work has been done in order to steer the processes to a desired target. This urged for building an intervention paradigm for them. Second, the existence of the network and peer influence between nodes (*e.g.* social network users or disease) raises difficult challenges. An intervention not only can affect the direct node but has influence on neighboring nodes via propagation. The spread of influence can include directly connected nodes as well as a broader span. Third, the problem of scale and curse of dimensionality in multi dimensional point processes make inference and optimization very difficult, especially when the size of the mark space is large or even infinite. In such cases, we impose a latent structure (*e.g.* community structure, mixed membership and LDA models) or regularization constraints (*e.g.* sparsity, smoothing), or approximation methods (function approximation like

Q-learning) to tackle the problem. It's notable that, we have used point processes framework in intervention in other contexts as well. For example in [109] we find the best time to post in social networks in order to maximize the visibility of a post to followers. Furthermore, in [88] we tackle invasive species management problem in which a few cells of landscape are going to be chosen for complete removal of invasive plant.

### 7.2.1 Topology Management

Activity shaping can also be achieved by manipulating the topology of the diffusion networks. From the relation $\boldsymbol{\eta}(t) = \boldsymbol{\Psi}(t)\boldsymbol{\mu}$, we observe that the matrix $\boldsymbol{\Psi}(t)$ which stands for the network topology plays an important role. It is natural to manage the topology by optimizing the overall activities. More specifically, we can formulate $\text{minimize}_{\boldsymbol{A}(t)}\ U(\boldsymbol{\eta})$ subject to $\boldsymbol{\eta}(t) = \boldsymbol{\Psi}(t)\boldsymbol{\mu}$. Note that the induced problem will not necessarily be convex, however, we can impose structural constraints such sparsity and low-rank constraints to simplify the learning process. On the other hand, we can manipulate the topology to optimize the influence of a networked system, e.g., bring a flu under control and propel a video to popularity. Specifically, the influence can be defined by the expected number of active entities via influence function:

$$\sigma(\mathcal{A}, \boldsymbol{A}, T) = \mathbb{E}[\sum_{i \in \mathcal{V}} \mathbb{I}\{t_i \leq T\}] = \sum_{i \in \mathcal{V}} \Pr(g_i(\mathcal{A}, \boldsymbol{A})) \leq T,$$

where, $\mathcal{A}$ is a set of source entities, $\boldsymbol{A}$ is the infectivity matrix, and the transformation $g_i(\cdot)$ is the value of shortest-path minimization. The influence maximization problem can be formulated as $\text{maximize}_{\boldsymbol{A}}\ \sigma(\mathcal{A}, \boldsymbol{A}, T)$. The influence optimization problem can lead to a submodular optimization problem and be approximately solved using randomized algorithms with error bounds. Similarly, we can formulate influence minimization problems as well. Furthermore, structural constraints such as sparsity and low-rank constraints can be imposed on $\boldsymbol{\Psi}$ to satisfy certain manipulation requirements. For topology management with evolving networks, it is more important to learn a time-dependent infectivity matrix change $\Delta\boldsymbol{A}(t)$. Under either activity shaping or influence optimization framework, $\Delta\boldsymbol{A}(t)$

can be modeled using either parametric or nonparametric approaches. For example, we can parametrize $\Delta \boldsymbol{A}_{jj'}(t) = \gamma^T x_{jj'}(t)$, where $x_{jj'}(t)$ is a known time-varying feature vector of dimension $j$ and $j'$ at time $t$. The nonparametric modeling can be achieved by assume $\Delta \boldsymbol{A}_{jj'}(t) = \sum_{\ell=1}^{L} \lambda_{jj'}^{\ell} \sigma_\ell(t)$, where $\{\sigma_\ell(t)\}$ is a set of nonparametric basis functions that needs to be learned from cascade data. In addition, structural constraints such as sparsity and low-rank constraints can be imposed on $\Delta \boldsymbol{A}(t)$.

### 7.2.2 Model-Free Intervention Models

When the dynamics of the environment (random functions $f$ and $g$) is unknown reinforcement learning (RL) [185, 46] framework needs to be adapted for the point process setting we consider. In what follows we propose a few cases and algorithms we are considering in the development of the point process intervention framework.

Q-LEARNING. The idea is to approximate the expected cost by a linear term and use the regression algorithms to learn the parameters. Define the *action-value function*

$$Q^{\pi}(s, a) = \mathbb{E}[\sum_{k=1} \gamma^k g_k(s_k, \pi(s_k)) | s_0 = s, a_0 = a].$$

Finding the optimal policy then reduces to finding the optimum value function. Since the state and action space both are (infinity) large we use approximation:

$$Q_\theta(s, a) = \theta^1 h_1(s, a) + \theta^2 h_2(s, a) + \ldots + \theta^d h_d(s, a),$$

where, $h_i(s, a)$ is a feature extracted from states and actions and $\theta = (\theta^1, \ldots, \theta^d)$ is a set of parameters. $\theta_k$ is the estimate of $\theta$ at the $k^{th}$ stage. To update it we first find the temporal difference:

$$\delta_{k+1}(Q_{\theta_k}) = g_{k+1}(s_{k+1}, a_{k+1}) + \gamma Q_{\theta_k}(s_{k+1}, a'_{k+1}) - Q_{\theta_k}(s_k, a_k),$$

and then

$$\theta_{k+1} = \theta_k + \alpha_k \delta_{k+1}(Q_{\theta_k}) \nabla_\theta Q_{\theta_k}(s_k, a_k).$$

The features capture the interaction of intervention for mark $m$ and state of mark $m'$ with

an offset of $l$, e.g., $h_i(s,a) = a_{k-l}^m s_k^{m'}$. Also one can consider the effect of time delay directly $h_i(s,a) = a_{k-l}^m s_k^{m'} e^{-\omega l}$. Related to Q-Learning is the class of off-policy evaluation algorithms [155, 189]. Developing efficient and robust off-policy evaluation methods [54] for intervention policies is an interesting direction for point processes networks. Better evaluation, besides providing a safe transition for deploying new policies, can lead to better policy optimization through policy iteration and has demonstrated promising results in advertisement and marketing too [127, 188].

POLICY GRADIENT. In the policy gradient methods instead of working with the expected reward or value functions we parameterize the policy and learn it directly: $a_k(t) = \mu + \int_0^t \Phi(t,s)\theta dN(s)$ where $\theta$ is vector of size $M$. The point process is then evolves with intensity $\lambda(t) = u(a_k(t))$ at the $k^{th}$ stage. The parameters are learned

$$\frac{d\theta_i(t)}{dt} = \alpha(g_k - b)\delta(t - t_i)e_i(t),$$

where, $e_i$ is updated by

$$\frac{de_i(t)}{dt} = -\frac{e_i}{\tau} + \frac{u'}{u}(N_i(t) - a_k(t)).$$

Policy gradient methods are beneficial in the continuous action space where $Q$-learning might fail.

SEMI-MARKOV PROCESSES. An intermediate step from discrete MDP to continuous MDP is the class of Semi-Markov Decision Processes (SMDP). They generalize MDPs by (1) modeling the user/patient's status evolution in continuous time; (2) allowing time spent in a particular state to follow an arbitrary distribution; and (3) state transitions happening following a process different form the process governing the treatment/decision epochs. Instead of a simple transition function $f(s,a)$ we assert that if the next state is $s'$, the waiting time $t$, until the transition from $s$ occurs follows the distribution $F_{ss'}(t,a)$. Furthermore, the probability of transition from state $s \in \mathcal{S}$ to another state $s' \in \mathcal{S}$ is $P_{ss'}(a)$ when treatment

$a \in \mathcal{A}$ is applied. For the *policy learning* we again define

$$V^{\pi} = \mathbb{E}\left[\int_0^{\infty} e^{-\alpha t} g(s(t), \pi(t))\, dt\right],$$

where $e^{-\alpha t}$ is a discount factor that manages the trade off the importance of the immediate and the delayed reward. A POSMDP with patient state space $\mathcal{S} = \{s_1, s_2, s_3\}$ and 2 treatment actions $\mathcal{A} = \{a_1, a_2\}$ and two observation types $\{o_1, o_2\}$. For action $a_1$ transition probabilities are depicted in orange while for $a_2$ they are depicted in green. For each state and observation type there is an associated intensity function governing how events happen.

A value-function equation and a continues version of Bellman optimality conditions referred to as Bellman-Hamilton-Jacobi tailored to our framework will be derived. Temporal Difference algorithm is one possibility to find the optimal policy. Having observed a transition from $s$ to $s'$ with sample reward $r(s, s', \pi(s))$ it updates

$$V^{(k+1)}(s) = V^{(k)}(s) + \beta_k \left(\frac{1 - e^{-\alpha\tau}}{\alpha} r(s, s', \pi(s)) + e^{-\alpha\tau} V^{(k)}(s') - V^{(k)}(s)\right),$$

where $\tau$ is waiting time and $\beta_k$ is the learning rate.

PARTIAL OBSERVATION. The point process intervention framework assumes we have direct knowledge of the user and patient state. In many realistic situations, however, one can not directly access the state, but receives an observation that stochastically depends on it. In this case, the patient's health or social network user status can be modeled as a Partially Observable Semi-Markov Decision Process (POSMDP). This paradigm extends the point process intervention framework by incorporating a set of observations $O$, and an observation model defined by $p(o|s, v)$ which is the probability that $o$ is observed when the patient with feature vector $v$ is in state $s$. Especially, the discrete nature of observations calls for a rich framework of point processes. Consider the point process with intensity function $\lambda_{s,o}(t)$. User/patient features can also be incorporated into the model as covariate, *e.g.*, $\lambda_{s,o}(t) = v^{\top}\mu_{s,o}$. Here is a proposed scheme for a solution. We need to infer the state of the patient/user from the received observations and execute treatments/actions. A usual representation for the knowledge about the system state is the so called *belief* value,

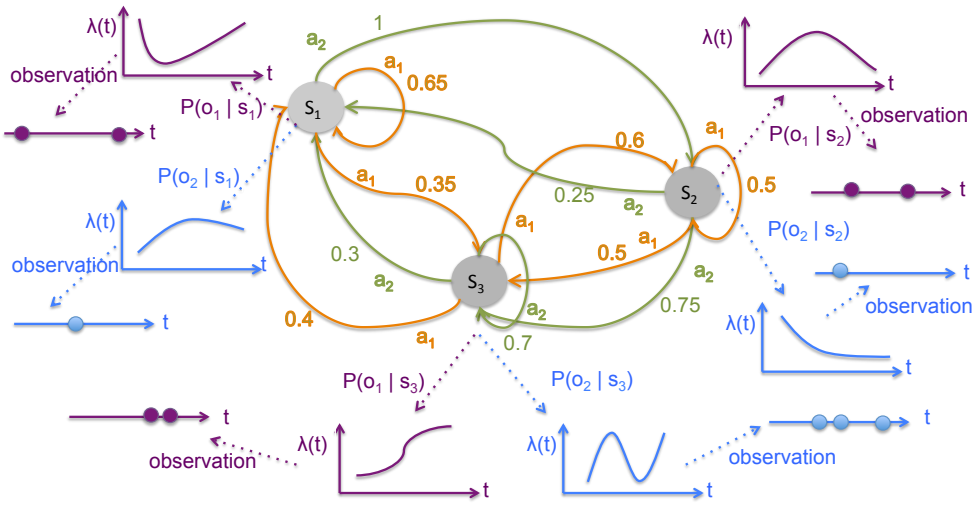



Figure 7.1: Partially Observable Semi Markov Decision Process for Point Processes

$b$, that is a probability distribution over the state space. The updated belief after executing treatment $a$ and observing $o$ is:

$$b^{a,o}(s') = p(s'|b,a)\frac{p(o|s',v)}{p(o|b,a)},$$

in which $p(s'|b,a)$ is the propagation of the belief $b$ through the transition model. One can think of this as an MDP with the pair of belief-state as the new state representation with a transition $p(b'|b,a) = p(o|b,a)$ if $b' = b^{a,o}$ and 0 otherwise. The new policy value function is:

$$V^*(b) = \max_{a\in\mathcal{A}} \sum_{o\in\mathcal{O}} p(o|b,a) \int_0^\infty \int_0^t e^{-\alpha t'} g(s,a)\, dt'\, d\eta_b(t,a) + \tag{7.12}$$

$$\sum_{o\in\mathcal{O}} p(o|b,a) \int_0^\infty e^{-\alpha t} V^*(b^{a,o})\, d\eta_b(t,a), \tag{7.13}$$

where $\eta_b(t,a)$ is the posterior of the waiting time for state transition estimated after the observation. The full framework is illustrated in Figure 7.1. These are potential ideas for future work in this area.

## 7.3 Generative Models of Point Process

Implicit generative models don't need any likelihood functions and provide samples that are are sharp and sufficiently representative of the population. Furthermore, they allow building highly accurate neural network classifiers. Wasserstein learning of deep generative point processes is a show-case of how can one learn a generative model for point processes indirectly via feedbacks from samples. This approach requires no prior knowledge about the underlying true process and can estimate it accurately across a wide scope of theoretical and real-world processes. Basically, the framework of optimal transport [199] could be leveraged to learn the implicit generative model via a variety of distance measures. For the future work, one can explore the connection of the WGAN with the optimal transport problem. Other possible distance metrics over the realizations of point processes, and more sophisticated transforms of point processes, particularly those that are causal, are also interesting venues for future work. Extending the current work to marked point processes and over structured spaces are other interesting ones.

This thesis focused on developing novel machine learning methodology and algorithms for high-dimensional asynchronous and interdependent event data arising from modern applications. Our premises are that asynchronous event temporal dynamics and observation features carry a great deal of information about the behavior of entities and their interactions we care about modeling, and a deeper understanding of the interplay between the dynamics of the events and the dynamics of the network structures will allow us to address the "*who will do what and when?*" question with time-varying exploratory and predictive models capable of solving real-world problems. Specifically, we proposed a framework based on point processes which explicitly model the rate of event occurrence as a function of timing and features of previous events for evolving diffusion networks. Furthermore, our modeling framework can explicitly take into account latent variables, low *intrinsic* dimensionality and sparsity of the datasets, and support subsequent decision making where the effects of

the decisions are assessed within a time upper bound. The flexibility and strength of point process combined with efficient learning and inference procedures turn it to an outstanding tool to analyze network event sequences.

## REFERENCES


[1] O. Aalen, O. Borgan, and H. Gjessing, *Survival and event history analysis: a process point of view*. Springer Science & Business Media, 2008.

[2] R. P. Adams, I. Murray, and D. J. C. MacKay, "Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities," in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, New York, New York, USA: ACM Press, 2009, pp. 1–8, ISBN: 9781605585161.

[3] N. Agarwal, H. Liu, L. Tang, and P. S. Yu, "Identifying the influential bloggers in a community," in *Proceedings of the 2008 international conference on web search and data mining*, ACM, 2008, pp. 207–218.

[4] A. H. Al-Mohy and N. J. Higham, "Computing the action of the matrix exponential, with an application to exponential integrators," *SIAM journal on scientific computing*, vol. 33, no. 2, pp. 488–511, 2011.

[5] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," National Bureau of Economic Research, Tech. Rep., 2017.

[6] D. Antoniades and C. Dovrolis, "Co-evolutionary dynamics in social networks: A case study of twitter," *arXiv preprint arXiv:1309.6001*, 2013.

[7] A. Antos, C. Szepesvári, and R. Munos, "Value-iteration based fitted policy iteration: Learning with a single trajectory," in *Approximate Dynamic Programming and Reinforcement Learning, 2007. ADPRL 2007. IEEE International Symposium on*, IEEE, 2007, pp. 330–337.

[8] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," in *NIPS 2016 Workshop on Adversarial Training. In review for ICLR*, vol. 2016, 2017.

[9] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv:1701.07875*, 2017.

[10] L. Backstrom, P. Boldi, M. Rosa, J. Ugander, and S. Vigna, "Four degrees of separation," in *Proceedings of the 4th Annual ACM Web Science Conference*, 2012, pp. 33–42.

[11] E. Bacry, S. Delattre, M. Hoffmann, and J.-F. Muzy, "Some limit theorems for hawkes processes and application to financial statistics," *Stochastic Processes and their Applications*, vol. 123, no. 7, pp. 2475–2499, 2013.

[12] E. Bacry and J.-F. Muzy, "Hawkes model for price and trades high-frequency dynamics," *Quantitative Finance*, vol. 14, no. 7, pp. 1147–1166, 2014.

[13] E. Bacry and J.-F. Muzy, "Second order statistics characterization of hawkes processes and non-parametric estimation," *arXiv preprint arXiv:1401.0903*, 2014.

[14] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: Quantifying influence on twitter," in *WSDM*, 2011, pp. 65–74.

[15] A. Barrat, M. Barthelemy, and A. Vespignani, *Dynamical processes on complex networks*. Cambridge University Press, 2008.

[16] D. P. Bertsekas, *Dynamic programming and optimal control*, 2. 1995, vol. 1.

[17] S. Bhagat, A. Goyal, and L. V. Lakshmanan, "Maximizing product adoption in social networks," in *Proceedings of the fifth ACM international conference on Web search and data mining*, ACM, 2012, pp. 603–612.

[18] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in *International Workshop on Web and Internet Economics*, Springer, 2007, pp. 306–311.

[19] R. Bhatt, V. Chaoji, and R. Parekh, "Predicting product adoption in large-scale social networks," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, ACM, 2010, pp. 1039–1048.

[20] P. Bhattacharya, T. Q. Phan, and E. M. Airoldi, "Analyzing the co-evolution of network structure and content generation in online social networks," *ECIS 2015 Completed Research Papers*, p. 18, 2015.

[21] D. Bloembergen, B. R. Sahraei, H. Bou-Ammar, K. Tuyls, and G. Weiss, "Influencing social networks: An optimal control study.," in *ECAI*, 2014, pp. 105–110.

[22] C. Blundell, J. Beck, and K. A. Heller, "Modelling reciprocating relationships with hawkes processes," in *nips*, 2012.

[23] S. Bornholdt and T. Rohlf, "Topological evolution of dynamical networks: Global criticality from local dynamics," *Physical Review Letters*, vol. 84, no. 26, p. 6114, 2000.

[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, England: Cambridge University Press, 2004.

[25] R. Bracewell, "The fourier transform and iis applications," *New York*, vol. 5, 1965.

[26] S. J. Bradtke and A. G. Barto, "Linear least-squares algorithms for temporal difference learning," *Machine Learning*, vol. 22, no. 1, pp. 33–57, 1996.

[27] U. Brandes, J. Lerner, and T. A. Snijders, "Networks evolving step by step: Statistical analysis of dyadic event data," in *Social Network Analysis and Mining, 2009. ASONAM'09. International Conference on Advances in*, IEEE, 2009, pp. 200–205.

[28] C. T. Butts, "A relational event framework for social action," *Sociological Methodology*, vol. 38, no. 1, pp. 155–200XI, 2008.

[29] J. H. Cha and M. Finkelstein, "Poisson shock model with applications to preventive maintenance," in *Point Processes for Reliability Analysis*, Springer, 2018, pp. 169–209.

[30] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring User Influence in Twitter: The Million Follower Fallacy," in *ICWSM*, 2010.

[31] D. Chakrabarti, Y. Zhan, and C. Faloutsos, "R-mat: A recursive model for graph mining," *Computer Science Department*, p. 541, 2004.

[32] D. Chen, L. Lü, M.-S. Shang, Y.-C. Zhang, and T. Zhou, "Identifying influential nodes in complex networks," *Physica a: Statistical mechanics and its applications*, vol. 391, no. 4, pp. 1777–1787, 2012.

[33] P.-Y. Chen, S.-M. Cheng, and K.-C. Chen, "Optimal control of epidemic information dissemination over networks," *Cybernetics, IEEE Transactions on*, vol. 44, no. 12, pp. 2316–2328, 2014.

[34] S. Chen, A. Shojaie, E. Shea-Brown, and D. Witten, "The multivariate hawkes process in high dimensions: Beyond mutual excitation," *arXiv preprint arXiv:1707.04928*, 2017.

[35] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2009, pp. 199–208.

[36] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec, "Can cascades be predicted?" In *Proceedings of the 23rd international conference on World wide web*, 2014, pp. 925–936.

[37] S.-N. Chow, X. Ye, H. Zha, and H. Zhou, "Influence prediction for continuous-time information propagation on networks," *arXiv preprint arXiv:1512.05417*, 2015.

[38] M. Costa, C. Graham, L. Marsalle, and V. C. Tran, "Renewal in hawkes processes with self-excitation and inhibition," *arXiv preprint arXiv:1801.04645*, 2018.

[39] M. Cuturi and M. Blondel, "Soft-dtw: A differentiable loss function for time-series," *arXiv preprint arXiv:1703.01541*, 2017.

[40] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes*. Springer Science & Business Media, 2007.

[41] A. De, I. Valera, N. Ganguly, S. Bhattacharya, and M. G. Rodriguez, "Modeling opinion dynamics in diffusion networks," *arXiv preprint arXiv:1506.05474*, 2015.

[42] G. F. De Arruda, A. L. Barbieri, P. M. Rodríguez, F. A. Rodrigues, Y. Moreno, and L. da Fontoura Costa, "Role of centrality for the identification of influential spreaders in complex networks," *Physical Review E*, vol. 90, no. 3, p. 032 812, 2014.

[43] L. Decreusefond, M. Schulte, C. Thäle, *et al.*, "Functional poisson approximation in kantorovich–rubinstein distance with applications to u-statistics and stochastic geometry," *The Annals of Probability*, vol. 44, no. 3, pp. 2147–2197, 2016.

[44] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2001, pp. 57–66.

[45] P. Doreian and F. Stokman, *Evolution of social networks*. Routledge, 2013.

[46] K. Doya, "Reinforcement learning in continuous time and space," *Neural computation*, vol. 12, no. 1, pp. 219–245, 2000.

[47] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 1555–1564.

[48] N. Du, M. Farajtabar, A. Ahmed, A. J. Smola, and L. Song, "Dirichlet-hawkes processes with applications to clustering continuous-time document streams," in *KDD*, ACM, 2015.

[49] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha, "Scalable influence estimation in continuous-time diffusion networks," in *NIPS*, 2013.

[50] N. Du, L. Song, H. Woo, and H. Zha, "Uncover topic-sensitive information diffusion networks," in *Proceedings of the sixteenth international conference on artificial intelligence and statistics*, 2013, pp. 229–237.

[51] N. Du, L. Song, M. Yuan, and A. J. Smola, "Learning networks of heterogeneous influence," in *Advances in Neural Information Processing Systems*, 2012, pp. 2780–2788.

[52] N. Du, Y. Wang, N. He, J. Sun, and L. Song, "Time-sensitive recommendation from recurrent user activities," in *NIPS*, 2015.

[53] P. Erdos and A Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hungar. Acad. Sci*, vol. 5, pp. 17–61, 1960.

[54] M. Farajtabar, Y. Chow, and M. Ghavamzadeh, "More robust doubly robust off-policy evaluation," *arXiv preprint arXiv:1802.03493*, 2018.

[55] M. Farajtabar, N. Du, M. G. Rodriguez, I. Valera, H. Zha, and L. Song, "Shaping social activity by incentivizing users," in *Advances in neural information processing systems*, 2014, pp. 2474–2482.

[56] M. Farajtabar, M. Gomez-Rodriguez, N. Du, M. Zamani, H. Zha, and L. Song, "Back to the past: Source identification in diffusion networks from partially observed cascades," in *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

[57] M. Farajtabar, Y. Wang, M. Gomez-Rodriguez, S. Li, and H. Zha, "Coevolve: A joint point process model for information diffusion and network evolution," *Journal of Machine Learning Research*, vol. 18, pp. 1–49, 2017.

[58] M. Farajtabar, Y. Wang, M. G. Rodriguez, S. Li, H. Zha, and L. Song, "Coevolve: A joint point process model for information diffusion and network co-evolution," in *Advances in Neural Information Processing Systems*, 2015, pp. 1954–1962.

[59] M. Farajtabar, J. Yang, X. Ye, H. Xu, R. Trivedi, E. Khalil, S. Li, L. Song, and H. Zha, "Fake news mitigation via point process based intervention," in *International Conference on Machine Learning*, 2017, pp. 1097–1106.

[60] M. Farajtabar, X. Ye, S. Harati, L. Song, and H. Zha, "Multistage campaigning in social networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 4718–4726.

[61] G. B. Folland, *Real analysis: modern techniques and their applications*. John Wiley & Sons, 2013.

[62] Z. Gao, Y. Shi, and S. Chen, "Identifying influential nodes for efficient routing in opportunistic networks," *Journal of Communications*, vol. 10, no. 1, 2015.

[63] K. Garimella, A. Gionis, N. Parotsidis, and N. Tatti, "Balancing information exposure in social networks," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017, pp. 4663–4671.

[64] A. Ghosh, V. Kulharia, A. Mukerjee, V. Namboodiri, and M. Bansal, "Contextual rnn-gans for abstract reasoning diagram generation," *arXiv preprint arXiv:1609.09444*, 2016.

[65] R. Ghosh and K. Lerman, "The role of dynamic interactions in multi-scale analysis of network structure," *CoRR*, 2012.

[66] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[67] S. Goel, D. J. Watts, and D. G. Goldstein, "The structure of online diffusion networks," in *Proceedings of the 13th ACM conference on electronic commerce*, 2012, pp. 623–638.

[68] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi, "A survey of statistical network models," *Foundations and Trends® in Machine Learning*, vol. 2, no. 2, pp. 129–233, 2010.

[69] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU Press, 2012, vol. 3.

[70] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," in *ICML*, 2011.

[71] M. Gomez-Rodriguez, K. Gummadi, and B. Schoelkopf, "Quantifying Information Overload in Social Media and its Impact on Social Contagions," in *ICWSM*, 2014.

[72] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2010, pp. 1019–1028.

[73] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf, "Modeling information propagation with survival theory," in *International Conference on Machine Learning*, 2013, pp. 666–674.

[74] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv:1701.00160*, 2016.

[75] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[76] J. Gottfried and E. Shearer, *News use across social media platforms 2016*, Pew Research Center, 2016.

[77] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "Learning influence probabilities in social networks," in *Proceedings of the third ACM international conference on Web search and data mining*, ACM, 2010, pp. 241–250.

[78] A. Goyal, F. Bonchi, L. V. Lakshmanan, and S. Venkatasubramanian, "Approximation analysis of influence spread in social networks," *arXiv preprint arXiv:1008.2005*, 2010.

[79] M. Granovetter, "The strength of weak ties," *American journal of sociology*, pp. 1360–1380, 1973.

[80] T. Gross and B. Blasius, "Adaptive coevolutionary networks: A review," *Journal of The Royal Society Interface*, vol. 5, no. 20, pp. 259–271, 2008.

[81] T. Gross, C. J. D. DLima, and B. Blasius, "Epidemic dynamics on an adaptive network," *Physical review letters*, vol. 96, no. 20, p. 208 701, 2006.

[82] T. Gross and H. Sayama, *Adaptive networks*. Springer, 2009.

[83] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, "Information diffusion through blogspace," in *Proceedings of the 13th international conference on World Wide Web*, ACM, 2004, pp. 491–501.

[84] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.

[85] A. Gunawardana, C. Meek, and P. Xu, "A model for temporal dependencies in event streams," in *Advances in Neural Information Processing Systems*, 2011, pp. 1962–1970.

[86] F. Guo, C. Blundell, H. Wallach, K. Heller, and U. Gatsby Unit, "The bayesian echo chamber: Modeling social influence via linguistic accommodation," in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015, pp. 315–323.

[87] F. Guo, C. Blundell, H. M. Wallach, K. A. Heller, and U. G. Unit, "The bayesian echo chamber: Modeling social influence via linguistic accommodation.," in *AISTATS*, 2015.

[88] A. Gupta, M. Farajtabar, B. Dilkina, and H. Zha, "Hawkes processes for invasive species modeling and management," *arXiv preprint arXiv:1712.04386*, 2017.

[89] E. C. Hall and R. M. Willett, "Tracking dynamic point processes on networks," *arXiv preprint arXiv:1409.0031*, 2014.

[90] N. R. Hansen, P. Reynaud-Bouret, and V. Rivoirard, "Lasso and probabilistic inequalities for multivariate point processes," *Bernoulli*, vol. 21, no. 1, pp. 83–143, 2015.

[91] F. B. Hanson, *Applied stochastic processes and control for Jump-diffusions: modeling, analysis, and computation*. Siam, 2007, vol. 13.

[92] D. Hatano, T. Fukunaga, and K.-i. Kawarabayashi, "Adaptive budget allocation for maximizing influence of advertisements.," in *IJCAI*, 2016, pp. 3600–3608.

[93] A. G. Hawkes, "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, pp. 83–90, 1971.

[94] A. G. Hawkes and D. Oakes, "A cluster process representation of a self-exciting process," *Journal of Applied Probability*, vol. 11, no. 3, pp. 493–503, 1974.

[95] X. He, T. Rekatsinas, J. Foulds, L. Getoor, and Y. Liu, "Hawkestopic: A joint model for network inference and topic modeling from text-based cascades," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 871–880.

[96] D. R. Heise, "Modeling event structures*," *Journal of Mathematical Sociology*, vol. 14, no. 2-3, pp. 139–169, 1989.

[97] O. Hijab, *Introduction to calculus and classical analysis*. Springer, 2007.

[98] T. Hogg and K. Lerman, "Social dynamics of digg," *EPJ Data Science*, vol. 1, no. 1, pp. 1–26, 2012.

[99] P. Holme, "Modern temporal network theory: A colloquium," *arXiv:1508.01303*, 2015.

[100] S. A. Hosseini, K. Alizadeh, A. Khodadadi, A. Arabzadeh, M. Farajtabar, H. Zha, and H. R. Rabiee, "Recurrent poisson factorization for temporal recommendation," *arXiv preprint arXiv:1703.01442*, 2017.

[101] Y. Hu and B. Oksendal, "Partial information linear quadratic control for jump diffusions," *SIAM Journal on Control and Optimization*, vol. 47, no. 4, pp. 1744–1761, 2008.

[102] D. Hunter, P. Smyth, D. Q. Vu, and A. U. Asuncion, "Dynamic egocentric models for citation networks," in *Proceedings of the 28th International Conference on Machine Learning*, 2011, pp. 857–864.

[103] D. R. Hunter and K. Lange, "A tutorial on mm algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[104] F. Huszár, "How (not) to train your generative model: Scheduled sampling, likelihood, adversary?" *arXiv preprint arXiv:1511.05101*, 2015.

[105] V. Isham and M. Westcott, "A self-correcting point process," *Stochastic Processes and Their Applications*, vol. 8, no. 3, pp. 335–347, 1979.

[106] T. Iwata, A. Shah, and Z. Ghahramani, "Discovering latent influence in online social activities via shared cascade poisson processes," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2013, pp. 266–274.

[107] K. Iwayama, Y. Hirata, and K. Aihara, "Definition of distance for nonlinear time series analysis of marked point process data," *Physics Letters A*, vol. 381, no. 4, pp. 257–262, 2017.

[108] K. Kandhway and J. Kuri, "Campaigning in heterogeneous social networks: Optimal control of si information epidemics," 2015.

[109] M. R. Karimi, E. Tavakoli, M. Farajtabar, L. Song, and M. Gomez-Rodriguez, "Smart broadcasting: Do you want to be seen?" *arXiv preprint arXiv:1605.06855*, 2016.

[110] A. Karnik and P. Dayama, "Optimal control of information epidemics," in *Communication Systems and Networks (COMSNETS), 2012 Fourth International Conference on*, 2012, pp. 1–7.

[111] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2003, pp. 137–146.

[112] M. Kim, R. Jurdak, and D. Paini, "Modeling reflexivity of social systems in disease spread," *arXiv preprint arXiv:1711.06359*, 2017.

[113] J. F. C. Kingman, *Poisson processes*. Wiley Online Library, 1993.

[114] S. Kolouri, S. R. Park, M. Thorpe, D. Slepcev, and G. K. Rohde, "Optimal mass transport: Signal processing and machine-learning applications," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 43–59, 2017.

[115] G. Krishnamurthy, N. Majumder, S. Poria, and E. Cambria, "A deep learning approach for multimodal deception detection," *arXiv preprint arXiv:1803.00344*, 2018.

[116] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?" In *Proceedings of the 19th International Conference on World Wide Web*, Raleigh, North Carolina, USA: ACM, 2010, pp. 591–600, ISBN: 978-1-60558-799-8.

[117] M. G. Lagoudakis and R. Parr, "Least-squares policy iteration," *Journal of machine learning research*, vol. 4, no. Dec, pp. 1107–1149, 2003.

[118] P. Lagrée, O. Cappé, B. Cautis, and S. Maniu, "Effective large-scale online influence maximization," in *International Conference on Data Mining*, 2017.

[119] R. Lemonnier and N. Vayatis, "Nonparametric markovian learning of triggering kernels for mutually exciting and mutually inhibiting multivariate hawkes processes," in *Machine Learning and Knowledge Discovery in Databases*, 2014, pp. 161–176.

[120] K. Lerman, A. Galstyan, G. Ver Steeg, and T. Hogg, "Social mechanics: An empirically grounded science of social media," in *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[121] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2009, pp. 497–506.

[122] J. Leskovec, L. Backstrom, and J. Kleinberg, "Meme-tracking and the dynamics of the news cycle," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2009, pp. 497–506.

[123] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins, "Microscopic evolution of social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2008, pp. 462–470.

[124] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *JMLR*, vol. 11, no. Feb, pp. 985–1042, 2010.

[125] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: Densification laws, shrinking diameters and possible explanations," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, ACM, 2005, pp. 177–187.

[126] E. Lewis and G. Mohler, "A nonparametric em algorithm for multiscale hawkes processes," *Journal of Nonparametric Statistics*, 2011.

[127] L. Li, W. C. an d J. Langford, and X. Wang, "Unbiased offline evaluation of contextual bandit-based news article recommendation algorithms," in *Proceedings of the 4th International Conference on Web Search and Data Mining*, 2011, pp. 297–306.

[128] S. Li, Y. Xie, M. Farajtabar, A. Verma, and L. Song, "Detecting weak changes in dynamic events over networks," *arXiv preprint arXiv:1603.08981*, 2016.

[129] W. Lian, V. A. Rao, B. Eriksson, and L. Carin, "Modeling correlated arrival events with latent semi-markov processes," in *Proceedings of the International Conference on Machine Learning*, 2014.

[130] W. Lian, R. Henao, V. Rao, J. Lucas, and L. Carin, "A multitask point process predictive model," in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 2030–2038.

[131] S. W. Linderman and R. P. Adams, "Discovering latent network structure in point process data.," in *ICML*, 2014, pp. 1413–1421.

[132] S. W. Linderman, Y. Wang, and D. M. Blei, "Bayesian inference for latent hawkes processes,"

[133] T. J. Liniger, "Multivariate hawkes processes," PhD thesis, Diss. Eidgenössische Technische Hochschule ETH Zürich, 2009.

[134] C. Lloyd, T. Gunter, M. A. Osborne, and S. J. Roberts, "Variational inference for gaussian process modulated poisson processes," in *ICML*, 2015.

[135] D. Marsan and O. Lengliné, "Extending earthquakes' reach through cascading.," *Science (New York, N.Y.)*, vol. 319, no. 5866, pp. 1076–9, Feb. 2008.

[136] D. Marsan and O. Lengline, "Extending earthquakes' reach through cascading," *Science*, vol. 319, no. 5866, pp. 1076–1079, 2008.

[137] H. Mei and J. M. Eisner, "The neural hawkes process: A neurally self-modulating multivariate point process," in *Advances in Neural Information Processing Systems*, 2017, pp. 6757–6767.

[138] A. Mirchev and S.-A. Ahmadi, "Classification of sparsely labeled spatio-temporal data through semi-supervised adversarial learning," *arXiv preprint arXiv:1801.08712*, 2018.

[139] T. Mitra, G Wright, and E. Gilbert, "A parsimonious language model of social media credibility across disparate events," in *Proc. CSCW*, 2017.

[140] O. Mogren, "C-rnn-gan: Continuous recurrent neural networks with adversarial training," *arXiv preprint arXiv:1611.09904*, 2016.

[141] M. G. Moore and M. A. Davenport, "Analysis of wireless networks using hawkes processes," in *Signal Processing Advances in Wireless Communications (SPAWC), 2016 IEEE 17th International Workshop on*, IEEE, 2016, pp. 1–5.

[142] M. G. Moore and M. A. Davenport, "A hawkes' eye view of network information flow," in *Statistical Signal Processing Workshop (SSP), 2016 IEEE*, IEEE, 2016, pp. 1–5.

[143] A. Mosseri, *Addressing hoaxes and fake news*, 2016.

[144] S. A. Myers and J. Leskovec, "On the convexity of latent social network inference," in *NIPS*, 2010.

[145] S. A. Myers and J. Leskovec, "The bursty dynamics of the twitter information network," in *WWW14*, 2014, pp. 913–924.

[146] M. Newman, *Networks: an introduction*. Oxford University Press, 2010.

[147] A. Nguyen, J. Yosinski, Y. Bengio, A. Dosovitskiy, and J. Clune, "Plug & play generative networks: Conditional iterative generation of images in latent space," *arXiv preprint arXiv:1612.00005*, 2016.

[148] Y Ogata, "On lewis' simulation method for point processes," *IEEE Transactions on Information Theory*, vol. 27, no. 1, pp. 23–31, 1981.

[149] Y. Ogata, "Seismicity analysis through point-process modeling: A review," *Pure & Applied Geophysics*, vol. 155, no. 2-4, p. 471, 1999.

[150] A. P. Parikh, A. Gunawardana, and C. Meek, "Conjoint modeling of temporal dependencies in event streams," *BMAW-12 Preface*, 2012.

[151] P. O. Perry and P. J. Wolfe, "Point process modeling for directed interaction networks," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 75, no. 5, pp. 821–849, 2013.

[152] G. Peyré, M. Cuturi, *et al.*, "Computational optimal transport," Tech. Rep., 2017.

[153] T. Q. Phan and E. M. Airoldi, "A natural experiment of social network formation and dynamics," *Proceedings of the National Academy of Sciences*, vol. 112, no. 21, pp. 6595–6600, 2015.

[154] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, "A stylometric inquiry into hyperpartisan and fake news," *arXiv preprint arXiv:1702.05638*, 2017.

[155] D. Precup, R. S. Sutton, and S. P. Singh, "Eligibility traces for off-policy policy evaluation.," in *ICML*, Citeseer, 2000, pp. 759–766.

[156] Z. Qin and C. R. Shelton, "Auxiliary gibbs sampling for inference in piecewise-constant conditional intensity models.," in *UAI*, 2015, pp. 722–731.

[157] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[158] H. Ramlau-Hansen, "Smoothing Counting Process Intensities by Means of Kernel Functions," *The Annals of Statistics*, vol. 11, no. 2, pp. 453–466, Jun. 1983.

[159] J. G. Rasmussen, "Temporal point processes the conditional intensity function," 2011.

[160] P. Reynaud-Bouret and S. Schbath, "Adaptive estimation for hawkes processes; application to genome analysis," *The Annals of Statistics*, vol. 38, no. 5, pp. 2781–2822, 2010.

[161] M Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2002, pp. 61–70.

[162] M.-A. Rizoiu, T. Graham, R. Zhang, Y. Zhang, R. Ackland, and L. Xie, "# debatenight: The role and influence of socialbots on twitter during the 1st us presidential debate," *arXiv preprint arXiv:1802.09808*, 2018.

[163] M.-A. Rizoiu, L. Xie, S. Sanner, M. Cebrian, H. Yu, and P. Van Henteryck, "Expecting to be hip: Hawkes intensity processes for social media popularity,"

[164] M. G. Rodriguez, J. Leskovec, and B. Schölkopf, "Modeling information propagation with survival theory," *arXiv preprint arXiv:1305.3616*, 2013.

[165] M. Rodriguez and B. Schölkopf, "Influence maximization in continuous time diffusion networks," in *ICML*, 2012.

[166] D. M. Romero and J. Kleinberg, "The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter," in *ICWSM*, 2010.

[167] S. M. Ross, *Introduction to Probability Models, Tenth Edition*. Academic Press, Inc., 2011, ISBN: 978-0-12-375686-2.

[168] N. Ruchansky, S. Seo, and Y. Liu, "Csi: A hybrid deep model for fake news," *arXiv preprint arXiv:1703.06959*, 2017.

[169] Y. Saad and M. H. Schultz, "Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on scientific and statistical computing*, vol. 7, no. 3, pp. 856–869, 1986.

[170] K. Saito, R. Nakano, and M. Kimura, "Prediction of information diffusion probabilities for independent cascade model," in *Knowledge-based intelligent information and engineering systems*, Springer, 2008, pp. 67–75.

[171] M. Sameki, T. Zhang, L. Ding, M. Betke, and D. Gurari, "Crowd-o-meter: Predicting if a person is vulnerable to believe political claims," 2017.

[172] Y.-L. K. Samo and S. Roberts, "Scalable nonparametric bayesian inference on point processes with gaussian processes," in *ICML*, 2015.

[173] J. Sampson, F. Morstatter, L. Wu, and H. Liu, "Leveraging the implicit structure within social media for emergent rumor detection," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ACM, 2016, pp. 2377–2382.

[174] H. Sayama, I. Pestov, J. Schmidt, B. J. Bush, C. Wong, J. Yamanoi, and T. Gross, "Modeling complex systems with adaptive networks," *Computers & Mathematics with Applications*, vol. 65, no. 10, pp. 1645–1664, 2013.

[175] D. Schuhmacher and A. Xia, "A new metric between distributions of point processes," *Advances in applied probability*, vol. 40, no. 3, pp. 651–672, 2008.

[176] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," *ACM SIGKDD Explorations Newsletter*, vol. 19, no. 1, pp. 22–36, 2017.

[177] C. Silverman, *This analysis shows how viral fake election news stories outperformed real news on facebook*, 2016.

[178] A. Simma and M. I. Jordan, "Modeling events with cascades of poisson processes," *arXiv preprint arXiv:1203.3516*, 2012.

[179] P. Singer, C. Wagner, and M. Strohmaier, "Factors influencing the co-evolution of social and content networks in online social media," in *Modeling and Mining Ubiquitous Social Media*, Springer, 2012, pp. 40–59.

[180] T. A. Snijders, "Siena: Statistical modeling of longitudinal network data," in *Encyclopedia of Social Network Analysis and Mining*, Springer, 2014, pp. 1718–1725.

[181] T. A. Snijders and S. Luchini, "Statistical methods for network dynamics," in *Proceedings of the XLIII Scientific Meeting, Italian Statistical Society*, CLEUP, 2006.

[182] J. Solomon, "Computational optimal transport," 2017.

[183] A. Stomakhin, M. B. Short, and A. L. Bertozzi, "Reconstruction of missing data in social networks based on temporal patterns of interactions," *Inverse Problems*, vol. 27, no. 11, p. 115 013, 2011.

[184] T. Sun, W. Chen, Z. Liu, Y. Wang, X. Sun, M. Zhang, and C.-Y. Lin, "Participation maximization based on social influence in online discussion forums.," in *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2011.

[185] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st. Cambridge, MA, USA: MIT Press, 1998, ISBN: 0262193981.

[186] B. Tabibian, I. Valera, M. Farajtabar, L. Song, B. Schölkopf, and M. Gomez-Rodriguez, "Distilling information reliability and source trustworthiness from digital traces," *arXiv preprint arXiv:1610.07472*, 2016.

[187] L. Theis, A. v. d. Oord, and M. Bethge, "A note on the evaluation of generative models," *arXiv preprint arXiv:1511.01844*, 2015.

[188] G. Theocharous, P. Thomas, and M. Ghavamzadeh, "Personalized ad recommendation systems for life-time value optimization with guarantees," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 2015, pp. 1806–1812.

[189] P. S. Thomas, G. Theocharous, and M. Ghavamzadeh, "High-confidence off-policy evaluation.," in *AAAI*, 2015, pp. 3000–3006.

[190] I. M. Toke, ""Market making" behaviour in an order book model and its impact on the bid-ask spread," *Arxiv*, p. 17, Mar. 2010. arXiv: 1003.3796.

[191] L. Tran, M. Farajtabar, L. Song, and H. Zha, "Netcodec: Community detection from individual activities," in *Proceedings of the 2015 SIAM International Conference on Data Mining*, SIAM, 2015, pp. 91–99.

[192] R. Trivedi, M. Farajtabar, Y. Wang, H. Dai, H. Zha, and L. Song, "Know-evolve: Deep reasoning in temporal knowledge graphs," *arXiv preprint arXiv:1705.05742*, 2017.

[193] Twitter, *Types of tweets and where they appear*, 2016.

[194] J. Ugander, L. Backstrom, and J. Kleinberg, "Subgraph frequencies: Mapping the empirical and extremal geography of large graph collections," in *Proceedings of the 22nd international conference on World Wide Web*, International World Wide Web Conferences Steering Committee, 2013, pp. 1307–1318.

[195] I. Valera and M. Gomez-Rodriguez, "Modeling adoption and usage of competing products," in *2015 IEEE International Conference on Data Mining*, 2015.

[196] S. Vaswani, B. Kveton, Z. Wen, M. Ghavamzadeh, L. V. S. Lakshmanan, and M. Schmidt, "Model-independent online learning for influence maximization," in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, International Convention Centre, Sydney, Australia: PMLR, 2017, pp. 3530–3539.

[197] M. Vergeer, L. Hermans, and S. Sams, "Online social networks and micro-blogging in political campaigning the exploration of a new campaign tool and a new campaign style," *Party Politics*, vol. 19, no. 3, pp. 477–501, 2013.

[198] M. Verstraete, D. E. Bambauer, and J. R. Bambauer, "Identifying and countering fake news," 2017.

[199] C. Villani, *Optimal transport: old and new*. Springer Science & Business Media, 2008, vol. 338.

[200] D. Q. Vu, A. U. Asuncion, D. R. Hunter, and P. Smyth, "Dynamic egocentric models for citation networks," in *ICML*, 2011.

[201] D. Q. Vu, D. Hunter, P. Smyth, and A. U. Asuncion, "Continuous-time regression models for longitudinal networks," in *Advances in Neural Information Processing Systems*, 2011, pp. 2492–2500.

[202] E. Wang, J. Silva, R. Willett, and L. Carin, "Time-evolving modeling of social networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, IEEE, 2011, pp. 2184–2187.

[203] Q. Wang and W. Chen, "Tighter regret bounds for influence maximization and other combinatorial semi-bandits with probabilistically triggered arms," *arXiv preprint arXiv:1703.01610*, 2017.

[204] Y. Wang, E. Theodorou, A. Verma, and L. Song, "Steering opinion dynamics in information diffusion networks," *arXiv preprint arXiv:1603.09021*, 2016.

[205] Y. Wang, D. J. Miller, K. Poskanzer, Y. Wang, L. Tian, and G. Yu, "Graphical time warping for joint alignment of multiple curves," in *Advances in Neural Information Processing Systems*, 2016, pp. 3648–3656.

[206] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[207] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "Twitterrank: Finding topic-sensitive influential twitterers," in *Proceedings of the third ACM international conference on Web search and data mining*, ACM, 2010, pp. 261–270.

[208] L. Weng, J. Ratkiewicz, N. Perra, B. Gonçalves, C. Castillo, F. Bonchi, R. Schifanella, F. Menczer, and A. Flammini, "The role of information diffusion in the evolution of social networks," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2013, pp. 356–364.

[209] D. M. West, *Air Wars: Television Advertising and Social Media in Election Campaigns, 1952-2012: Television Advertising and Social Media in Election Campaigns, 1952-2012*. Sage, 2013.

[210] L. Wu and H. Liu, "Tracing fake-news footprints: Characterizing social media messages by how they propagate," 2018.

[211] S. Xiao, M. Farajtabar, X. Ye, J. Yan, L. Song, and H. Zha, "Wasserstein learning of deep generative point process models," in *Advances in Neural Information Processing Systems*, 2017, pp. 3247–3257.

[212] S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha, "Joint modeling of event sequence and time series with attentional twin recurrent neural networks," *arXiv preprint arXiv:1703.08524*, 2017.

[213] S. Xiao, J. Yan, X. Yang, H. Zha, and S. M. Chu, "Modeling the intensity function of point process via recurrent neural networks.," in *AAAI*, 2017, pp. 1597–1603.

[214] H. Xu, L. Carin, and H. Zha, "Learning registered point processes from idiosyncratic observations," *arXiv preprint arXiv:1710.01410*, 2017.

[215] H. Xu, X. Chen, and L. Carin, "Superposition-assisted stochastic optimization for hawkes processes," *arXiv preprint arXiv:1802.04725*, 2018.

[216] H. Xu, M. Farajtabar, and H. Zha, "Learning granger causality for hawkes processes," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 1717–1726.

[217] Q. Xu, D. Yang, J. Tan, A. Sawatzky, and M. A. Anastasio, "Accelerated fast iterative shrinkage thresholding algorithms for sparsity-regularized cone-beam ct image reconstruction," *Medical Physics*, vol. 43, no. 4, pp. 1849–1872, 2016.

[218] S.-H. Yang and H. Zha, "Mixture of mutually exciting processes for viral diffusion," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013, pp. 1–9.

[219] Y. Yang, J. Etesami, N. He, and N. Kiyavash, "Nonparametric hawkes processes: Online estimation and generalization bounds," *arXiv preprint arXiv:1801.08273*, 2018.

[220] J. Yuan and S.-J. Tang, "Adaptive discount allocation in social networks," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ACM, 2017, p. 22.

[221] D. H. Zanette and S. Risau-Gusmán, "Infection spreading in a population with evolving contacts," *Journal of biological physics*, vol. 34, no. 1-2, pp. 135–148, 2008.

[222] A. Zarezade, U. Upadhyay, H. Rabiee, and M. Gomez-Rodriguez, "Redqueen: An online algorithm for smart broadcasting in social networks," in *WSDM '17: Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, 2017.

[223] A. Zarezade, A. Khodadadi, M. Farajtabar, H. R. Rabiee, and H. Zha, "Correlated cascades: Compete or cooperate," *arXiv preprint arXiv:1510.00936*, 2015.

[224] L. Zeng, K. Starbird, and E. S. Spiro, "# unconfirmed: Classifying rumor stance in crisis-related social media messages," in *Tenth International AAAI Conference on Web and Social Media*, 2016.

[225] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec, "Seismic: A self-exciting point process model for predicting tweet popularity," in *KDD*, 2015.

[226] Z. Zhao, P. Resnick, and Q. Mei, "Enquiring minds: Early detection of rumors in social media from enquiry posts," in *Proceedings of the 24th International Conference on World Wide Web*, ACM, 2015, pp. 1395–1405.

[227] D. Zhou, J. Li, and H. Zha, "A new mallows distance based metric for comparing clusterings," in *Proceedings of the 22nd international conference on Machine learning*, ACM, 2005, pp. 1028–1035.

[228] K. Zhou, H Zha, and L. Song, "Learning triggering kernels for multi-dimensional hawkes processes," in *International Conference on Machine Learning*, 2013, pp. 1301–1309.

[229] K. Zhou, H. Zha, and L. Song, "Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes," in *Artificial Intelligence and Statistics (AISTATS)*, 2013.

[230] G. Zschaler, G. A. Böhme, M. Seißinger, C. Huepe, and T. Gross, "Early fragmentation in the adaptive voter model on directed networks," *Physical Review E*, vol. 85, no. 4, p. 046 107, 2012.

# VITA

Mehrdad Farajtabar is a Ph.D. candidate in the School of Computational Science and Engineering, College of Computing, Georgia Institute of Technology. He earned a Bachelor of Science degree in software engineering from Sharif University of Technology, Iran in 2009, and a Master of Science degree from the same university in 2012. His research interests are machine learning and scalable data mining methods for the modeling, analysis, optimization, and control of processes over networks.