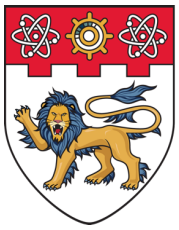


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

ADVANCED HAWKES PROCESSES FOR PRACTICAL EVENT SEQUENCE ANALYSIS: MODELS AND ACCELERATIONS

TIANBO LI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2021

Advanced Hawkes Processes for Practical Event Sequence Analysis: Models and Accelerations

Tianbo Li

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

2021

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

20 June 2021
.....
Date


.....
Tianbo Li

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

20 June 2021

.....

Date

.....

Prof. Yiping Ke

Authorship Attribution Statement

This thesis contains material from three papers published at conferences in which I am listed as an author.

Chapter 4 was published as Li, T., & Ke, Y. (2020). Tweedie-Hawkes Processes: Interpreting the Phenomena of Outbreaks. The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020. New York, USA. [1]

The contributions of the co-authors are as follows:

- I proposed the idea of incorporating Tweedie regression into Hawkes processes.
- Prof. Ke and I participated in initial discussion and designed the experiments for evaluating the proposed method.
- I derived the theoretical analysis including sub-criticality and convergence analysis.
- I collected data and conducted the experiments.
- I wrote the drafts of the manuscript.
- Prof. Ke revised the drafts and gave many valuable comments, which helped increase the quality and made the draft more readable.

Chapter 5 was published as Li, T., Wei, P., & Ke, Y. (2018). Transfer Hawkes Processes with Content Information. Proceedings - IEEE International Conference on Data Mining, ICDM, 2018, 1116–1121. Singapore. [2]

The contributions of the co-authors are as follows:

- Dr. Wei and I initialized the problem setting and the general idea of the solution.
- Dr. Wei, Prof. Ke and I participated in the technical discussion part.
- I formulated the solution and proposed two algorithms for solving the task.
- Dr. Wei and I designed the experiments.
- I collected the data and conducted the experiments.
- I wrote the drafts of the manuscript. The manuscript was revised together with Prof. Ke and Dr. Wei.
- Prof. Ke helped edit the manuscript drafts.

Chapter 6 was published as Li, T., Luo, T., Ke, Y., & Pan, S. (2021). Mitigating Performance Saturation in Neural Marked PointProcesses: Architectures and Loss Functions. Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD, 2021 [3].

The contributions of the co-authors are as follows:

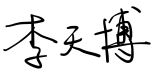
- Tianze Luo and I initialized the problem and proposed that graph convolutional network can be incorporated into Hawkes processes and accelerate the training of Hawkes process learning.
- Tianze Luo, Prof. Ke, Prof. Pan and I discussed for many rounds and comfirmed the research direction and experiment design.
- Tianze Luo and I implement the model and performance the experiments for evaluation.
- I mainly wrote the draft.
- Tianze Luo, Prof. Ke, and Prof Pan helped revised the drafts.

Chapter 7 was published as Li, T., & Ke, Y. (2019). Thinning for accelerating the learning of point processes. Advances in Neural Information Processing Systems, NeurIPS. Vancouver, Canada. [4]

The contributions of the co-authors are as follows:

- I proposed the idea of using thinning for acceleration.
- Prof. Ke and I discussed the idea for many times and provided the initial solution.
- I derived the theoretical result for parameter, gradient estimation and the thinning-based stochastic gradient decent algorithm.
- I performed the experiments and wrote the drafts of manuscript.
- Prof. Ke provided constructive suggestions and revised the drafts.

20 June 2021 ..... Date



..... Tianbo Li

Acknowledgements

It is difficult to overstate my gratitude to my Ph.D supervisor, Prof. Kelly, Yiping Ke for providing me an opportunity to work in her group and continuously supporting my study and related research. Her inspiring thought, insightful guidance, and kind encouragement help me move to the right track during these years. Without her supervision, I could not achieve what I have. Her enthusiasm and vast knowledge of machine learning plus her rigorous attitude to perform state-of-the-art research were always an inspiration for me.

I would like to thank the other members of my thesis advisory committee (TAC) meetings: Prof. Kong Wai Kin Adams and Prof. Yi Li, for their time to attend the yearly progess reporting meeting and thesis defense and their insightful suggestions and comments on my research. Additionally, I would like to thank all the co-authors of my works: Dr. Pengfei Wei, Tianze Luo and Prof. Sinno Jialin Pan, for their enthusiastic discussions and constructive suggestions, which would not be possible without their deep wisdom and sharing of many new ideas.

I want to thank my fellow colleagures for all the fun memories that I take with me from NTU. Singapore is an amazing place where east meets west, and these four years were a wonderful experience that I will never forget. Pursuing Ph.D here was fantastic, but it was also a journey full of difficulties and challenges. Without the constant support from the amazing people I met here, I don't know how I could come through those disheartening and stressful nights. I feel grateful to my dear friends: Tianze Luo, Qiuhao Zeng, Yue Deng, Shangyu Chen, Yu Chen, Dr. Xiaoming Li for the wonderful time we spend together. I would also like to thank Zuoyi Lin, Xinghua Qu, Jianda Chen and Dr. Longkai Huang for being my partners doing sports with me, which gave me the energy to face the actual challenges in research. I would also like to thank my old friends who are now living in different countries: Cheng Cheng, Xiaoyu Wu, Shimeng Li and Hong Chen for always being there supporting me in spite of my endless complaining and grumbling.

Most importantly, I am very much indebted to my family back home. I thank my parents for so many years of love and understanding. Their love is uncountable. They are truly the best parents one could ask for. I cannot express in words my gratitude to them for their sacrifices for me.

To my dear family

Abstract

Hawkes processes, first proposed in the name of self- and mutually exciting point process, have been widely used to model event sequences produced from natural and social systems. Hawkes processes can capture both individual and interactive behaviors and have achieved satisfactory results in a variety of disciplines. Ranging from recommender systems to earthquake prediction to neural activities, applications regarding Hawkes process have confirmed the effectiveness of the model as a competent tool for dealing with event sequences. In this thesis, I specifically focus on the extension of Hawkes processes to practical event sequences analysis. In particular, I make efforts to the study of Hawkes processes in two aspects: modeling and algorithm acceleration.

For the **modeling** part of event sequences, I pay special attention to the features associated with events and mainly deal with two subproblems. First, I present a novel model named *Tweedie Hawkes process* (THP), which links features associated with heave-tailed excitation. The model is essentially an instance of probabilistic graphical models and is able to learn from the outbreaks of events and find out the dominant factors behind. The model parameterizes the excitation parameter in Hawkes process with a Tweedie regression over event features. THP leverages on the Tweedie distribution in capturing various excitation effects. A variational EM algorithm is developed for model inference. Some theoretical properties of THP, including the sub-criticality and convergence of the learning algorithm, are discussed. Second, I study the problem of learning from cross-domain event sequences for Hawkes processes. One of the most important characteristics of Hawkes processes is that they link the occurrence of events up to the network structure, which makes it possible to infer the network structure from nothing but the dynamics of the event. However, cross-domain and feature information, which is also instrumental in modeling, is always neglected in existing works. I explore the idea of network transfer for Hawkes processes to leverage cross-domain information. The idea is instantiated by two models *trHLSH* and *BTHM*, from parametric and Bayesian

perspective, respectively. Both models augment Hawkes processes with features and cross-domain information. We also present effective learning algorithms for each model. Evaluation of both synthetic and real-world datasets demonstrates that the proposed models can jointly learn knowledge from the temporal, feature, and cross-domain information, and have better performance in terms of network recovery and prediction.

I also study the problem of **acceleration** for the learning process of Hawkes processes in the second part of this thesis. Traditional maximum likelihood estimation and expectation-maximization methods often suffer from high computational complexities. To speed up the process of model inference, I propose two methods: a deep learning model, called *Graph Convolutional Hawkes Processes* (GCHP), and a generic downsampling method called *thinning* for not only Hawkes processes but more general point processes. Deep learning has been attempted to the learning of event sequences in recent years. Existing methods, however, suffer from the limitations of failing to consider continuous features, relatively time-consuming training process, and restricted intensity assumptions. GCHP eschews recurrent units and is able to learn a non-linear marked Hawkes process via graph convolutional layers. The model provides a general framework for feature embedding in attributed event sequences and learns a nonlinear intensity without pre-defined form. Our model learns point processes with only graph convolutional layers and therefore it can be easily accelerated by the parallel mechanism. The model shows great prediction accuracy and efficiency in the experiment. The second method discusses one of the most fundamental issues about point processes that what is the best sampling method for point processes. Thinning as a downsampling method can be used for accelerating the learning of point processes. I find that the thinning operation preserves the structure of intensity, and is able to estimate parameters with less time and without much loss of accuracy. Theoretical results including intensity, parameter, and gradient estimation on a thinned history are presented for point processes with decouplable intensities. A stochastic optimization algorithm based on the thinned gradient is proposed. Experimental results on synthetic and real-world datasets validate the effectiveness of thinning in the tasks of parameter and gradient estimation, as well as stochastic optimization.

Keywords: Hawkes process, point process, attributed event sequences, recommender system, social network analysis, graphical model, Granger causality.

Contents

List of Figures

List of Tables

Symbols and Notations

$\mathbb{R}^n$	the n-dimensional Euclidean space
$\mathbb{N}$	the set of natural numbers
$N(t)$	a temporal point process with time index $t \in [0, \infty)$
Ω	Sample space
$\boldsymbol{\mathcal{F}}$	filtration
$\mathcal{H}_t$	σ-algebra generated by $\mathcal{N}(t)$
$\boldsymbol{\mathcal{H}}$	intrinsic filration of $\mathcal{N}(t)$
M	dimension of a multi-dimentional stochastic process
$\lambda(t)$, $\lambda_m(t)$	intensity function of uni-/multi-dimensional point process
$\lambda^*(t)$, $\lambda^*_m(t)$	conditional intensity function
$\Lambda(t)$	compensator
t_i, $t_{i,k}$	the i-th event (on the k-th dimension)
m_i	the mark associate with event t_i
$\mathbb{E}$	expectation
$\mathbb{V}$	variance
$p(\cdot)$	probability density function
$\mathbb{P}$	probability
$\odot$	the Hadamard (component-wise) product
$\otimes$	the Kronecker product
$\langle \cdot, \cdot \rangle$	the inner product of two vectors
$\circ$	the composition of functions
$*$	the convolution operator

$\|\cdot\|$	the 2-norm of a vector or matrix in Euclidean space
∇	the gradient operator

Acronyms

TPP, PP	(temporal) point process
HP	Hawkes process
MLE	maximum likelihood estimation
LSE	least-square estimation
EM	expectation maximization
VEM	variational expectation maximization
MCMC	Markov chain Monte Carlo
NN	neural network
CNN	convolutional nerual network
RNN	recurrent nerual network
GCN	graph convolutional network
GD	gradient descent
SGD	stochastic gradient descent
i.i.d.	independent and identically distributed
$a.s.$	almost sure convergence of a random sequence
p	comvergence in probability

Chapter 1

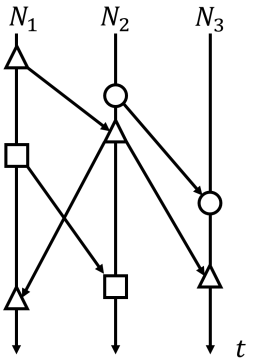
Introduction

Hawkes processes, first proposed in the name of self-/mutually excited point processes, are a powerful tool for modeling event sequences. Hawkes processes are able to model event sequences that occur asynchronously on a continuous time domain, finding the intrinsic relations from the observations among entities. In recent years, massive successes have been witnessed about this flexible and versatile tool. Some examples of the applications regarding Hawkes processes are:

- Recommendation based on user's purchase history, which can be treated as a sequence of the purchased items [8–13]. Hawkes processes are capable of finding the triggering patterns among the combinations of items and users.
- Trajectory, location, and demand prediction for taxi orders. Hawkes processes can capture both the temporal and spatial information of taxi orders [14–16].
- Social network analysis. Hawkes processes are able to find the unobserved network structure from information cascades [17–26].

Hawkes process modeling is a data-driven method for temporal data streams. The development of Hawkes processes tightly depends on the increasing demand for modeling a special type of data, called **event cascades**, or **event sequences**. The original meaning of the Italian origin word *cascade* is a steplike waterfall. In computer science, "cascade" usually refers to a successive process or operation where an event triggers another, which can be seen in many practical systems. For example, information in social networks always disseminates in such a way that

(a) Real cascades



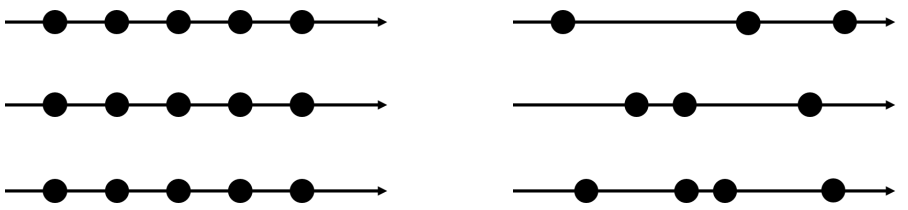
(b) Event cascades.

FIGURE 1.1: An analogy between real and event cascades.

one event may trigger a series of responses from different users. An illustration is presented in the Figure 1.1.

Some researchers refers to the data as asynchronous event sequences [24, 27, 28], which describes the feature of data that events do not occur at the same time, unlike the synchronous counterparts in traditional time series analysis. A comparison of synchronous and asynchronous event sequences is illustrated in Figure 1.2. For example, if we want to analyze the correlation among the stock prices of the 30 Dow Jones companies, we may use the opening or closing prices during a specific observation window. These data are collected at the same time each day, i.e., synchronously. However, in social networks and other computer systems, events do not take place simultaneously. Take Twitter for example. The users' timelines are asynchronous, as these events (such as post/retweets/...) in timelines have respective timestamps. Generally, asynchronous event sequences can be modeled by a multi-dimensional counting process, which is the category that Hawkes processes fall into.

Despite these multifarious applications regarding Hawkes processes, some practical



(a) Synchronous Event Sequences

(b) Asynchronous Event Sequences

FIGURE 1.2: Synchronous & Asynchronous Event Sequences

issues still hinder the usage of Hawkes processes in real-life applications. For example, many state-of-the-art models [27, 29, 30] suffer from relatively high time complexity of $\mathcal{O}(d^2n^3)$, where n is the number of events and d is the dimension, which makes such models lack scalability in realistic applications. On the other hand, as new application scenarios have been emerging, new tasks and demands brings new challenges for employing algorithms for Hawkes processes. Take the cold start problem for example. Recommending suitable items for new users is an important but still challenging problem when it comes to Hawkes processes-related applications.

On the other hand, these methods mainly deal with temporal information in event sequences. Other information that may also be helpful for us to understand the dynamics of event sequences is often neglected. For example, each event may associate with an event type, or other event features that describe what this event is about. Taking the event features into account will help us predict what will happen in the next. Therefore utilizing these different resources of information is important for event sequences analysis and that is also what I would like to contribute.

In this thesis, I aim to discuss these practical problems that are often encountered in real-world applications. I would like to improve existing Hawkes processes in two aspects: **modeling** and **acceleration**. The Meanwhile, with the development of machine learning in recent years, especially deep learning, many new ideas and techniques have been proposed to tackle traditional and new challenges in practical applications. I attempt to incorporate these recent developments to event sequence modeling and analysis and study several advanced topics regarding Hawkes processes. The **research problems** studied in this thesis include:

- **Model 1: Learning non-Gaussian excitation from event features for Hawkes processes.** Despite its success in many applications, traditional Hawkes processes and most Hawkes-related models are not competent in capturing outbreaks. They mainly suffer from one or many of the following drawbacks — *invariant excitation*, *neglecting content/features*, *weak interpretability*, *unrealistic distribution of aggregation*, and *failure to sub-criticality*. Existing models often assume that the excitation of an event is a constant or has a Gaussian-like distribution, which may not be realistic when it comes to practical applications.

- **Model 2: Network Transfer for Hawkes Processes: Learning from Cross-domain Temporal and Feature Information.** Vanilla Hawkes processes and most related models only deal with temporal information of event sequences. Besides temporal information, cross-domain knowledge can also be beneficial. Recently, transfer learning [31], which has been a topic of active interest, sheds light on how to exploit the knowledge from other domains and help improve the performance of models. However, to the best of our knowledge, none of the existing works has explored transfer learning for Hawkes processes.

- **Acceleration 1: Learning representation embedding for event features via graph convolution neural network for Hawkes processes.** Recently, some researchers [32–35] have explored the idea of incorporating marked Hawkes processes with recurrent networks (RNNs). These models, however, only focus on the marked Hawkes processes with categorical features, which are equivalent to multi-dimensional Hawkes processes, but fail to consider continuous features. The recurrent architecture of these models makes it difficult to be accelerated by parallel mechanisms as convolutional networks (CNNs) do [36, 37], therefore these models are usually more time-consuming.

- **Acceleration 2: Downsampling method for improving the scalability of Hawkes processes.** Despite the popularity of Hawkes processes and other point processes, related applications are often plagued by the scalability issue. Some state-of-the-art models [27, 29, 30] suffer from the drawback that, as the number of events increases, learning such a model would be very time-consuming, if not infeasible. This becomes a major obstacle when applying point processes.

To tackle these questions, I attempt to utilize advanced machine learning techniques and incorporate them into Hawkes processes. Specifically, as shown in Figure 1.3, the thesis involves concepts and theories from transfer learning, probabilistic graphical model, deep learning, stochastic analysis and other recent developments. For example, to enhance the representative ability of Hawkes processes for modeling events that happen less frequently, I try to adopt the methodology from probabilistic graphical model theory, and impose a Tweedie distribution on the excitation parameters. A variational expectation maximization method is proposed for parameter inference. I also make an effort to include some recent developments in

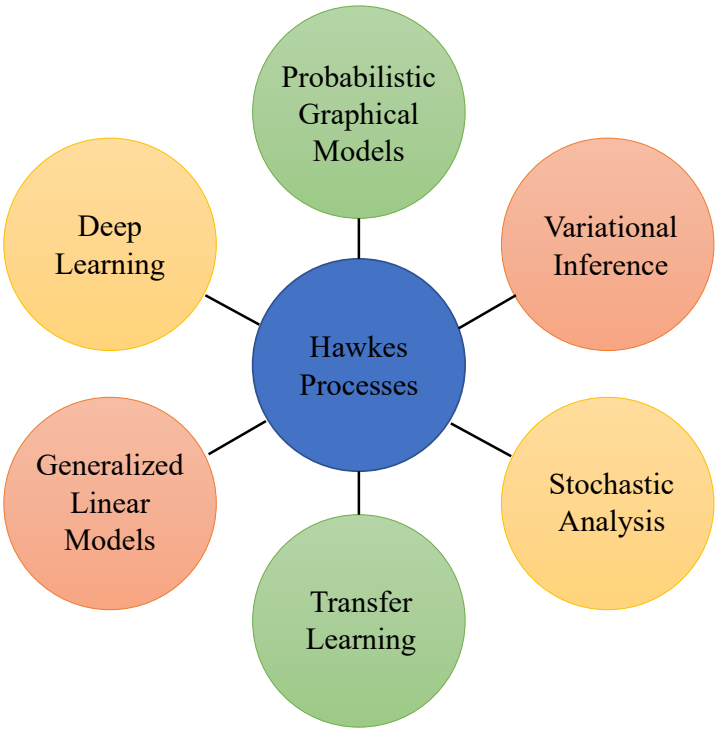


FIGURE 1.3: An illustration of the tools and methods.

deep learning. As the graph structure has been embedded in the multi-dimensional Hawkes processes, graph neural networks has the potential to improve the performance and representative ability of multi-dimensional Hawkes processes. In the following chapters, we will see an attempt to incorporate Hawkes processes with graph convolutional networks, which shows superiority in terms of scalability and prediction accuracy.

The remaining part of this thesis is organized as follows. Chapters 2 and 3 introduce the mathematical background and recent developments of the studies on Hawkes processes and event sequence modeling. Chapters 4 and 5 present two models to enhance Hawkes processes by learning from not only the temporal information but also the attributes and cross-domain information. Chapters 6 and 7 introduce two methods for the acceleration of the learning of Hawkes processes. Here are the brief summaries of each part:

- **Part I (Modeling).** In this part, I focus on enhancing the modeling ability

of Hawkes process. I utilize the techniques from the recent developments of transfer learning, probabilistic graphical models, and generalized linear model to augment the flexibility and efficacy of Hawkes processes when dealing with practical tasks. Specifically, two tasks are discussed: first, I introduce a Bayesian model that incorporate Tweedie regression to Hawkes processes, which is able to model the dominant factor of an outbreak of events; second, I study the problem of network transfer for Hawkes processes. Special focus is put on the incorporation of the features associated with events.

- **Part II (Acceleration).** In this part, I introduce two methods for the acceleration of Hawkes processes inference. The first method utilize the graph convolutional network, which eschews the recurrent units of the RNN-based Hawkes processes. The second method proposes thinning as a downsampling method for Hawkes processes. This method is not only valid for the vanilla Hawkes process, but also valid for a broader categories of point processes.

Chapter 2

Mathematical Background

In this chapter, I am focusing on introducing the mathematical background of Hawkes processes, including definition, properties, and model inference.

2.1 Hawkes Processes

First, I introduce some basic concepts regarding Hawkes processes: the conditional intensity function, multi-dimensional Hawkes process, and the likelihood.

Conditional Intensity Function. Hawkes processes, as a class of self- and mutually excited point processes, are characterized by its conditional intensity function, which depicts the expected occur rate of event given the history up to some time. It measures the probability that an event will occur at or around time t. For a unidimensional point process $N(t)$, the conditional intensity function is defined by,

$$\lambda^*(t) := \lambda(t|\mathcal{H}_{t\_}) = \lim_{h\to 0} \frac{\mathbb{E}\left[N(t+h) - N(t)|\mathcal{H}_{t\_}\right]}{h}, \tag{2.1}$$

where $\mathcal{H}_{t\_}$ is the σ-algebra of events occurring at times up to but not including t. Eq. 2.1 can also be written as $\lambda(t)dt = \mathbb{E}dN(t)$. Another way to define the conditional intensity function is to express it in terms of survival function:

$$\lambda^*(t) = \frac{p(t|\mathcal{H}_{t\_})}{S(t|\mathcal{H}_{t\_})}, \tag{2.2}$$

where $p(t|\mathcal{H}_{t_-})$ is the conditional density function given the history, and $S(t|\mathcal{H}_{t_-})$ the conditional survival function.

Multi-dimensional Hawkes Process. A classic M-dimensional Hawkes Process $N(t)$ is defined with intensities $\lambda_m(t)$, $m = 1, \ldots, M$ given by:

$$\lambda_m(t) = \mu_m + \sum_{k=1}^{M} \int_0^t \phi_{mk}(t-u) dN_k(u), \tag{2.3}$$

where μ_m is the **exogenous intensity**, also known as the base intensity or background rate. $\phi_{mk}(t)$ is the decay kernel function that measures the evolution of the excitation from node k to m. In principle, the conditional intensity function is always assumed to be non-negative, thus I stipulate $\phi_{mk}(t) \geqslant 0$ and $\phi_{mk}(t) = 0$ when $t > 0$. The remaining part of the intensity function is usually referred to as the **endogenous intensity**, which measures the total excitation effect from the other events that have arrived before the given time t. Writing the parameters above in the matrix form, Eq. (2.3) can be expressed in the dense form:

$$\boldsymbol{\lambda}(t) = \boldsymbol{\mu} + \int_0^t \boldsymbol{\Phi}(t-u) d\boldsymbol{N}(u). \tag{2.4}$$

where $\boldsymbol{\mu} = \{\mu_i\}_{i=1,\ldots,M}$, and $\boldsymbol{\Phi}(t) = \{\phi_{mk}(t)\}_{m,k=1,\ldots,M}$. Eq. (2.4) can also be recast by defining convolution operation $*$ as,

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

Given an M-dimensional asynchronous event sequence $\{t_{i,m}\}_{m=1,\ldots,M;i=1,\ldots,n_m}$, the empirical intensity function $\lambda_m(t)$ can be written as:

$$\lambda_m^*(t) = \mu_m + \sum_{k=1}^{M} \sum_{t>t_{i,k}} \phi_{mk}(t - t_{i,k}). \tag{2.5}$$

Stationarity. The property of stationarity originates from the assumption that the distribution of the stochastic patterns of the process do not change over time. Stationarity is a fundamental assumption for statistical prediction and simulation. If a process $N(t)$ has asymptotically stationary increments and $\lambda(t)$ is asymptotically

stationary, the expectation of the conditional intensity function can be written as:

$$\mathbb{E}\boldsymbol{\lambda}(t) = (\boldsymbol{I} - \boldsymbol{\Gamma})^{-1}\boldsymbol{\mu},$$

where $\boldsymbol{\Gamma} = \int_0^\infty \boldsymbol{\Phi}(u)du$. Here $\boldsymbol{I} - \boldsymbol{\Gamma}$ should be invertible and $(\boldsymbol{I} - \boldsymbol{\Gamma})^{-1}$ nonnegative. Therefore a sufficient condition for a multivariate Hawkes process would be:

$$\rho(\boldsymbol{\Gamma}) < 1, \tag{2.6}$$

where $\rho(\boldsymbol{\Gamma})$ is the spectral radius of matrix $\boldsymbol{\Gamma}$.

Decay Kernels. Commonly used decay kernels are listed in Table 2.1.

TABLE 2.1: Commonly-used decay kernels

Kernals	**Functions**
Exponential Kernel	$\phi_{jk}(t) = \alpha_{jk}\beta_{jk}e^{-\beta_{jk}t}$
Linear Kernels	$\phi_{jk}(t) = \alpha_{jk}\max\{-\beta_{jk}t, 0\}$
Inverse Kernels	$\phi_{jk}(t) = \dfrac{\alpha_{jk}}{\beta_{jk}t + \gamma}$
Gaussian Kernels	$\phi_{jk}(t) = \alpha_{jk}\exp\left(-\dfrac{t^2}{2\beta_{jk}^2}\right)$
Power-law Kernels	$\phi_{jk}(t) = \dfrac{\alpha_{jk}\beta_{jk}}{(1+\beta_{jk}t)^{1+\gamma}}$
Rayleigh Kernels	$\phi_{jk}(t) = \alpha_{jk}\dfrac{t}{\beta_{jk}^2}e^{-\frac{t^2}{2\beta_{jk}^2}}$

All of these kernels contain two types of parameters: the **infectivity parameter** α_{jk}, which controls the magnitude of the excitation, and the **decay parameter** β_{jk}, which controls the change of the excitation over time. Usually these parameters can be assembled as two $M \times M$ matrices $\boldsymbol{A} = \{\alpha_{jk}\}_{j,k=1,\dots,M}$ and $\boldsymbol{B} = \{\beta_{jk}\}_{j,k=1,\dots,M}$, called **infectivity** and **decay matrix**, respectively. Among the kernels listed above, the exponential kernel enjoys many desirable properties, making it the most popular one for practical applications.

As the modeling of Hawkes processes is unavoidably implemented via its conditional intensity function, no matter what kernel function is employed, the behavior of the process is generally controlled by three bunches of parameters: the exogenous intensity $\boldsymbol{\mu}$, the infectivity matrix $\boldsymbol{A}$ and decay parameter $\boldsymbol{B}$. By manipulating these parameters, Hawkes processes are capable of dealing with a wide variety of tasks. Table 2.2 summarizes some typical applications using Hawkes process regarding particular parameters.

TABLE 2.2: Parameters in Hawkes and Usage.

Parameter	Meaning
$\boldsymbol{\mu}$	The exogenous intensity is usually used to model the events triggered externally. These events are called *immigrant events*, which take place spontaneously, rather than infected by others. If generalized to a function of time, it can be used to model seasonality.
$\boldsymbol{A}$	$\boldsymbol{A}$ is highly relevant to the network structure. It measures the strength of mutual influence. $\boldsymbol{A}$ plays key role in network inference, clustering, network dynamics, recommendation as well as other network-related applications.
$\boldsymbol{B}$	$\boldsymbol{B}$ governs how the intensity changes over time. Some applications with respect to these parameters include popularity prediction, information intervention, etc.

Likelihood. The log-likelihood function of a unidimensional non-homogeneous point process in terms of the intensity function is given by,

$$\ell = \sum_i \log \lambda(t_i) - \int_0^T \lambda(t) dt.$$

For an m-dimensional Hawkes process, the log-likelihood function is expressible as:

$$\ell = \sum_{m=1}^{M} \left\{ \sum_{t_l} \log \left[\mu_m + \sum_{k=1}^{M} \sum_{t_l > t_{i,k}} \phi_{mk}(t_l - t_{i,k}) \right] - \mu_m T - \sum_{k=1}^{M} \sum_{t > t_{i,k}} \int_0^T \phi_{mk}(t - t_{i,k}) dt \right\}. \tag{2.7}$$

Marked Hawkes processes. Marked Hawkes processes (MHP) [38] are commonly used for modeling the temporal dynamics of attributed event sequences. A marked Hawkes process is a point process $N(\cdot,\cdot)$ on $\mathcal{T}\times\mathcal{M}$, where $\mathcal{T}=[0,T]$ is the observation window and $\mathcal{M}$ the mark (feature) space. It is worth noting that if $\mathcal{M}$ is finite discrete, N is degenerated to a multi-dimensional Hawkes processes. In this chapter, we assume that $\mathcal{M}$ can be continuous, i.e., $\mathcal{M}=\mathbb{R}^p$. The continuous assumption is more general and common in real world. *Spatio-temporal Hawkes processes* [39] are a good example of continuous mark space, as the location of a point (latitude and longitude) is in $\mathbb{R}^2$.

Given the *nature history* $\mathcal{H}_{t-}$, which is defined by the σ-algebra: $\mathcal{H}_{t-}=\sigma\{N(s,\mathcal{M};\omega): 0<s<t\}$, where ω is a sampled path, the *conditional intensity function* of a marked point process is defined by

$$\lambda(t,m|\mathcal{H}_{t-})=\lim_{\Delta_t,\Delta_m\to 0}\frac{\mathbb{E}\left[N\left([t,t+\Delta_t)\times B(m,\Delta_m)\right)|\mathcal{H}_{t-}\right]}{\Delta_t|B(m,\Delta_m)|},$$

where $|B(m,\Delta_m)|$ is the Lebesgue measure of the ball $B(m,\Delta_m)$ with radius Δ_m. It can be decomposed by [38]

$$\lambda(t,m|\mathcal{H}_{t-})=\lambda_g(t|\mathcal{H}_{t-})p(m|t,\mathcal{H}_{t-}),$$

where $\lambda_g(t)$, which is the marginal intensity w.r.t. time, is referred to as the *ground intensity*. A marked point processes is said to be a marked Hawkes process if it admits the form of a temporal Hawkes process. $p(m|t,\mathcal{H}_{t-})$ is the conditional mark density which refers to the distribution to be anticipated at the end of a time interval, not immediately after the next interval has begun.

Given a realization of attributed event sequence $\{(t_i,m_i): i=1,\dots,N\}$, the log-likelihood function is given by [38]

$$\ell=\sum_{i=1}^{N}\left[\log\lambda_g(t_i|\mathcal{H}_{t_i-})-\int_{t_{i-1}}^{t_i}\lambda_g(t|\mathcal{H}_{t_i-})dt+\log p(m_i|t_i,\mathcal{H}_{t_i-})\right].$$

It is worth noting that the definition of multi-dimensional Hawkes processes (MHP) complies with that of marked Hawkes processes, in that dimensions of an MHP can be regarded as multi-categorical marks, who have a multinomial distribution conditioned upon the history.

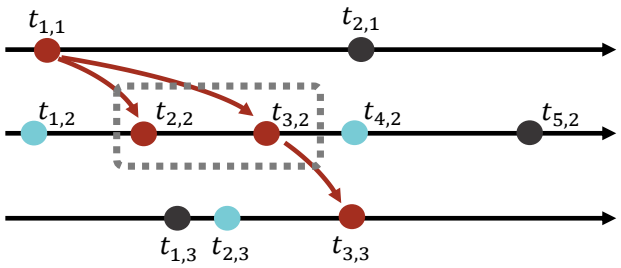


FIGURE 2.1: An illustration of branching structure. $t_{2,2}$ and $t_{3,2}$ as in the dashed rectangle are the offsprings of $t_{1,1}$, therefore belong to the subprocess $\mathcal{P}_{21,1}$. $t_{3,3}$ is triggered by $t_{2,3}$

2.2 Properties

2.2.1 Branching Structure

Branching structure is first proposed in the name of *cluster process representation* [40]. Consider the conditional intensity function presented in Eq. (2.5). The intensity can be decomposed into a bunch of conditionally independent subprocesses:

- homogeneous Poisson processes $\mathcal{P}_{i0}$ with constant intensity μ_m. These processes explain the arrival of *immigrant events*.
- $\sum_{m=1}^{M} N_m(t)$ nonhomogeneous Poission processes $\mathcal{P}_{ij,k}$ with intensity $\lambda_{ij,k} = \alpha^{ij}\phi(t - t_{k,j})$. The subscripts and superscripts represent the subprocess is triggered subsequently by event $t_{k,j}$ over dimension i, therefore called cluster center. That is, the events in this subprocess are *offspring events* of the ancestor event $t_{k,j}$ in dimension i.

From this point of view, a Hawkes process can be regarded as superposition of these processes $\{\mathcal{P}_{i0}\}_{i=0,\dots,M}$ and $\{\mathcal{P}_{ij,k}\}_{k=1,\dots,n^j;i=1,\dots,M;j=1,\dots,M}$. A three-dimensional Hawkes process branching structure is shown in Figure 2.1. Event $t_{1,1}$ is an immigrant event, since it is not triggered by any preceding event. Events $t_{2,2}$ and $t_{3,2}$ are the offspring events of $t_{1,1}$, generated by the subprocess $\mathcal{P}_{21,1}(\lambda_{21,1})$, where $\lambda_{21,1}(t) = \alpha_{21}\phi(t - t_{1,1})$. Based on the branching structure, we introduce a hidden variable Ψ of each individual event, representing which subprocess it belongs to. Simply speaking, Ψ denotes which previous event triggers it. Ψ's take values from $\psi_0 \cup \{t_k^i\}$, where ψ_0 means the event is an immigrant which has no ancestor, and

$\Psi_{k',j} = t_{k,i}$ suggests that event $t_{k,i}$ triggers $t_{k',j}$. The branching will be used in the next two models: Tweedie Hawkes process and Bayesian Transfer Hawkes Model.

2.2.2 Some Limit Theorems

Here I talk about the two fundamental limit theorems for Hawkes processes, which help us understand the asymptotic behavior of multi-dimensional Hawkes processes at large time scales. Proof can be found in [41].

Law of Large Numbers Assume stationarity (Eq. (2.6)) holds for a multi-dimensional Hawkes process $\boldsymbol{N}(t)$. Then,

$$\lim_{t\to\infty} \sup_{v\in[0,1]} \left\| \frac{\boldsymbol{N}(tv)}{t} - v(\boldsymbol{I}-\boldsymbol{\Gamma})^{-1}\boldsymbol{\mu} \right\|_2 = 0 \tag{2.8}$$

almost surely and in $L^2(p)$, where $\boldsymbol{\Gamma} = \int_0^\infty \boldsymbol{\Phi}(u)du$. This result is valid for any stationary kernel functions $\boldsymbol{\Phi}(t)$. This result assures that if the observation time is long enough, Hawkes processes *almost surely* converge to a certain value, which means if the number of sample events is adequate, the probability that model prediction goes far from the ground truth is low.

Central Limit Theorem Assume stationarity (Eq. 2.6) holds for a multi-dimensional Hawkes process $\boldsymbol{N}(t)$, as well as the restriction on $\boldsymbol{\Phi}(t)$:

$$\int_0^\infty t^{1/2}\boldsymbol{\Phi}(t)dt < \infty,$$

componentwise. Then, as $t \longrightarrow \infty$, in law for the Skorokhod Topology,

$$\sqrt{t}\left(\frac{\boldsymbol{N}(tv)}{t} - v(\boldsymbol{I}-\boldsymbol{\Gamma})^{-1}\boldsymbol{\mu}\right) \longrightarrow (\boldsymbol{I}-\boldsymbol{\Gamma})^{-1}\boldsymbol{\Sigma}^{1/2}\boldsymbol{B}(v), \tag{2.9}$$

where $v \in [0,1]$ and $\boldsymbol{B}(v)$ is a standard multi-dimensional Brownian motion and $\boldsymbol{\Sigma}$ is the diagonal matrix such that $\boldsymbol{\Sigma}_{ii} = ((\boldsymbol{I}-\boldsymbol{\Gamma})^{-1}\boldsymbol{\mu})_i$. This central limit theorem shows that Hawkes processes behave like Brownian motion asymptotically.

2.2.3 Stochastic Differential Equation

Consider a unidimensional Hawkes process with exponential kernel $\phi(t) = \alpha e^{-\beta t}$. Applying Ito's Lemma, its intensity function satisfies the following SDE,

$$d\lambda(t) = -\left(\lambda(t) - \mu\right) dt + \alpha dN(t). \tag{2.10}$$

The proof can be found in [42]. Note that here $\lambda(t)$ is no more conditional intensity function but an unconditional one. In Stochastic Calculus, $-\left(\lambda(t) - \mu\right)$ is referred to as the drift coefficient, which depicts the trend of how the intensity function varies regarding time, while α is the diffusion coefficient. The SDE shows that the increment of intensity function is factored by the time interval dt with strength of the endogenous intensity, and the arrival of the events $dN(t)$. Since $dN(t)$ only depends on the past of $\lambda(t)$ by means of the current value of $\lambda(t)$, the exponential decay kernal preserves the Markov property of the intensity function $\lambda(t)$.

2.3 Parameter Inference

I show several methods for parameter inference. All methods assume exponential kernels.

2.3.1 Least Square Estimator

The least square estimator for nonhomogeneous Poisson process is a special case of a wide class of estimators, namely Martingale-estimators (M-estimator) as explained in [43]. The objective is to minimize the product of conditional intensity function and its deviation from the actual sample (a realization) of the process $N_m(t)$:

$$\min_{\theta} \sum_{i}^{M} \int_{0}^{T} H_m(t) \left(\lambda_m(t)dt - dN_m(t)\right), \tag{2.11}$$

where

$$H_m(t) = \frac{\partial \lambda_m(t|\boldsymbol{\theta}_m)}{\partial \boldsymbol{\theta}_m}$$

is the first derivative of the intensity function with respect to the parameter $\boldsymbol{\theta}_i$. Note that maximum likelihood estimator also belongs to the class of M-estimators.

However in the case of Hawkes process, maximum likelihood estimator is more computationally difficult than least square estimator, as it cannot yield a closed-form solution as least square estimator does. Least square estimator can also give satisfactory final estimates from the same asymptotic distribution as maximum likelihood estimators [43] do. Therefore, we adopt least square estimator as the starting point of our model trHLSH, which is going to be discussed later.

The intensity function $\lambda_i(t)$ as shown in Eq. (2.5) can be recasted as the product of:

- $\boldsymbol{\theta}_i = (\mu_i, \alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iM})'$, the parameter vector to be inferred, and
- $\boldsymbol{x}_i(t) = (1, \sum_k \psi(t - t_{k,1}|\beta_{i1}), \ldots, \sum_{k'} \psi(t - t_{k',M}|\beta_{iM}))'$, the kernel vector.

Applying the notations, Eq. (2.11) is equivalent to:

$$\min_{\boldsymbol{\theta}_i} \int_0^T \boldsymbol{\theta}_i' \boldsymbol{x}_i(t) \boldsymbol{x}_i(t)' \boldsymbol{\theta}_i dt - 2 \int_0^T \boldsymbol{\theta}_i' \boldsymbol{x}_i(t) dN_i(t).$$

Let

$$\boldsymbol{Z}_i = \int_0^T \boldsymbol{x}_i(t) \boldsymbol{x}_i(t)' dt, \tag{2.12}$$

and

$$\boldsymbol{y}_i = \int_0^T \boldsymbol{x}_i(t) dN_i(t) = \sum_{t_{k,i}} \boldsymbol{x}_i(t_{k,i}). \tag{2.13}$$

The objective function becomes,

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \boldsymbol{\theta}' \boldsymbol{Z} \boldsymbol{\theta} - 2\boldsymbol{\theta}' \boldsymbol{y} \\ \text{s.t.} \quad & \boldsymbol{\theta} \geqslant \boldsymbol{0} \end{aligned} \tag{2.14}$$

Here the subscript denoting the dimension i is omitted. This optimization problem can be directly solved by quadratic programming.

2.3.2 Maximum Likelihood Estimation

Maximum likelihood estimation (hereinafter MLE) is a common probabilistic method for parameter inference. MLE enjoys many merits, such as easy to implement and

in most scenarios the estimation is unbiased. MLE start from the log-likelihood function, which is shown in Equation 2.7. Assuming exponential kernels,

$$\begin{aligned}
&\ell(\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{B} | \{t_{im}\}_{m=1,\dots,M;i=1,\dots,n_m}) \\
&= \sum_{m=1}^{M} \sum_{j=1}^{n_m} \log \lambda_m(t_{jm}) - \int_0^T \lambda_m(t) dt \\
&= \sum_{m=1}^{M} \sum_{j=1}^{n_m} \ln \left(\mu_m + \sum_{k=1}^{M} \sum_{t_{ik}<t_{jm}} \alpha_{mk} e^{-\beta_{mk}(t_{jm}-t_{ik})} \right) \\
&- \mu_m T - \sum_{m=1}^{M} \sum_{k=1}^{M} \sum_{t_{ik}<T} \frac{\alpha_{mk}}{\beta_{mk}} (1 - e^{-\beta_{mk}(T-t_{ik})}).
\end{aligned}$$

From this, I can obtain first order derivatives:

$$\frac{\partial l}{\partial \mu_m} = \sum_{j=1}^{n_m} \frac{1}{\mu_m + \sum_{k=1}^{M} \sum_{t_{ik}<t_{jm}} \alpha_{mk} e^{-\beta_{mk}(t_{jm}-t_{ik})}} - T \tag{2.15}$$

$$\begin{aligned}
\frac{\partial l}{\partial \alpha_{mn}} = &\sum_{j=1}^{n_m} \frac{\sum_{t_{ik}<t_{jm}} e^{-\beta_{mn}(t_{jm}-t_{ik})}}{\mu_m + \sum_{k=1}^{M} \sum_{t_{ik}<t_{jm}} \alpha_{mk} e^{-\beta_{mk}(t_{jm}-t_{ik})}} \\
&- \frac{1}{\beta_{mn}} \left(n_n - \sum_{t_{in}<T} e^{-\beta_{mn}(T-t_{in})} \right)
\end{aligned} \tag{2.16}$$

$$\begin{aligned}
\frac{\partial l}{\partial \beta_{mn}} = &- \sum_{j=1}^{n_m} \frac{\sum_{t_{in}<t_{jm}} (t_{jm} - t_{in}) \alpha_{mn} e^{-\beta_{mn}(t_{jm}-t_{in})}}{\mu_m + \sum_{k=1}^{M} \sum_{t_{ik}<t_{jm}} \alpha_{mk} e^{-\beta_{mk}(t_{jm}-t_{ik})}} \\
&+ \frac{\alpha_{mn}}{(\beta_{mn})^2} \left(n_n - \sum_{t_{in}<T} \left(1 + \beta_{mn}(T - t_{in})\right) e^{-\beta_{mn}(T-t_{in})} \right)
\end{aligned} \tag{2.17}$$

A gradient descent algorithm is shown in Algorithm 1.

Algorithm 1: Gradient Descent Algorithm for MLE with Hawkes Process

Input : maximum observation time T_{max}, initial parameter matrices $\boldsymbol{\mu}^0, \boldsymbol{A}^0, \boldsymbol{B}^0$, likelihood difference ϵ, descent stepsize *step*, maximum iteration number *iter*

Output : optimal $\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{B}$

Initialize $\alpha_{mn} = \alpha^0_{mn}$, $\beta_{mn} = \beta^0_{mn}$, $\mu_m = \mu^0_m$, $i = 0$, $l^0 = l(\boldsymbol{\mu}^0, \boldsymbol{A}^0, \boldsymbol{B}^0)$;

do

$i+ = 1$;

$\mu_m+ = step \times \dfrac{\partial l}{\partial \mu_m}$, using Equation 2.15; $\alpha_{mn}+ = step \times \dfrac{\partial l}{\partial \alpha_{mn}}$, using Equation 2.16;

$\beta_{mn}+ = step \times \dfrac{\partial l}{\partial \beta_{mn}}$, using Equation 2.17;

while $|l_i - l_{i-1}| > \epsilon$ *OR* $i < iter$;

2.3.3 EM Algorithm

Another algorithm that can be used for parameter inference is the EM algorithm. In practice, EM usually run faster than gradient descent MLE algorithm.

Algorithm 2: EM Algorithm for Hawkes Processes

Input : time sequences $\{t_{im}\}_{m=1,\dots,M;i=1,\dots,n_m}$, initial parameter matrices $\theta_0(\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{B})$, likelihood difference ϵ

Output : optimal $\theta(\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{B})$

Initialize $n = 0$, set $\theta^{(n)} = \theta_0$;

compute $Pr\{Z_{ik} = z | X_i, \theta_i^{(n)}\}$ using Equation 2.19;

do

$n+ = 1$;

compute $u_i^{(n+1)}$ using Equation 2.20;

using $\theta_\beta^{(n)}$ as initial values, find $\theta_\beta^{(n+1)} = \arg\max_{\theta_\beta} Q(\theta_\beta | \theta^{(n)})$ subject to Equation 2.21;

compute $\alpha_{ij}^{(n+1)}$ using $\theta_\beta^{(n+1)}$ and Equation 2.21;

while $l(\theta^{(n+1)} | \{t_{im}\} - l(\theta^{(n)} | \{t_{im}\}) > \epsilon$;

Return $\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{B}$;

To develop an EM algorithm, first I introduce some hidden variables $\{Z_{imk}\}$ and $\{Z_{0m}\}$ indicating each event is triggered by which subprocess, as I illustrate in

Section 2.2.1. Let $\theta_\beta = \{\beta_{ij}\}_{i,j=1,\dots,M}$, the Q function is defined by:

$$Q(\theta_\beta|\theta^{(n)}) = \sum_{i=1}^{M}\sum_{z\in z_i}\left(\sum_{k=1}^{n_i}\log\lambda_z(t_{ik})Pr\{Z_{ik}=z|\{t_{im}\},\theta_i^{(n)}\} - \int_0^T \lambda_z(t)dt\right) \tag{2.18}$$

$Pr\{Z_{ik} = z|X_i, \theta_i^{(n)}\}$ is the probability that event t_{ik} belongs to subprocess Z_{ik}. The posterior probabilities:

$$Pr\{Z_{ik}=z|X_i,\theta_i^{(n)}\} = \frac{\lambda_z^{(n)}(t_{ik})}{\sum_{m\in z_i}\lambda_m^{(n)}(t_{ik})} \tag{2.19}$$

There is a closed form solution for the base rate parameters:

$$u_i^{(n+1)} = \frac{\left(\hat{N}_i^0\right)^{(n)}}{T} \tag{2.20}$$

where $\hat{N}_i^0$ is the expected number of events that happens in i-th dimension.

Then, the updated intensity parameters $\alpha_{ij}^{(n+1)}$ can be written in terms of the $\beta_{ij}^{(n+1)}$ and $\kappa_{ij}^{(n+1)}$.

$$\alpha_{ij}^{(n+1)} = \frac{\beta_{ij}^{(n+1)}\hat{N}_{ij}^{(n)}}{N_j - \sum_{k=1}^{N_j} e^{-\beta_{ij}^{(n+1)}(T-t_{jk})}} \tag{2.21}$$

where $\hat{N}_{ij}^{(n)}$ is the estimated number of events that happens in the i-th dimension triggered by a j-th dimensional event, aka, propagating from j to i. N_j is the total number of events in the j-th dimension. A complete algorithm is presented as Algorithm 2.

Chapter 3

A Literature Review

In this chapter, I introduce some recent developments in the study of Hawkes processes. The literature and related works are reviewed from two perspectives: modelings and applications.

3.1 Modelings

In Hawkes process modeling, the basic Hawkes process may not necessarily meet all the assumptions in various scenarios. Modifications have to be made in order to adapt the basic model for new purposes. For instance, it is common that the features associated with events are obtainable. A reasonable guess is that the performance of Hawkes process models can be more accurate and robust, if the features are considered in the model.

I summarize three ways to enhance the modeling ability of Hawkes processes as follows:

- **Featurization** [17, 27, 44–48]. Instead of using a single parameter, featurization expresses the parameter in terms of a linear or nonlinear combination of its features. For example, if we want to take the features of each node into consideration, a possible solution is to parameterize the original parameters in the model as a function of these features. Featurization can also be used for parameter reduction [17].

- **Functionization** [49–55]. This method is to replace a parameter with a deterministic nonlinear function $f(\cdot)$ to capture some peculiar characteristics of the parameter. The functions that are used for substituting can be either parametric or non-parametric. For example, twitter users may tend to generate more events during their leisure time than in the early morning when they are still asleep. To involve this kind of intra-day seasonality for the exogenous intensity, one can model it as a cyclical function. Another example is to model the refractoriness effect of exogenous intensity, which is the opposite of excitation. It means when an exogenous event happens, the exogenous intensity will decrease dramatically for a short period.

- **Randomization** [50, 56–59]. Randomization means to introduce extra non-determinacy into the conditional intensity function by substituting certain parameters for a random variable or a stochastic process. Actually this is in the case of Cox process, which is a non-homogeneous Poisson process where the time-dependent intensity itself is a stochastic process.

Specifically, I list some sub-areas of the study of Hawkes processes and related point processes.

Learning of parametric Hawkes processes. Parametric point processes are the most conventional and popular method in the study of point processes. For example, [29] designs an algorithm ADM4 for learning the parameter representing the hidden network of social influences. [45] parameterizes the infectivity parameter in Hawkes processes and employs the technique of ADMM for parameter estimation. [27] proposes a learning algorithm combining MLE with a sparse-group-lasso regularizer to learn the so-called "Granger causality graph". All these models are decouplable, therefore thinning is applicable to the learning of them.

Learning of non-parametric Hawkes processes. There has been an increasing amount of studies on non-parametric point processes and their learning algorithms in recent years. Isotonic Hawkes process [55] is an interesting and representative work among them, which combines isotonic regression and Hawkes processes. [60] proposes a algorithm to learn the infectivity matrix without any parametric modeling and estimation of the kernels. Another category of non-parametric models related to point processes is Bayesian non-parametric models, such as [18], [23] and [61].

Besides, some explorations of combining point processes and deep neural networks are emerging. Some typical works include [32], [34], and [62].

Acceleration for the learning of Hawkes processes. [63] proposes a method of low rank approximation of the kernel matrix for large-scale datasets. The online learning algorithm for Hawkes [64] discretizes the time axis into small intervals for learning the triggering kernels. [65] designs a hardware acceleration method for MLE of Hawkes processes. A recent work [66] introduces a stochastic optimization method for Hawkes processes. Unfortunately, none of existing works considers thinning as a sampling method to reduce the time complexity.

Hawkes processes with marks. Basic Hawkes process only considers temporal information. Recent works explore to involve the textual information in two fashions: parametric and Bayesian. The first category includes the parametric Hawkes process (PHP) [45] model, which parameterizes $\boldsymbol{\alpha}$ with a linear regression on event features. However, the distribution of α in the model is a symmetrical Gaussian and sub-criticality is not necessarily satisfied, as illustrated before. [17] proposes a model whose excitations are individuals' participation in communities. The model is similar to PHP, with an application to a clustering task. The other category is more related to our model, where the generation process of features is involved in the model. Typical models also belong to *marked point processes*, such as [67] and [68], or mixture models combined Hawkes process and the generation process of features, such as [21], [25] and [16]. These models always assume that all the events share the same distribution of α (or even same α), which is somehow unrealistic.

Bayesian learning for Hawkes processes. As Hawkes processes are a versatile probabilistic model, many recent works apply them to non-parametric Bayesian framework to ease the pain of parameters selection. For example, both DHP [23] and DMHP [24] combine Dirichlet process and Hawkes process and apply respective models to clustering tasks. However, they fail to consider the content of the event and assume that the distribution of α only depends on the cluster or nodes (invariant excitation). [25] takes into account the contents and combines Hawkes processes with topic model, but it is a shame that α is treated as a fixed parameter which is only associated to nodes. [18] propose a Bayesian non-parametric model combining Hawkes processes and the infinite relational model. The model claims to discover the implicit social structure by decompose the base intensity term into the products

of several factors, but fails to consider the transmission/diffusion of the events between groups, which also follows the basic setting of Hawkes.

Deep learning for Hawkes processes. An active research line is to learn Hawkes processes with neural networks. The RMTPP [32] model views the intensity function as a nonlinear function of the history, and uses a recurrent neural network to learn a representation of influences from the event history. Experimental results show that the model has better performance in both model fitting and prediction than traditional methods. [34] proposes a neural Hawkes process model named NHPP, which considers the interactions between events. The IRNN model [33] uses an intensity recurrent architecture that synergistically models time series and event sequence, making it able to capture both background and history effect. All of the above methods define respective intensity functions to be a specific parametric form. The fully neural point process model (FulNN) [35] relaxes the assumption of a parametric intensity function, and uses a fully connected neural network to output the cumulative hazard functions, which avoids defining a specific form of the intensity function. However, the model fails to consider the features associated with each event. The geometric Hawkes process (GeoHP) model [69] treats the parameter estimation of a vanilla Hawkes process as a matrix completion problem, and uses graph convolutional recurrent neural networks [70] to solve it. Note that though GCN layers are used in GeoHP, they are used for learning the user/item embeddings. The main architecture of GeoHP is still RNN. Besides, the intensity function of the model is linear with even fewer parameters than the vanilla Hawkes process.

3.2 Applications

Hawkes process is pervasive in every research problem of social network analysis and provides adequate satisfactory results. Mainstream research problems regarding Hawkes processes are listed in Table 3.1.

TABLE 3.1: Research Problems Overview

Problem	**Description**	**Literature**
Network Structure Inference	To infer underlying hidden network structure, or dependence and causality structure from observed data when the network structure is not observable.	[13, 27, 48, 54, 56, 71–77]
Information Diffusion	To analyze, predict, maneuver, intervene or track an information diffusion process, such as campaigning propaganda promotion and fake news diffusion intervention. Usually the network structure is assumed to be known.	[19, 51, 59, 78, 79]
Network Evolution & Dynamics	To model an evolutionary network the time axis, assumed the network changes overtime. Subtopics include link prediction, community emergence and expansion, etc.	[9, 21, 59, 80, 81]
Community Detection & Nodes Clustering	These two topics are both aiming at uncovering and extracting the sub-structure of networks. Though this two terms are interchangeable, the nuance is community detection emphasizes interactive data, whereas nodes clustering leverages individual data, especially nodes' attributes.	[17–20]
Events Clustering & Topic Detection	To cluster events occurred in the network. An emblematic example is to cluster document streams. Topic detection can be viewed as an events clustering problem, which is to discover topics from events accompanying with text content, usually combined with topic models.	[21–26]
Popularity Prediction	To understand why a song, a tweet, of an item spread virally through social networks and become popular, and predict future popularity. This applications would help service providers tackle information overload and make information delivery more efficient.	[28, 53, 82–89]

Table 3.1 – continued from the previous page.

Problem	Description	Literature
Recommendation	To unveil the underlying relation between items or users based on the events information.	[8–13]

Network Structure Inference. Can we unveil the hidden network of social influence exclusively from the observable asynchronous event sequences, when the structure of the network is inaccessible or intangible? This kind of need is ubiquitous in reality. Take product recommendation for example. The advice from a "close" friend, no matter in general or topological sense, can be really decisive for the purchasing intention. Thus, knowing the interpersonal influence represented as a graph will help companies more precisely promote their products. Indeed, Hawkes process can cater to such demand. As we mentioned before, the parameter $\boldsymbol{A}$ encodes the structure information, which is widely used to represent the adjacency matrix of the network to infer. Over all the literature, most studies concentrate in the estimation of the contagion matrix $\boldsymbol{A}$, no matter in what context.

- **Regularizers.** According to different conditions and requirements, some constraints sometimes are employed into the contagion matrix $\boldsymbol{A}$. Therefore, some regularizers are applied so that $\boldsymbol{A}$ can yield to these constraints such as sparsity, or low-rank. Frequently used regularizers in network inference with Hawkes are listed in Table 3.2.

- **Granger Causality.** Another branch in the study of Hawkes process applied in Social Network Analysis which is strongly related to network inference is to learn the *Granger causality graph*, which is also called *local independence graph*. The Granger causality graph presents that an event in the dimension i will directly influence the occurrence of the events in other dimensions. Suppose $\boldsymbol{M} = \{1, 2, \ldots, M\}$ is the set of nodes in the network. The marginal counting process $N_m(t)$ is *Granger-noncausal* for $N_{m'}(t)$ given $\boldsymbol{N}_{\boldsymbol{M}\setminus\{m,m'\}}(t)$ if the intensity function $\lambda_m(t)$ is measurable with respect to history $\mathcal{H}_{\boldsymbol{M}\setminus m}(t^-)$. Otherwise $N_m(t)$ is *Granger-causal* for $N_{m'}(t)$ [90] .With the above definition

of Granger noncausality, Granger causality can be easily defined in the case of multi-dimensional Hawkes process. It suffices to the case that there is a directed edge $i \longrightarrow j$ if and only if $\phi_{ji}(t) = 0$ for all $t \in [0, t]$ ([54]). Given non-zero decay kernels, the Granger causality graph is solely determined by the contagion matrix $\boldsymbol{A}$.

TABLE 3.2: Regularizers for Network Structure

Regularizer	**Expression**	**Constraint**
L_1 Norm	$\sum_{i,j=1}^{M} \lvert a_{ij} \rvert$	Sparsity
Frobenius Norm	$\lVert \boldsymbol{A} \rVert_F^2 = \sum_{i,j=1}^{M} a_{ij}^2$	Avoiding overfitting
Nuclear Norm	$\lVert \boldsymbol{A} \rVert_N = \sum_{i=1}^{r(\boldsymbol{A})} \sigma_i$, where $r(\boldsymbol{A})$ is the rank of matrix $\boldsymbol{A}$ and σ_i's are the singular value.	Low-rank
$L_{2,1}$ Norm	$\lVert \boldsymbol{A} \rVert_{L_{2,1}} = \sum_{j=1}^{M} \sqrt{\sum_{i=1}^{M} \lVert a_{ij} \rVert_2}$	Group Sparsity
Symmetry Norm	$\lVert \boldsymbol{A} - \boldsymbol{A}^T \rVert_F$	Symmetry

Information Diffusion Intervention. Information diffusion intervention is to manage, control, maneuver or track information diffusion process. Modeling information diffusion can be view as an inverse process of the network inference. The task for information diffusion modeling is to manage and direct the information flow given a beforehand known network $G = (V, E)$, whereas network inference is to discover the unknown network structure from the information diffusion process, as shown in Figure 3.1.

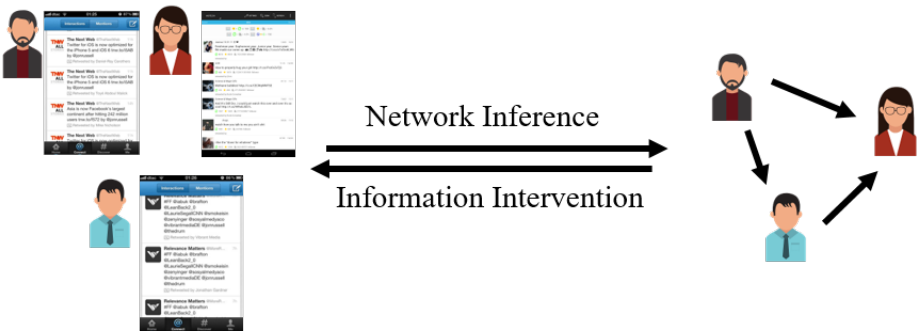


FIGURE 3.1: An illustration of the difference between network inference and information diffusion intervention. Network inference is infer the unobservable network structure from events information, while information intervention is to navigate information flow with the network known.

An application regarding information diffusion is the information intensification and mitigation, which is a traditional research problem in computer science. Recently, some researchers attempt to apply Hawkes processes for the task. [51] explores the idea of optimizing multi-stage campaigning over social networks using Hawkes processes. A closed form relationship between the expected total intensity $\mathbb{E}[\lambda(t)]$, which represents the overall network exposure of the campaign, and the intensity function $\mu(t)$ of exogenous events is derived in this chapter. By altering the exogenous intensity function, combined with different strategy and objective function, campaigning activities and propaganda can be intervened. [78] proposes an intervention algorithm for mitigating fake news diffusion by combining reinforcement learning with Hawkes processes. The strategy is to optimize the performance of real news propagation over the network, and balance users' exposure to real and fake news under budget constraint. A policy iteration method is used for optimization.

Popularity Prediction. Predicting the popularity of online multimedia content is an important problem for the practice of information dissemination and consumption. These applications are of great concern to the social network service providers, since they want to know whether the database will overload, and they can improve the efficiency of content distribution and finding the most valuable and influential information from massive amounts of content.

Hawkes processes are valid when dealing with this problem. As a probabilistic method, Hawkes processes can directly model the distribution as well as the expectation of the volume of events at certain time. Popularity prediction with Hawkes processes has been applied to tweet delivery [74, 82, 84], trending video detection

[28, 53, 86, 87], and patent citation prediction [85] , etc.

The task for popularity prediction is to infer the information evolution after observation window $[0, t]$. Here we introduce a typical modeling method via the decomposition of the Hawkes processes [74, 82, 84, 86]. The core for popularity prediction with Hawkes is to estimate the ultimate number of events $\boldsymbol{N}_\infty$ [82, 84], or the ultimate intensity function defined by $\lim_{t\to\infty} \boldsymbol{\lambda}(t|\mathcal{H}(\tau)) \triangleq \boldsymbol{\lambda}_\infty$ [74]. Fortunately, one similar result can be easily obtained by limit theorems as we discussed in Chapter 2.2.2. Suppose $v = 1$, we directly apply Eq. (2.8), yielding

$$\boldsymbol{\lambda}_\infty = (\boldsymbol{I} - \boldsymbol{\Gamma})^{-1}\boldsymbol{\mu},$$

where $\boldsymbol{\Gamma} = \int_0^\infty \boldsymbol{\Phi}(u)du$. Next, how should we deal with $\boldsymbol{\lambda}_\infty$? According to different constraints and objectives, which can be represented by a function f, we just need to estimate $\mathbb{E}f(\boldsymbol{\lambda}_\infty)$ [74].

Recommendation. Recommendation is another crucial application in social network services. A good recommender system can improve user experience and user viscosity. The target for recommendation is to match users to the right items. Similar to popularity prediction, recommendation is to predict the most possible event that may occur to a user, like what items the user will purchase, and who the user will friend next. Recommendation is make personalized suggestions for users, while popularity prediction is to analyze collective behaviors, like what items will be purchased the most, which video will go viral.

The most fundamental algorithms in recommendation are collaborative filtering and latent factor models. These models are static and fail taking into consideration the temporal behavior and the recurrent activities of users. Sometimes, to predict when the user will buy is as important as to predict what the user will buy. Some studies apply Hawkes processes to recommender systems and have obtained satisfactory results [11].

The basic idea for recommendation with Hawkes process is to couple the user set and the item set. Every element in the Cartesian product of the user set and the item set is a dimension in Hawkes process, as illustrated in the Figure 3.2. A basic

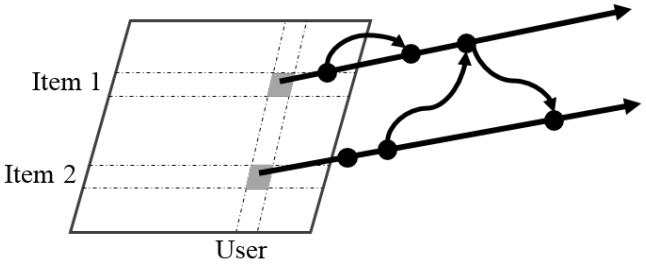


FIGURE 3.2: An illustration of Hawkes process for recommender systems. Each entry of the User-Item matrix can be regarded as a dimension in Hawkes process and generates an event sequence.

conditional intensity function is defined by,

$$\lambda_{u,i}(t) = \mu_{u,i} + \sum_{u' \in U} \sum_{i' \in I} \int_0^t \phi_{ui,u'i'}(t-s) dN_k(s),$$

where $u, u' \in U$ and $i, i' \in I$ are the user set and the item set, respectively. Let $\boldsymbol{\mu} = [\mu_{u,i}]$ be the base intensity matrix, which represents the probability of an event that randomly occurs and is not triggered by other users. If we want to reduce the number of parameters, latent factor models can be applied jointly. For example, we can use two vectors $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ to approximate $\boldsymbol{\mu}$. The exogenous intensity matrix is formulated as:

$$\boldsymbol{\mu} = \boldsymbol{\theta}\boldsymbol{\beta}^T.$$

Two central questions regarding recommendation are how to recommend the most desirable item to a user, and the next returning time of a user to an item. With the above model, this two questions can be addressed easily.

Clustering. Clustering is a pervasive application in social network analysis. By combining Hawkes processes and some cluster techniques, we can involve not only the static features, but also the temporal information, which cannot be utilized by traditional algorithms. Clustering with Hawkes can be categorized into two manners. One is clustering events, usually with features [23]. The other is to cluster the nodes or event sequences [24, 68].

Part I

Models

Chapter 4

Tweedie Hawkes Process: Linking Features with Heavy-tailed Excitations

4.1 Motivation

Self-exciting event sequences are ubiquitous. In such an event sequence, the occurrence of an event will raise the probability of triggering succeeding events. The self-exciting nature often brings about outbreaks of events in a short period of time. People care about why an outbreak happens. Why are certain tweets re-tweeted so many times but not the others? What factors activate an outbreak of a certain epidemic? To answer these questions, an effective and interpretable tool to model and understand outbreaks is needed.

A typical tool for modeling self-/mutually excited data is Hawkes process. Despite its success in many applications, traditional Hawkes processes and most Hawkes-related models are not competent in capturing outbreaks. They mainly suffer from one or many of the following drawbacks:

- *invariant excitation*—the probability of events triggering subsequent ones are either same or i.i.d. distributed;

The work in this chapter has been published as Li, T., & Ke, Y. (2020). Tweedie-Hawkes Processes: Interpreting the Phenomena of Outbreaks. The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020. New York, USA.[1]

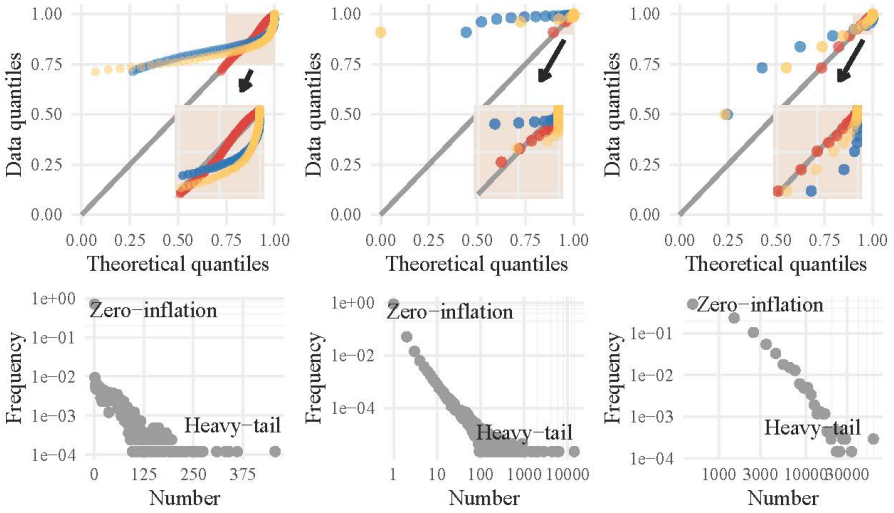


FIGURE 4.1: The omnipresence of zero-inflated and heavy-tailed distributions. Columns: Datasets of degrees of the wiki vote network [5], the number of retweets in a retweet network [6] and the amount of claims in vehicle insurance [7]. Top row: Q-Q plots between the actual and thoeretical quantiles of Tweedie (●), Gaussian (●) and exponential (●) distributions. Bottom row: actual distributions of each datasets.

- *neglecting content/features*—some models only consider the temporal information of the event sequences, but the contents are usually neglected;
- *weak interpretability*—parameters in the model are inaccessible, especially for those combined with neural networks and non-parametric techniques;
- *unrealistic distribution of aggregation*—Taylor's power law [91], which states that the variance of species population density is proportional to a fractional power of the mean, is more natural and common for population and aggregation, whereas some existing works simply adopt Gaussian distribution;
- *failure to sub-criticality*—sub-criticality is a property that the diffusion process produces finite number of events, which many existing works fail to consider.

In this chapter, I propose a Bayesian model called *Tweedie-Hawkes Process* (THP). The model parameterizes the excitation parameter in Hawkes process with a Tweedie regression [92] over event features, and provides a solution to all the aforementioned drawbacks.

Why Tweedie distribution is more realistic? There are two reasons. First, Tweedie distribution obeys Taylor's power law [91]. This law is applicable to

many circumstances, such as the spatial distribution of Colorado beetle [93] and daily turnovers of stocks traded on the NYSE [94]. Hence, equipped with this law, Tweedie distribution is powerful in modeling data that exhibit aggregation (outbreak) phenomena. Second, Tweedie distribution has two important characteristics: *heavy-tail* and *zero-inflation*. Zero-inflation means the probability has a large mass at zero, resulting a natural sparsity. Empirical studies such as [95] and [96], especially in social networks, suggest that many population distributions are heavy-tailed and zero-inflated. In the context of self-excited data, the majority of events are "silent" (i.e., do not have much excitation effect), while a small percentage of events would trigger numerous descendants (in fact, this is how outbreaks are formed). I call the former "silent majority" and the latter "vocal minority". Tweedie distribution is able to well describe this phenomena.

Contributions. The contributions of this chapter are summarized as follows:

- I propose a Bayesian model combined Hawkes processes and Tweedie distribution called THP, which is able to model outbreaks of self-exciting event sequences and understand the influential factors behind. By leveraging on Tweedie distribution, THP is able to capture the "silent majority" and "vocal minority" in excitation effects, which is also dependent on the features of events.

- I develop an effective mean-field variational EM learning algorithm for model inference. Several theoretical properties of THP, including the sub-criticality and the local optima and convergence of the learning algorithm are also presented. A novel kernel bandwidth selection method is proposed.

- I apply THP to 4 tasks in the experiments, to show the versatility and effectiveness of the model. Two applications to Epidemiology and information diffusion analysis demonstrate the potential of the model. Experimental results also show that THP outperforms the state-of-the-art baselines in data fitting and event prediction.

4.2 Background on Tweedie Regression

Tweedie regression is a generalized linear model (GLM) [97] with the response variable following the Tweedie distribution [98]. Tweedie distribution belongs to the class of the exponential dispersion models [92] (EDMs). The probability density function of an EDM is defined by:

$$f(y|\theta,\psi) = c(y|\psi)\exp\left(\frac{y\theta - b(\theta)}{\psi}\right), \qquad y \in \mathbb{R}_{\psi}. \tag{4.1}$$

Here y is a random variable, θ is called the canonical parameter, and ψ the dispersion parameter. $b(\theta)$ is the cumulant function, and $c(y|\psi)$ is a known function. It is easy to verify that the expectation of y, denoted as η, equals the derivative of $b(\theta)$:

$$\eta \triangleq \mathbb{E}y = b'(\theta). \tag{4.2}$$

In GLM, the mean of the response variable y is connected to the explanatory variables x in the linear predictor via a smooth and invertible link function g such that $g(\eta) = \boldsymbol{x}'\boldsymbol{\beta}$. Here $\boldsymbol{\beta}$ is the regression coefficients to be inferred. Note that ψ is a nuisance parameter in the estimation of beta. I thus preset ψ and treat it as a constant in this chapter.

If y follows a Tweedie distribution, denoted by $y \sim \text{Tweedie}_p(\eta,\psi)$, the variance $\mathbb{V}(y)$ and the mean $\eta = \mathbb{E}(y)$ obeys Taylor's power law [91],

$$\mathbb{V}(y) = \psi\mathbb{E}(y)^p, \tag{4.3}$$

where $p \notin (0,1)$. To reformulate the probability density function according to Eq. (4.2), θ and $b(\theta)$ can be written as,

$$\theta = \frac{\eta^{1-p}}{1-p}, \qquad b(\theta) = \frac{\eta^{2-p}}{2-p},$$

$$c(y;\psi) = \begin{cases} \frac{1}{y}\sum_{k=1}^{\infty} \frac{y^{k\gamma}}{[\frac{(p-1)^{\gamma}}{2-p}]^k \psi^{k(1+\gamma)} k!\Gamma(k\gamma)} & y > 0, \\ 1 & y = 0, \end{cases}$$

where

$$\gamma = \frac{2-p}{p-1}.$$

4.3 The Tweedie-Hawkes Process

The idea of the Tweedie-Hawkes process (THP) is to parameterize and randomize the excitation parameter α by the features associated with each event. More specifically, our model defines different α for different events. The intuition is that, different events characterized by different features should have different excitation effects on triggering new events. I achieve this by defining the α in the original Hawkes process as a random variable drawn from a Tweedie distribution. The event features are then naturally incorporated through Tweedie regression. The essential part of our method is to perform a Bayesian treatment on Hawkes. Our THP is detailed as follows.

Consider a sequence of events (i.e., a realization) $\{(t_i, \boldsymbol{x}_i)\}$, $i = 1, \ldots, n$, where $t_i \in [0, T]$ is a timestamp in the observation window and $\boldsymbol{x}_i \in \mathbb{R}^m$ is the corresponding m-dimensional feature vector. Note that I adopt a fixed design setting here, which means that the events associate with the features are served as input. Features are treated as fixed affects here, thus they do not appear in the likelihood function. A discussion on this setting is given in the supplementary material. I denote the timestamp vector by $\boldsymbol{t} = \{t_1, \ldots, t_n\}'$, which is modeled by a Hawkes process with the excitation parameters $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_n\}'$ and the base parameter μ:

$$\boldsymbol{t} \sim \text{Hawkes}(\mu, \boldsymbol{\alpha}|\boldsymbol{\eta}). \tag{4.4}$$

The log-likelihood of Hawkes process is given by,

$$\ln p(\boldsymbol{t}|\mu, \boldsymbol{\alpha}) = \sum_{i}^{n} \ln \lambda(t|\mathcal{H}_{t_-}) - \int_0^T \lambda(t|\mathcal{H}_{t_-})dt. \tag{4.5}$$

Each α_i is drawn from a Tweedie prior distribution, with its mean η_i being a regression over the corresponding feature vector $\boldsymbol{x}_i$,

$$\alpha_i \sim \text{Tweedie}_p(\eta_i, \psi), \tag{4.6}$$

$$\theta_i = \eta_i^{1-p}/(1-p), \tag{4.7}$$

$$g(\eta_i) = \boldsymbol{x}_i^T \boldsymbol{\beta}. \tag{4.8}$$

Here $g(\eta)$ is a monotonically increasing link function that connects features and the prior distributions. The choice of $g(\eta)$ affects the convergence and sub-criticality of the model. Further details will be discussed later. The prior distribution of α can be written as,

$$\ln p(\boldsymbol{\alpha}|\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[\ln c(\alpha_i; \psi) + \frac{\alpha_i \theta_i - b(\theta_i)}{\psi} \right]. \tag{4.9}$$

Combining Eq. (4.5)(4.9) yields the complete log-likelihood function,

$$\ln p(\boldsymbol{t}, \boldsymbol{\alpha}|\mu, \boldsymbol{\beta}) = \ln p(\boldsymbol{t}|\mu, \boldsymbol{\alpha}) + \ln p(\boldsymbol{\alpha}|\boldsymbol{\beta}). \tag{4.10}$$

The incomplete likelihood can be written as,

$$p(\boldsymbol{t}|\mu, \boldsymbol{\beta}) = \int_{\boldsymbol{\alpha}} p(\boldsymbol{t}|\mu, \boldsymbol{\alpha}) p(\boldsymbol{\alpha}|\boldsymbol{\beta}) d\boldsymbol{\alpha}. \tag{4.11}$$

4.4 Inference

As illustrated in the chapter, the complete log-likelihood function of THP can be written as,

$$\begin{aligned}
&\ln p(\boldsymbol{t}, \boldsymbol{\alpha}|\mu, \boldsymbol{\beta}) \\
=&\ln p(\boldsymbol{t}|\mu, \boldsymbol{\alpha}) + \ln p(\boldsymbol{\alpha}|\boldsymbol{\beta}). \\
=&\sum_{i=1}^{n} \ln \left(\mu + \sum_{j=1}^{i-1} \alpha_j \phi(t_i - t_j) \right) - \mu T \\
&- \sum_{i=1}^{n} \alpha_i \int_0^T \phi(t - t_i) dt + \sum_{i=1}^{n} \left(\ln c(\alpha_i; \psi) + \frac{\alpha_i \theta_i - b(\theta_i)}{\psi} \right).
\end{aligned} \tag{4.12}$$

Integrating the latent variable $\boldsymbol{\alpha}$, I have the incomplete likelihood,

$$p(\boldsymbol{t}|\mu, \boldsymbol{\beta}) = \int_{\boldsymbol{\alpha}} p(\boldsymbol{t}|\mu, \boldsymbol{\alpha}) p(\boldsymbol{\alpha}|\boldsymbol{\beta}) d\boldsymbol{\alpha}. \tag{4.13}$$

E-step. I use Tweedie distributions as variational ones for the hidden variables α's. Meanwhile, I factorize the joint distribution of α's by the mean field approximation:

$$q(\tilde{\boldsymbol{\alpha}}) = \prod_{i=1}^{n} q(\tilde{\alpha}_i),$$

where $q(\tilde{\alpha}_i)$ is a Tweedie distribution with corresponding parameter $\tilde{\eta}_i$ (expectation), and the tilde represents variational. Though Tweedie distribution are not the conjugate priors for Hawkes processes, there are two reasons for this choice. First, it results in an evidence lower bound (ELBO) that can be easily approximated by a concave function, guaranteeing local optima. Second, the thorny normalization term $c(y|\psi)$ in the posterior can be canceled out. The approximation turns out to be accurate, which justifies our choice of Tweedie as variational distributions. The ELBO for approximating the true posterior distribution of $\tilde{\alpha}_i$ with $q(\tilde{\alpha}_i)$ is given by,

$$\text{ELBO}(\tilde{\alpha}_i) = \int_{\tilde{\alpha}_i} \mathop{\mathbb{E}}_{j \neq i} \ln(\boldsymbol{t}, \boldsymbol{\alpha}|\mu, \boldsymbol{\beta}) \; q(\tilde{\alpha}_i) \; \mathrm{d}\tilde{\alpha}_i - \mathop{\mathbb{E}}_{i} \ln q(\tilde{\alpha}_i),$$

where $\mathop{\mathbb{E}}_{j\neq i}$ is the notation for the expectation over all the other $\tilde{\alpha}_j$'s but $\tilde{\alpha}_i$, therefore leading to a function of the variational variable $\tilde{\alpha}_i$. It can be written explicitly as,

$$\begin{aligned}
\mathop{\mathbb{E}}_{j\neq i} \ln(\boldsymbol{t}, \boldsymbol{\alpha}|\boldsymbol{\beta}, \mu) &= \int \cdots \int_{j\neq i} \ln(\boldsymbol{t}, \boldsymbol{\alpha}|\boldsymbol{\beta}, \mu) \prod_{j\neq i} q(\tilde{\alpha}_j) \mathrm{d}\tilde{\alpha}_j \\
&\approx \sum_{k=1}^{n} \ln \left[\mu + \sum_{j\neq i}^{k-1} \eta_k \phi(t_k - t_j) + \mathbb{I}_{\{k\leqslant k-1\}} \tilde{\alpha}_i \phi(t_k - t_i) \right] - \\
&\quad \sum_{k=1}^{n} \frac{\sum_{j\neq i}^{k-1} \eta_j^p \phi^2(t_k - t_j)}{2 \left[\mu + \sum_{j\neq i}^{k-1} \eta_k \phi(t_k - t_j) + \mathbb{I}_{\{i\leqslant k-1\}} \tilde{\alpha}_i \phi(t_k - t_i) \right]^2} - \\
&\quad \tilde{\alpha}_i \int_0^T \phi(t - t_i) dt + \ln c(\tilde{\alpha}_i; \psi) + \frac{\tilde{\alpha}_i \theta_i}{\psi} + \text{const.}
\end{aligned}$$

Here I applied the multivariate second-order Taylor expansion of the expectation of logarithm of random variable that

$$\mathbb{E} f(\boldsymbol{X}) \approx f(\mathbb{E}(\boldsymbol{X})) - \frac{1}{2} \mathbb{E} \left[(\boldsymbol{X} - \mathbb{E}\boldsymbol{X})' \boldsymbol{H}(f) (\boldsymbol{X} - \mathbb{E}\boldsymbol{X}) \right],$$

where $\boldsymbol{H}$ is the Hessian matrix. Note that the intractable term $\ln c(\tilde{\alpha}_k; \psi)$ can be canceled in the ELBO if the same dispersion parameter ψ is assumed, as it also appears in the entropy $\mathop{\mathbb{E}}_{k} \ln q(\tilde{\alpha}_k)$. Then I continue to look at the ELBO, which can be explicitly written as,

$$\begin{aligned}
\text{ELBO} \approx &\sum_{k=i+1}^{n} \ln \left\{ \mu + \sum_{j\neq i}^{k-1} \eta_j \phi(t_k - t_j) + \mathop{\mathbb{E}}_{\tilde{\alpha}_i} \tilde{\alpha}_i \phi(t_k - t_i) \right\} - \\
&\sum_{k=i+1}^{n} \frac{\mathop{\mathbb{V}}_{\tilde{\alpha}_i} \tilde{\alpha}_i \phi^2(t_k - t_i) + \sum_{j\neq k}^{k-1} \psi \eta_j^p \phi^2(t_k - t_j)}{2 \left[\mu + \sum_{j\neq i}^{k-1} \eta_i \phi(t_k - t_j) + \mathop{\mathbb{E}}_{\tilde{\alpha}_i} \tilde{\alpha}_i \phi(t_i - t_i) \right]^2} - \\
&\mathop{\mathbb{E}}_{\tilde{\alpha}_i} \tilde{\alpha}_i \int_0^T \phi(t - t_i) dt + \frac{\mathop{\mathbb{E}}_{\tilde{\alpha}_i} \tilde{\alpha}_i (\theta_i - \tilde{\theta}_i) + b(\tilde{\theta}_i)}{\psi} + \text{const.}
\end{aligned} \tag{4.14}$$

I apply the Taylor expansion of $\mathbb{E}X^{-2} = (\mathbb{E}X)^{-2} - O((\mathbb{E}X)^{p-4})$ again in above derivation. In the ELBO, i.e. Eq. (4.14), the expectation $\mathbb{E}\tilde{\alpha}_k$ and variance $\mathbb{V}\tilde{\alpha}_k$ can be respectively replaced by $\tilde{\eta}_k$ and $\psi\tilde{\eta}_k^p$. Here I need to highlight the difference of θ_k and $\tilde{\theta}_k$. The one without a tilde over it represents the parameter corresponding

to α_k which is learned in the last iteration step, i.e. M-step, whereas the other with a tilde over it is from the variational distribution and exactly what I need to optimize in the variational Bayesian updating step, i.e. E-step.

Reformulating the equation, we obtain the objective function to be maximized for the expectation $\tilde{\eta}_i$ for the i-th hidden variable $\tilde{\alpha}_i$ is given by,

$$\begin{aligned}\max_{0<\tilde{\eta}_i<1} \quad & \sum_{k=i+1}^{n} \ln\left\{\mu + \sum_{j\neq i}^{k-1} \eta_j \phi(t_k - t_j) + \tilde{\eta}_i \phi(t_k - t_i)\right\} \\ & - \sum_{k=i+1}^{n} \frac{\psi\left[\tilde{\eta}_i^p \phi^2(t_k - t_i) + \sum_{j\neq i}^{k-1} \eta_j^p \phi^2(t_k - t_j)\right]}{2\left[\mu + \sum_{j\neq i}^{k-1} \eta_k \phi(t_k - t_j) + \tilde{\eta}_i \phi(t_k - t_i)\right]^2} \\ & - \tilde{\eta}_i \int_0^T \phi(t - t_i)dt + \frac{\tilde{\eta}_i\left(\eta_i^{1-p} - \tilde{\eta}_i^{1-p}\right)}{\psi(1-p)} + \frac{\tilde{\eta}_i^{2-p}}{\psi(2-p)}.\end{aligned} \tag{4.15}$$

This objective function, which is to maximize the ELBO with respect to the variational parameters, however, is not necessarily convex. The second term is a convex-convex fractional function, which spoils the convexity [99]. We find that the objective function is concave when $p < 2$, ensuring convergence to global optima. We summarize this finding in Lemma 4.1 and defer the proof to the Appendix.

Lemma 4.1 (Concavity)**.** The ELBO is concave in $\tilde{\eta}_i$ for each i, if $p < 2$.

M-step. I maximize the expected complete log-likelihood using the variational distribution $q(\tilde{\boldsymbol{\alpha}})$. The Q function with the current parameters μ' and $\boldsymbol{\beta}'$ is given by,

$$Q\left(\mu, \boldsymbol{\beta} \middle| \mu', \boldsymbol{\beta}'\right) = \mathop{\mathbb{E}}_{\tilde{\boldsymbol{\alpha}}} \ln(\boldsymbol{t}, \tilde{\boldsymbol{\alpha}} | \mu, \boldsymbol{\beta}). \tag{4.16}$$

From Eq. (11) in the chapter, the Q function can be decomposed into two parts,

$$Q\left(\mu, \boldsymbol{\beta} \middle| \mu', \boldsymbol{\beta}'\right) = \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{\alpha}}} \ln p(\boldsymbol{t} | \mu, \tilde{\boldsymbol{\alpha}})}_{Q(\mu|\mu', \boldsymbol{\beta}')} + \underbrace{\mathop{\mathbb{E}}_{\tilde{\boldsymbol{\alpha}}} \ln p(\tilde{\boldsymbol{\alpha}} | \boldsymbol{\beta})}_{Q(\boldsymbol{\beta}|\mu', \boldsymbol{\beta}')}, \tag{4.17}$$

where the first part contains only μ and the second part only $\boldsymbol{\beta}$. Therefore the optimization of the Q function with respect to μ and $\boldsymbol{\beta}$ can be decoupled. More explicitly,

$$Q(\mu|\mu', \boldsymbol{\beta}') = \sum_{i=1}^{n} \ln \left[\mu + \sum_{j}^{i-1} \tilde{\eta}_i \phi(t_i - t_j) \right] - \mu T + \text{const},$$

and,

$$Q(\boldsymbol{\beta}|\mu', \boldsymbol{\beta}') = \frac{1}{\psi} \sum_{i=1}^{n} \frac{\tilde{\eta}_i \eta_i^{1-p}}{1-p} - \frac{\eta_i^{2-p}}{2-p} + \text{const},$$

where $\eta_i = g^{-1}(\boldsymbol{x}_i^T \boldsymbol{\beta})$. It is worth noting that the Q functions here do not contain the current parameters μ' and $\boldsymbol{\beta}'$ outwardly. In fact they are embedded in the variational expectation $\tilde{\eta}_i$. Though the closed-form solutions do not exist, the decoupled Q functions can be easily optimized by various gradient-based methods as they are continuous and smooth. In our implementation, I adopt the Broyden-Fletcher-Goldfarb-Shanno [100] algorithm. The first derivatives of the Q-function are:

$$\frac{\partial Q}{\partial \mu} = \sum_{i=1}^{n} \frac{\mu}{\mu + \sum_{j}^{i-1} \tilde{\eta}_i \phi(t_i - t_j)} - T,$$

$$\frac{\partial Q}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \frac{\partial Q}{\partial \eta_i} \frac{\partial \eta_i}{\partial \boldsymbol{\beta}} = \frac{1}{\psi} \sum_{i=1}^{n} (\tilde{\eta}_i - \eta_i) \, \eta_i^{1-p} \frac{\partial \eta_i}{\partial \boldsymbol{\beta}}.$$

4.5 Theoretical Properties

In this section, I present some theoretical properties of THP. Detailed proofs can be found in the supplementary material.

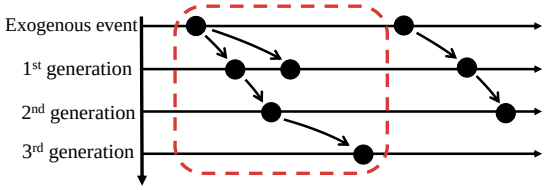


FIGURE 4.2: Illustration on the decomposition of THP. Each black dot stands for an event.

4.5.1 Sub-Criticality and the Link Function

One desirable property of Hawkes processes is *sub-criticality*, which states that the total progeny of each event is a.s. finite [101]. This is an important property as it ensures that the effect of an event will eventually vanish, which is a rule commonly present in many natural phenomena. In this subsection, I show that our THP possesses this important property as long as the link function g satisfies certain conditions. Note that the state-of-the-art model PHP is not sub-critical.

Essentially, the concept of sub-criticality describes a Galton-Watson branching process with a finite total number of events. A Hawkes process can be equivalently interpreted as a collection of branching processes, each centered at an exogenous event [40]. Built upon Hawkes, our THP can also be decomposed likewise. More specifically, the events generated by THP come from two sources:

1. $N^{\dagger}$: the process that generates exogenous events;
2. $N^{\ddagger}_{ij}$: the j-th generation of the i-th exogenous event.

An illustration on the decomposition of THP is given in Fig. 4.2. The event sequence in the red box forms a Galton-Watson branching process. An exogenous event is one triggered by μ whereas endogenous by α. The process $\boldsymbol{N}$ of THP is then the superposition of the above sub-processes:

$$\boldsymbol{N} = \sum_{i=1}^{\boldsymbol{N}^{\dagger}} \sum_{j} \boldsymbol{N}^{\ddagger}_{ij} \tag{4.18}$$

A Galton-Watson branching process is said to be sub-critical if the expected number of events at each generation is smaller than that at the previous one, as formulated below.

Definition 4.2 (Sub-Criticality)**.** A Galton-Watson branching process $N_i^{\ddagger}$ is **sub-critical** if $\mathbb{E}\boldsymbol{N}_{i,j+1}^{\ddagger} < \mathbb{E}\boldsymbol{N}_{i,j}^{\ddagger}$ for each generation $j = 1, 2, \cdots$.

Theorem 4.3 (Sub-Criticality of THP)**.** THP consists of a finite number of sub-critical Galton-Watson branching processes if the link function is, (1) invertible, and (2) mapping $(0, 1)$ onto $\mathbb{R}$. Besides, the number of Galton-Watson branching processes $N^{\dagger}$ is a Poisson process of rate μ.

Proof sketch. The proof is by using the conditional expectation between generations. Applying the expectation equation of non-homogeneous Poisson process, the ratio of the expected number of events between generations is η. If η is bounded by 1, the branching processes comply with Definition 4.2.

4.5.2 Convergence Analysis

The learning algorithm presented in the last section is able to achieve local optima and convergence. Theorem 4.4 states that each iteration of the learning algorithm will consistently increase the likelihood until convergence. The convergence of the model parameters is stated in Theorem 4.5.

Theorem 4.4 (Local Optima)**.** For any $k = 1, 2, \cdots$, I have,

$$\mathcal{L}^{(k+1)} \geq \mathcal{L}^{(k)}, \tag{4.19}$$

where $\mathcal{L}^{(k)} = \ln p(\boldsymbol{t}|\mu^{(k)}, \boldsymbol{\beta}^{(k)})$ denotes the incomplete log-likelihood of k-th iteration in the learning algorithm of THP.

In addition, the convergence of parameters $\left\{\mu^{(k)}\right\}$ and $\left\{\boldsymbol{\eta}^{(k)}\right\}$ can also be guaranteed. The convergence of $\left\{\boldsymbol{\beta}^{(k)}\right\}$ is associated with the link function. This result is illustrated in Theorem 4.5.

Theorem 4.5 (Convergence)**.** If the updating method for $Q(\mu|\mu', \boldsymbol{\beta}')$ and $Q(\boldsymbol{\beta}|\mu', \boldsymbol{\beta}')$ is gradient descent (or Newton-like methods), then as $k \to \infty$, $\|\mu^{(k+1)} - \mu^{(k)}\| \to$

0, $\|\boldsymbol{\eta}^{(k+1)} - \boldsymbol{\eta}^{(k)}\| \to 0$. In particular, the convergence holds for $\boldsymbol{\beta}$ that as $k \to +\infty$, $\|\boldsymbol{\beta}^{(k+1)} - \boldsymbol{\beta}^{(k)}\| \to 0$, if either of the following conditions is satisfied: (1) the link function g is uniformly continuous, and (2) the link function g satisfies the sub-critical conditions stated in Theorem 4.3 and $\eta_i^{(k)} \not\to 0$ or 1 for all i.

4.5.3 Smoothing Kernel Bandwidth Selection

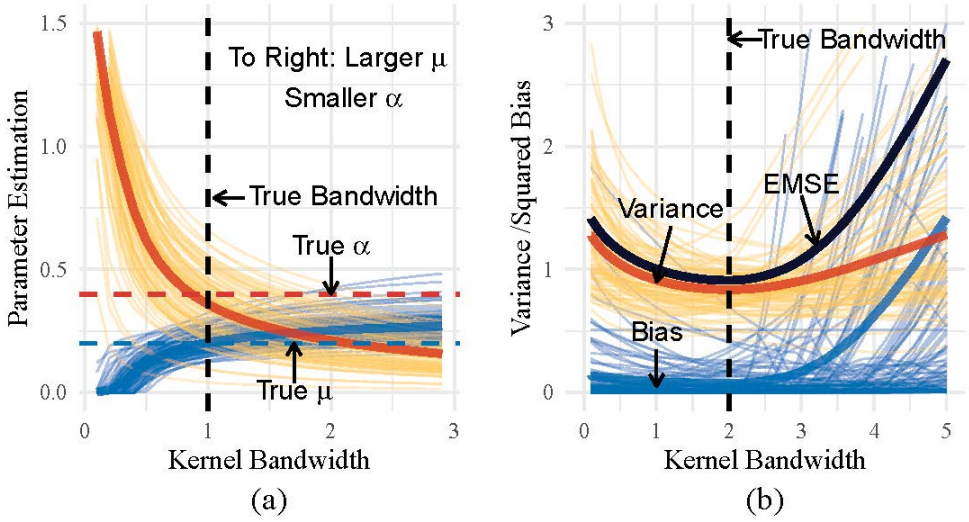


FIGURE 4.3: Illustration on the kernel bandwidth in controlling the trade-offs in exogenous-endogenous events, and in variance-bias. (a): The light yellow curves (—) show the estimation of α with respect to the kernel bandwidth h, while the light blue curves (—) represent the estimation of μ. The solid curves (═) represent the respective expectations of the light curves. (b): Variance (═) and bias (═) as functions of bandwidth h; a decomposition of the EMSE (—).

In this subsection, I look at the problem of selecting the best kernel bandwidth h. A bad choice of h may result in poor parameter estimations that deviate from true ones. Despite its importance, this problem is often neglected in existing works on Hawkes processes.

I consider a vanilla Hawkes process with exponential decay kernels and aim to develop a selection mechanism that applies generally to the models that build upon Hawkes process.

I propose a kernel bandwidth selection method based on the bias-variance trade-off. The main idea is to minimize the expected mean square error (EMSE) on the

integrated intensity, which can be decomposed into three parts, the variance, the squared bias, and the irreducible error.

Consider a vanilla Hawkes process, with conditional intensity function. The integrated intensity at time interval $[t_{i-1}, t_i]$ given any α and μ is defined as $\tau_i(\alpha, \mu) = \int_{t_{i-1}}^{t_i} \lambda(t|\mu, \alpha)\, dt$. Correspondingly, with the true μ^* and α^*, the true integrated intensity is defined as $\tau_i^* = \int_{t_{i-1}}^{t_i} \lambda(t|\mu^*, \alpha^*)\, dt$. According to Proposition 7.4.IV and Lemma 7.4.II in [101] I have with probability 1 that each τ_i has i.i.d. unit exponential distribution:

$$\tau_i^* \underset{i.i.d.}{\sim} \exp(1). \tag{4.20}$$

Note that τ_i is a random variable with the randomness originated from the inter-arrival time. I now define the EMSE given any estimated parameters α and μ:

$$\begin{aligned} \text{EMSE}(\mu, \alpha) &= \sum_{i=1}^{n} \mathbb{E}\left(\tau_i - \tau_i^*\right)^2 \\ &= \sum_{i=1}^{n} \underbrace{\mathbb{E}(\tau_i - \mathbb{E}\tau_i)^2}_{\text{Variance}} + \underbrace{(\mathbb{E}\tau_i - \mathbb{E}\tau_i^*)^2}_{\text{Bias}^2} + \underbrace{\mathbb{E}(\tau_i^* - \mathbb{E}\tau_i^*)^2}_{\text{Irreducible Error}} \end{aligned} \tag{4.21}$$

Eq. (4.21) indicates that the kernel bandwidth h controls the trade-off between the variance and the squared bias. According to Eq. (4.20), I have $\mathbb{E}\tau_i^* = 1$ and $\mathbb{E}\left(\tau_i^* - \mathbb{E}\tau_i^*\right)^2 = 1$. The EMSE with remnant parts can be regarded as a function of α and μ. Applying the least square estimator introduced in [43], both α and μ have closed-form solutions given h,

$$\begin{pmatrix} \mu \\ \alpha \end{pmatrix} = \boldsymbol{Z}(h)^{-1} \boldsymbol{y}(h), \tag{4.22}$$

where

$$\boldsymbol{Z}(h) = \begin{pmatrix} T & \int_0^T \sum_j \phi(t - t_j|h) dt \\ \int_0^T \sum_j \phi(t - t_j|h) dt & \int_0^T \sum_j \phi^2(t - t_j|h) dt \end{pmatrix},$$

$$\boldsymbol{y}(h) = \begin{pmatrix} n \\ \sum_i \sum_j \phi(t_i - t_j | h) \end{pmatrix}.$$

T is the length of the observation window. Combining the least square estimator with Eq. (4.21), I arrive at the EMSE as a function of h, which is then minimized to obtain the best h^*.

Interestingly, I find that the kernel bandwidth also governs the trade-off in generating exogenous and endogenous events. A smaller h means a slower decay on the excitation effect, making an event generate more endogenous offspring. It can be seen that with a true h, the estimation of α and μ is quite close to the ground truth. This demonstrates the necessity and benefits of selecting the best kernel bandwidth in Hawkes processes.

4.6 Experiments

In this section, I demonstrate applications and evaluations on both synthetic and real-world datasets. In the first two tasks, I present two applications to Epidemiology and the diffusion of textual information. Then in Task 3, I present that our model has better aggregation of events, which explains why THP is better at capturing outbreaks of events. Last but not least, I test our model on several real-world datasets for forecasting future events in Task 4, which shows our model outperforms the rival baselines.

4.6.1 Task 1: An Application to the Transmission of MERS-CoV

In this task, I apply THP to study the transmission of Middle East respiratory syndrome-coronavirus (MERS-CoV) in Saudi Arabia in 2017. MERS-CoV is a viral respiratory illness that can cause fever, shortness of breath, Pneumonia, and even death. According to the World Health Organization (WHO), approximately 35% of reported MERS patients have died. There are two major routes for transmission:

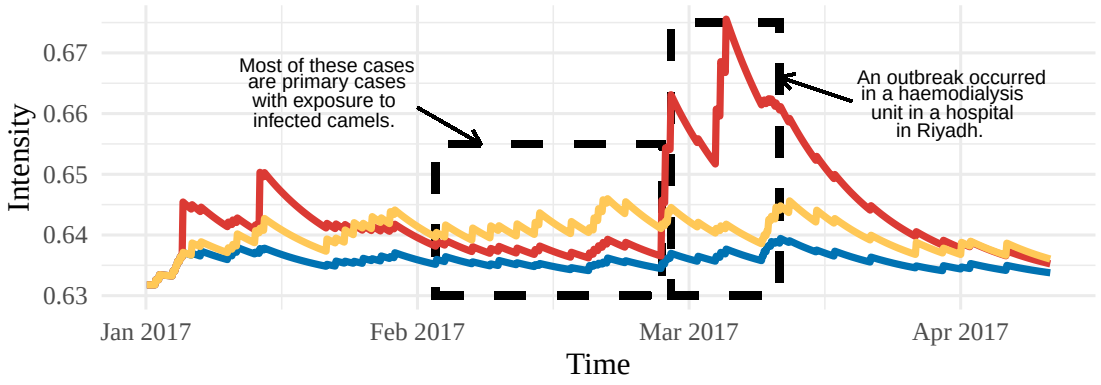


FIGURE 4.4: A decomposition of the intensity function with features: `camel milk consumption` (—), `exposure to MERS-CoV case` (—) and `exposure to camels` (—).

non-human to human transmission, especially from dromedary camels, and human-to-human transmission. However, the transmission patterns of the virus are not fully understood. The aim of this study is to provide some statistical insights for health care workers.

The data is collected from the WHO website, where I study the reported cases in Saudi Arabia in the first 200 days of 2017. The dataset contains 155 cases (i.e., events in our model). I split them into two halves. The first half is for training and the second half for testing. There are three potential risk factors: "exposure to camels", "camel milk consumption", and "exposure to MERS-CoV case". All of them are boolean. They are treated as event features in our model.

I find that "exposure to MERS-CoV case" has the largest regression coefficient, which means that human-to-human transmission is the most statistically important risk factor for MERS-CoV, comparing with the other two factors. Fig. 4.4 visualizes how the contribution of each factor changes over time by decomposing the intensity function. The respective intensity is calculated by using each feature only. By investigating the decomposed intensities in the figure, I have two findings. First, there are big spikes in intensity from roughly day 60 to day 75. This is supported by an outbreak that occurred in a haemodialysis unit in a hospital in Riyadh between 23 February and 16 March 2017. Second, "exposure to MERS-CoV case" accounts more for this outbreak, as the other two intensity curves are far below the red one. This finding is consistent with the WHO's observation that several outbreaks occurred primarily due to community transmission within health care

settings and households. The cases that were infected by direct or indirect contact with dromedary camels tend to happen individually and occasionally.

4.6.2 Task 2: An Application to Information Diffusion of Textual Contents

Hawkes processes are a potent tool for modeling the dynamics of information and have been applied in many works to the propagation of textual contents, such as [25] and [68]. Due to the omnipresence of the Tweedie distribution, THP greatly enhances Hawkes processes when dealing with text-based cascades in social networks, especially in the following aspects:

- **Identifying influential texts.** The most important difference between THP and other Hawkes-related models is that in THP, every event has an individual α, which controls the probability of triggering subsequent events. For those events that have larger α, they are more likely to bring about more events. Therefore, η, which is the expectation of α, can be regarded as a indicator of how influential the text is.
- **Popular topic detection.** The Tweedie regression part of THP explains why an event has a larger α (through Eq. (4.8)). If a bag of words or topic models are used as features, then the value of β represents the contribution of each topic/word, in which
- **Information diffusion modeling based on the latent parent-child relationship of texts.** As a derivative model of Hawkes processes, THP also possesses the branching structure as illustrated in Fig. 4.2, which provides a method to infer the parent-child relationships among texts. As a consequence, the *textual cascade tree*, which is a directed tree showing the propagation of information in timeline, can be inferred. Fig. 4.5 demonstrates a toy example.

I test our model on the `MemeTracker` dataset [102]. Due to the lack of ground truths (labels), I are not able to really evaluate the model's performance on finding popular texts or topics. Alternatively, I aim at a prediction task, which is to forecast future dynamics of event sequences. The results are shown in Task 4 with Table 4.1.

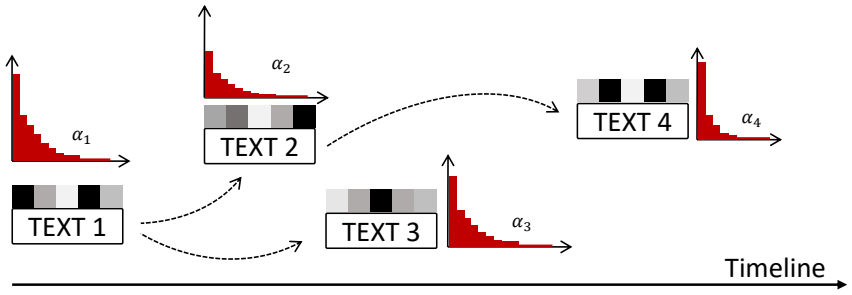


FIGURE 4.5: An illustration of the *textual cascade tree* inferred by THP. Each histogram shows the distribution of α for respective text, which is related to the features and the coefficient $\boldsymbol{\beta}$ through Eq. (4.8). Arrows represent the inferred parent-child relationships, which depend on two factor: (1) temporal distance and (2) textual similarities. Text 1 is a root node. Texts 2, 3 and 4 are descendant of Text 1.

Moreover, a case study on the `MemeTracker` dataset is given in the supplementary material.

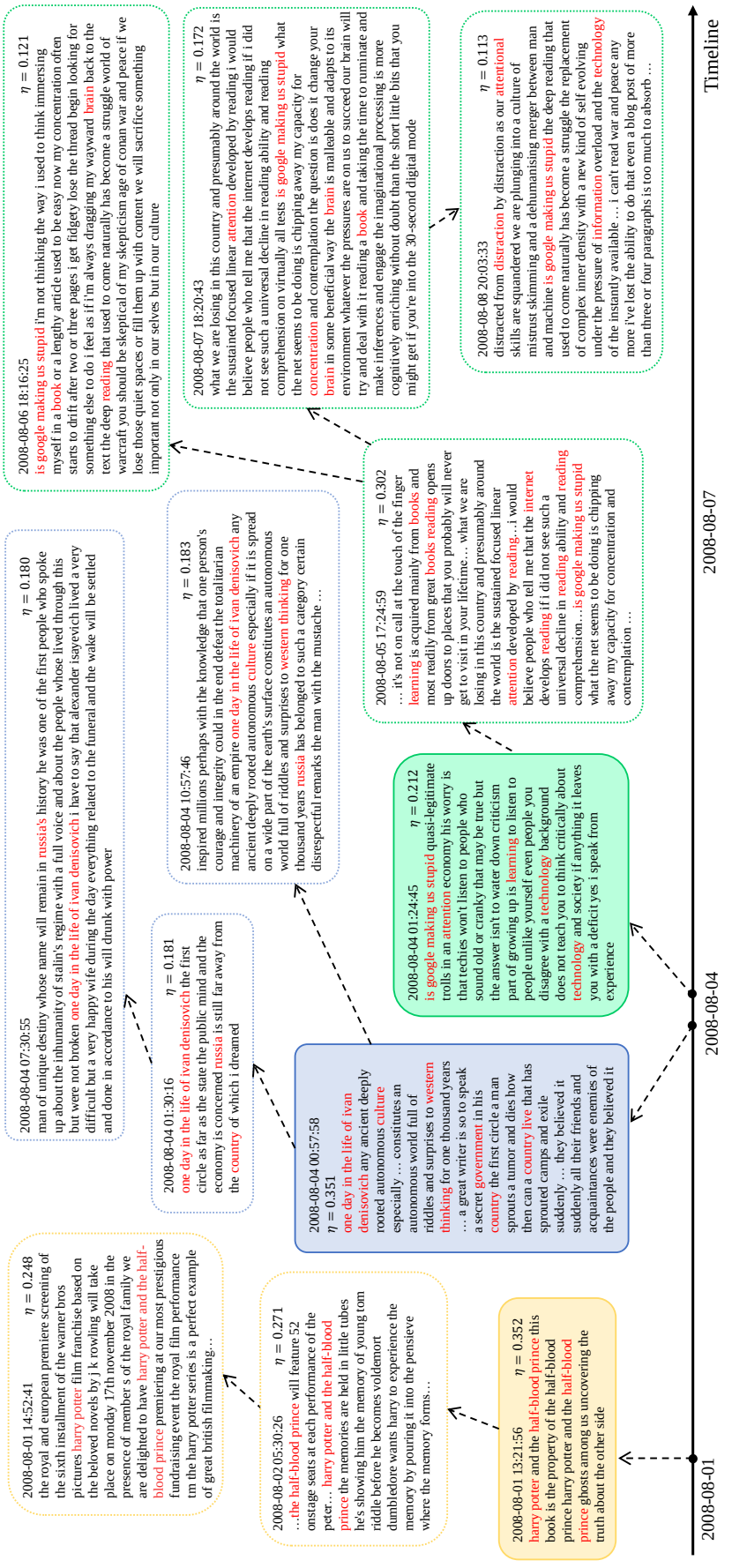


FIGURE 4.6: A Detailed Version of the Information Diffusion Tree in Task 2. A textual cascade tree inferred by THP on `MemeTracker` dataset. Three Trees are presented. η's represent the probabilities of triggering following events. Latent Dirichlet allocation (LDA) is used for generating the features of contents. Arrows represent the inferred "parent-child" relationships. Here the propagation of 3 memes: *harry potter and half-blood prince, one day in the life of ivan denisovich* and *is google making us stupid*, are presented.

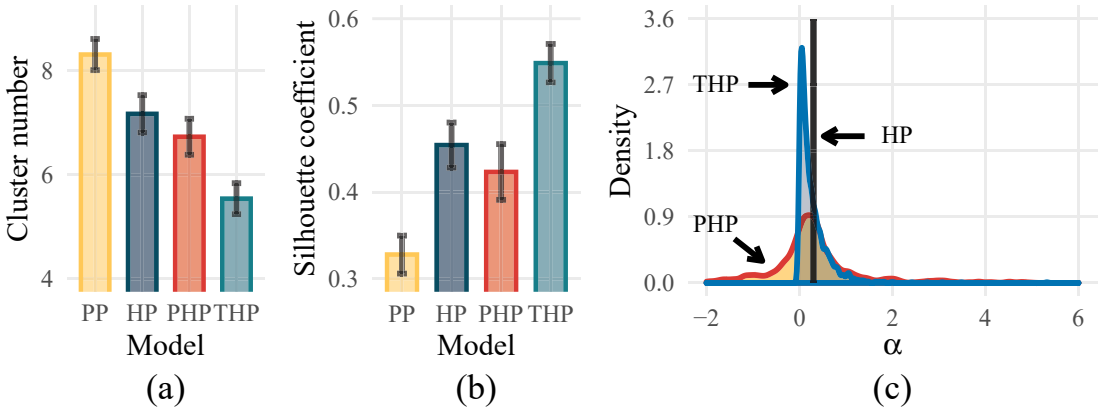


FIGURE 4.7: An illustration of the aggregation of events generated by different models. (a)(b): Number of Clusters and Silhouette coefficients of the clustering of events by DBSCAN. It shows that THP has a better aggregation of events in timelines. (c): Comparison of the estimated distributions of α. THP's zero-inflation and heavy-tail can be seen, whereas PHP and HP result a Gaussian distribution and a scalar, respectively.

4.6.3 Task 3: Temporal Aggregation of Events on Synthetic Dataset

Thanks to the Tweedie component, THP is able to generate event data that are more temporally aggregated. I compare with the following baseline models:

- a basic Poisson process (`PP`),
- a vanilla Hawkes process (`HP`) [103];
- Parametric Hawkes process (`PHP`) [45].

I generate 100 datasets using each model (with the same mean of event numbers, and the same mean of α's for HP, PHP, and THP) and use DBSCAN [104] to cluster the timestamps in each dataset. As shown in Fig. 4.7(a) and 4.7(b), THP obtains the smallest number of clusters and the highest Silhouette coefficient, indicating that it indeed achieves the highest aggregation of events. This is particularly useful for capturing the bursty patterns of real event sequences, where outbreaks of events often occur. Fig. 4.7(c) further shows the distribution of α learned by different models. HP has a fixed α, while PP has $\alpha = 0$. PHP exhibits a Gaussian-like α distribution. In THP, most α's are around 0, and only a very small percentage of

α's have large values. This demonstrates the silent majority and the vocal minority characteristics of THP and explains the highly clustered nature of generated events.

4.6.4 Task 4: Predictions on Real-world Datasets

In this part, I evaluate our model against several state-of-the-art models on 4 real-world datasets in a task of predicting future event sequences. Besides the 3 baselines in Task 3, I also involve 4 more state-of-the-art models to compare with:

- Gaussian Marked Hawkes Processes (`GMHP`) [68];
- Sparse Low-rank Hawkes process (`SLRH`) [105];
- Majorization Minmization Euler-Lagrange algorithm (`MMEL`) [47];
- Granger Causality for Hawkes `(GC)` [27];

It is worth noting that only `PHP`, `GMHP` and our `THP` incorporate both temporal and feature information, whereas the other several baselines only consider the temporal information. The datasets used for testing are:

- `MERS-CoV` (MC): the dataset I use in Task 1.
- `MemeTracker` (MT) [102]: the dataset in the Task 2.
- `IPTV` [106]: the dataset consists of IPTV viewing events, which records the timestamps and the category that the video belongs to.
- `Ieplace` [107] : This dataset contains the check-in histories of users at different locations. The categories of events include food, education, shops, and 10 others.

The sizes of the datasets I use in the above are: 155, 11275, 2916, 948. I divide each dataset into two parts: 60% as training dataset, and 40% as testing dataset.

Two metrics are used to assess the prediction quality:

- `NegLogLik`, negative log-likelihood of the test dataset. The likelihood only considers time, with features excluded.

- `RMSE`, root mean square error of predicting the arrival times of the next N events ($N = 1/5/10$).

A lower `NegLogLik` and `RMSE` means the model can better capture the transmission patterns. Due to the page limit, I list the results of log-likelihood in Table 4.1. For more results, please refer to the supplementary material.

4.7 Summary

In this chapter, I propose Tweedie-Hawkes Processes, which is powerful in modeling and understanding outbreaks of events. Our model leverages upon the Tweedie distribution in capturing the "silent majority" and "vocal minority" in excitation effects. I showed that the model enjoys a number of theoretical merits and outperforms the state-of-the-art baselines in data fitting and event prediction when tested on several real-world datasets. I also showed the versatility of our model by applying to two tasks in Epidemiology and information diffusion analysis, respectively. It is worth noting that our model is built upon one-dimensional Hawkes processes. Multi-dimensional Hawkes processes can be modified into a THP by setting the dimensionality as a feature. Besides, the distribution of α can be something beyond Tweedie or Gaussian (e.g., binomial distribution). Our model offers a feasible framework to such problem that one can change the Tweedie regression to any other generalized linear models (GLMs).

The model can also be seen as a probabilistic graphical model, where the Tweedie distribution can be seen as a prior upon the infectivity parameters. As a probabilistic model, it is convenient to extend Hawkes processes to more complex ones via the probabilistic graphical model framework. In the next chapter, we will see another Hawkes-based probabilistic graphical model for the task of transfer learning.

TABLE 4.1: The log-likelihood of the predicted future event sequences on various real-world datasets.

Dataset	Model	Log-likelihood	RMSE of Next Events		
			1	5	10
MC	Ours	**-76.344**	**1.213** (0.208)	**2.748** (0.301)	**3.601** (0.355)
	PHP	-94.617	1.261 (0.195)	2.969 (0.250)	4.847 (0.437)
	GMHP	-80.066	1.472 (0.250)	3.808 (0.287)	5.585 (0.375)
	HP	-94.389	1.965 (0.250)	4.262 (0.373)	6.376 (0.569)
	GC	-103.287	1.315 (0.300)	3.029 (0.338)	4.279 (0.391)
	SLR	-95.838	1.301 (0.301)	2.859 (0.328)	3.830 (0.368)
	MML	-103.693	1.404 (0.302)	2.938 (0.289)	3.902 (0.356)
	PP	-104.612	1.374 (0.299)	3.707 (0.377)	6.078 (0.483)
MT	Ours	**128.455**	**0.183** (0.027)	0.466 (0.082)	**1.006** (0.129)
	PHP	126.922	0.202 (0.033)	0.704 (0.107)	1.340 (0.170)
	GMHP	108.048	0.275 (0.037)	**0.462** (0.048)	1.376 (0.105)
	HP	102.111	0.197 (0.036)	0.499 (0.069)	1.176 (0.128)
	GC	125.889	0.191 (0.032)	0.483 (0.060)	1.319 (0.087)
	SLR	52.827	0.203 (0.030)	0.552 (0.066)	1.764 (0.154)
	MML	108.340	0.232 (0.030)	0.534 (0.053)	1.469 (0.080)
	PP	107.982	0.223 (0.027)	0.493 (0.059)	1.253 (0.116)
IPTV	Ours	**-783.270**	**18.965** (1.541)	**199.170** (7.443)	300.050 (7.888)
	PHP	-930.218	24.686 (1.059)	226.182 (4.584)	365.688 (5.504)
	GMHP	-969.380	27.377 (3.106)	200.314 (6.672)	**291.003** (10.086)
	HP	-965.310	25.894 (1.298)	233.696 (3.745)	367.950 (5.398)
	GC	-1081.648	24.270 (1.900)	201.379 (7.183)	313.940 (8.137)
	SLR	-967.232	23.402 (1.361)	232.057 (3.836)	367.647 (5.846)
	MML	-966.795	23.691 (1.265)	225.500 (4.841)	331.317 (7.105)
	PP	-1027.998	24.836 (0.935)	234.828 (3.682)	370.261 (5.307)
Ieplace	Ours	**-1114.906**	**7.117** (1.424)	**18.408** (1.728)	**41.904** (4.275)
	PHP	-1123.601	10.412 (1.772)	26.386 (3.221)	68.002 (5.078)
	GMHP	-1187.383	8.038 (1.335)	19.662 (2.844)	55.327 (3.652)
	HP	-1121.381	10.691 (2.227)	20.494 (2.618)	56.514 (4.533)
	GC	-1281.400	10.837 (2.562)	20.936 (2.073)	106.169 (2.981)
	SLR	-1348.445	10.791 (2.180)	22.675 (3.175)	95.834 (3.856)
	MML	-1214.991	9.182 (2.145)	22.110 (2.569)	81.229 (3.558)
	PP	-1151.463	10.531 (1.575)	22.935 (2.659)	55.327 (3.652)

Chapter 5

Network Transfer for Hawkes Processes: Learning from Cross-domain Temporal and Feature Information

5.1 Motivation

Vanilla Hawkes processes and most related models only deal with temporal information of event sequences. Besides temporal information, cross-domain knowledge can also be beneficial. Take cold-start problem for example. Suppose we would like to build recommender systems for the e-commerce start-ups, but we do not have adequate data on hand. In this case we can use data from mature businesses like Amazon and transfer the knowledge cross platforms. Similar application scenarios are commonly encountered. Recently transfer learning [31], which has been a topic of active interest, sheds light on how to exploit the knowledge from other domains and help improve the performance of models. However, to the best of our knowledge, none of the existing works has explored transfer learning for Hawkes processes.

The work in this chapter has been published as Li, T., Wei, P., & Ke, Y. (2018). Transfer Hawkes Processes with Content Information. Proceedings - IEEE International Conference on Data Mining, ICDM, 2018, 1116–1121. Singapore. [2]

Then, how can we leverage cross-domain information? The most important characteristic of Hawkes processes is that they link the occurrence of events up to the network structure. As suggested by their initial name, Hawkes processes are designed for modeling self- and mutually exciting pattern of event sequences. Mathematically, most information concerning a Hawkes process is embedded in the infectivity matrix, which is a network representing one's influence over another. Therefore, the key to transferring a Hawkes process is to transfer the network, i.e., the infectivity matrix. More specifically, we assume that the pattern of influence in social media is stable. For example, if Alice always comments on Bob's sharing and posts on Facebook, then she probably tends to do the same things to Bob's activities on Twitter. This is the basic assumption that the infectivity matrices among the same users on different platforms should be analogous. When training a Hawkes process on a dataset that has only a few records (target domain), the estimated network structure may not be accurate. If we can learn a more accurate infectivity matrix from plentiful data (source domain), and transfer the matrix to target domain. the performance of the model should be better. The overarching framework of network transfer for Hawkes processes is shown in Figure 5.1.

Although Hawkes processes cater to the need for dealing with temporal information from event cascades and have achieved satisfactory results, many models related to Hawkes processes fail to take into account the features associated with each event. The feature information, which is also conducive to improving accuracy and robustness for various tasks, should also be considered. For instance, Alice posts something on her own timeline. Bob finds the post is hilarious, and retweets. It can be seen that here might be an edge in the underlying social network from Alice to Bob, since Alice is influential on Bob. In this case Hawkes processes can be applied to capture the mutual influence. However, if what Alice and Bob post are totally irrelevant, the edge seems untenable. Neglecting feature information, Hawkes model may be deceptive. In this study, we would like to pay special attention to leverage the often-neglected feature information.

In this chapter, I investigate the idea of network transfer for Hawkes processes. We augment Hawkes processes with both cross-domain and feature information, and instantiate the idea by two models: transfer **H**ybrid **L**east **S**quare for **H**awkes (trHLSH) and **B**ayesian **T**ransfer **H**awkes **M**odel (BTHM). These two models are from the perspectives of frequentism and Bayesianism, respectively. The trHLSH

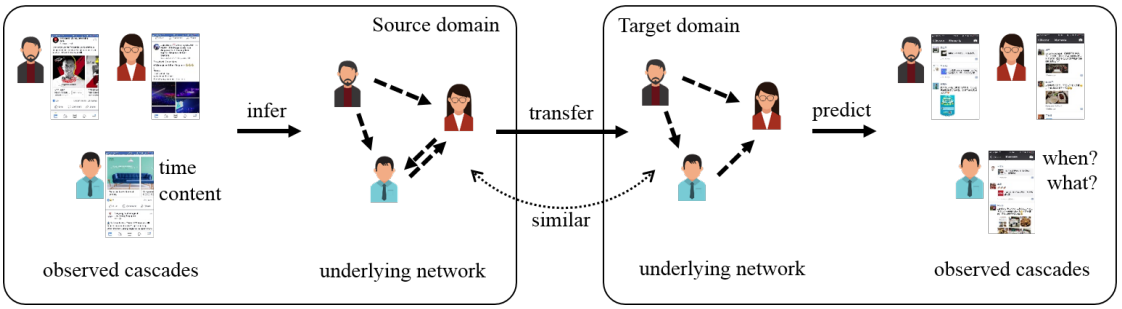


FIGURE 5.1: An illustration of network transfer for Hawkes processes. We first infer the underlying hidden network from the temporal and feature information in source domain. We assume the same users, and the networks of users in source domain and target domain should be similar. Then we transfer network structure to target domain and predict next event.

is based on the least square estimation of Hawkes and auto-regression of feature vectors. A regularizer term is added to control that estimation of parameters on the target domain will not deviate too much from those on the source domain. The BTHM is a Bayesian mixture model based on the clustering representation and branching structure of Hawkes processes. A variational expectation maximization (VEM) algorithm is proposed for learning model parameters. Both of the models capture all of the three aforementioned types of information:

- temporal information of event cascades;
- event feature information;
- cross-domain knowledge.

Transfer learning. Transfer learning [31] aims to transfer knowledge from one domain (called the *source domain*) to help the tasks of another domain (called the *target domain*). Typically, the source and target domains are related in the sense that they share some common knowledge which can be transferred across domains. Based on the knowledge transferred, transfer learning is categorized into the feature-based one, the instance-based one, and the parameter-based one. In this chapter, we focus on the parameter transfer, which is to discover the relationship of parameters, between two different Hawkes processes. Specifically, in the experiments on real-world data, we use the event sequences from Twitter as source domain and those from Facebook as target domain. We assume that the network structures of both domains are similar and transferable. Therefore in the proposed models, the

knowledge we transfer is actually the infectivity parameter α's, which describe the network structure.

The noteworthy novelties and contributions of this chapter can be summarized as follows:

- To the best of our knowledge, the models I propose are the first attempts to deal with temporal and feature information, as well as cross-domain transfer simultaneously.
- I instantiate the idea in two manners: a parametric model trHLSH, and a Bayesian model BTHM. We derive effective and flexible learning algorithms for parameter inference. trHLSH, which enjoys many computational merits and is able to be solved efficiently by quadratic programming. BTHM follows Bayesian framework and learns parameters in source and target domain synchronously.
- I test our models on both synthetic datasets and two real-world ones. The experimental results on both synthetic and real-world data demonstrate superiority of our model in terms of parameter recovery and prediction.

5.2 The Proposed Models

In this section we present the two proposed models and their learning algorithms. We first introduce the parametric model, trHLSH, followed by a Bayesian model, BTHM.

5.2.1 A Parametric Model with trHLSH

5.2.1.1 Leveraging features

In Section 2.3.1, I describe the least square estimator for Hawkes process that models temporal information. Now I illustrate an auto-regression model for incorporating the features. For an event that occurs at t, its features can be represented as a feature vector $\boldsymbol{f}(t)$, or $\boldsymbol{f}_{ik}$ corresponding to t_{ik}, via feature embedding. Assume

features and time are independent, and $\boldsymbol{f}^i(t)$ can be defined by all the events (with their features) that happen prior to t as,

$$\boldsymbol{f}_i(t) = \sum_{j=1}^{M} \frac{1}{N_j(t)} \sum_{k=1}^{N_j(t)} w_{ijk} \boldsymbol{f}_{j,N_j(t)+1-k} + w_{i0}\mathbf{1} + \epsilon, \tag{5.1}$$

where w_{ijk} is the regression coefficient, w_{i0} the intercept and ϵ the Gaussian noise. $\mathbf{1}$ is a vector with each entry as 1. The intuition behind is what you may post depends on what you have read. The feature is modeled as a linear combination of the historical events. However the number of coefficients equals to the total number of events, which makes the estimation intractable as a result of the short rank of the design matrix. Therefore we assume that all the features of the same dimension share the same coefficient. Under this assumption, Eq. (5.1) becomes,

$$\boldsymbol{f}_i(t) = \sum_{j=1}^{M} \frac{w_{ij}}{N_j(t)} \sum_{k=1}^{N_j(t)} \boldsymbol{f}_{jk} + w_{i0}\mathbf{1} + \epsilon. \tag{5.2}$$

The least square estimation for w_{ij} is,

$$\min_{w_{ij}} \sum_{i=1}^{M} \sum_{k=1}^{n_i} \left\| \boldsymbol{f}_{ik} - \sum_{j=1}^{M} \frac{1}{N_j(t_{ik})} \sum_{k'=1}^{N_j(t_{ik})} w_{ij} \boldsymbol{f}_{jk'} - w_{i0}\mathbf{1} \right\|_2^2. \tag{5.3}$$

Let

$$\boldsymbol{F}_{ik} = \left(\mathbf{1}, \frac{1}{N_1(t_{ik})} \sum_{k'=1}^{N_1(t_{ik})} \boldsymbol{f}_{jk'}, \ldots, \frac{1}{N_M(t)} \sum_{k'=1}^{N_M(t_{ik})} \boldsymbol{f}_{jk'} \right), \tag{5.4}$$

$$\boldsymbol{w}_i = (w_{i0}, w_{i1}, \ldots, w_{iM})', \tag{5.5}$$

$$\boldsymbol{\Psi}_i = \sum_{k=1}^{n_i} (\boldsymbol{F}_{ik})' \boldsymbol{F}_{ik}, \tag{5.6}$$

$$\boldsymbol{\psi}_i = \sum_{k=1}^{n_i} (\boldsymbol{F}_{ik})' \boldsymbol{f}_{ik}. \tag{5.7}$$

Eq.(5.3) can be rewritten as an optimization problem,

$$\min_{\boldsymbol{w}} \quad \boldsymbol{w}'\boldsymbol{\Psi}\boldsymbol{w} - 2\boldsymbol{w}'\boldsymbol{\psi}. \tag{5.8}$$

Here the superscript i is also omitted.

5.2.1.2 Hybrid Least Square for Hawkes (HLSH)

Note that α_{ij} quantifies the influence from dimension j to i which is inferred by temporal information. Symmetrically w_{ij} also reflects the influence from dimension j to i which is, however, inferred by the features. Since α_{ij} and w_{ij} both reflect the network structure, they should be similar. Therefore I impose a $L2$ norm regularization of $\boldsymbol{\theta} - \boldsymbol{w}$. Combining Eq.(2.14) and Eq.(5.8), now we have the objective function of HLSH:

$$\begin{aligned} \min_{\boldsymbol{\theta},\boldsymbol{w}} \quad & \boldsymbol{\theta}'\boldsymbol{Z}\boldsymbol{\theta} - 2\boldsymbol{\theta}'\boldsymbol{y} + \eta_1\left(\boldsymbol{w}'\boldsymbol{\Psi}\boldsymbol{w} - 2\boldsymbol{w}'\boldsymbol{\psi}\right) + \eta_2\left\|\boldsymbol{\theta} - \boldsymbol{w}\right\|_2^2, \\ \text{s.t.} \quad & \boldsymbol{\theta} \geqslant \mathbf{0}, \end{aligned} \tag{5.9}$$

where η_1 and η_2 are hyperparameters of the regularizations.

5.2.1.3 Transfer HLSH

I now take into account cross-domain knowledge for augmenting the HLSH model. The objective function is formulated as,

$$\begin{aligned} \min_{\boldsymbol{\theta}_T,\boldsymbol{w}_T} \quad & \boldsymbol{\theta}_T'\boldsymbol{Z}\boldsymbol{\theta}_T - 2\boldsymbol{\theta}_T'\boldsymbol{y} + \eta_1\left(\boldsymbol{w}_T'\boldsymbol{\Psi}\boldsymbol{w}_T - 2\boldsymbol{w}_T'\boldsymbol{\psi}\right) + \\ & \eta_2\left\|\boldsymbol{\theta}_T - \boldsymbol{w}_T\right\|_2^2 + \eta_3\left\|\boldsymbol{\theta}_T - \boldsymbol{\theta}_S\right\|_2^2 + \eta_4\left\|\boldsymbol{w}_T - \boldsymbol{w}_S\right\|_2^2, \\ \text{s.t.} \quad & \boldsymbol{\theta}_T \geqslant \mathbf{0}. \end{aligned} \tag{5.10}$$

Here $\boldsymbol{\theta}_S$ and $\boldsymbol{w}_S$ are the parameters that are pre-learned from the source domain by HLSH. The last three regularization terms and their meanings are:

- $\left\|\boldsymbol{\theta}_T - \boldsymbol{w}_T\right\|_2^2$: constraint on similarity of the network structure learned from temporal and feature information;
- $\left\|\boldsymbol{\theta}_T - \boldsymbol{\theta}_S\right\|_2^2$ and $\left\|\boldsymbol{w}_T - \boldsymbol{w}_S\right\|_2^2$: constraints on similarity of the parameters learned from target and source domains.

Algorithm 3 presents the procedure of the learning algorithm above.

Algorithm 3: trHLSH: Transfer Hybrid Least Square for Hawkes

input : $\boldsymbol{S}_S, \boldsymbol{S}_T$, η_1, η_1', η_2, η_2', η_3, η_4, $\psi(\cdot)$
output : $\boldsymbol{\theta}_S$, $\boldsymbol{\theta}_T$, $\boldsymbol{w}_S$, $\boldsymbol{w}_T$
Calculate $\boldsymbol{Z}_S$, $\boldsymbol{y}_S$, $\boldsymbol{\Psi}_S$, $\boldsymbol{\psi}_S$, $\boldsymbol{Z}_T$, $\boldsymbol{y}_T$, $\boldsymbol{\Psi}_T$, $\boldsymbol{\psi}_T$ by Eq.(2.13), (2.12), (5.6), (5.7);
Calculate $\boldsymbol{\theta}_S$ and $\boldsymbol{w}_S$ with η_1', η_2' by Eq.(5.9);
Calculate $\boldsymbol{\theta}_T$ and $\boldsymbol{w}_T$ with η_1, η_2, η_3, η_4 by Eq.(5.10);
return $\boldsymbol{\theta}_S$, $\boldsymbol{\theta}_T$, $\boldsymbol{w}_S$, $\boldsymbol{w}_T$.

5.2.2 A Bayesian Generative Model with BTHM

Next we illustrate the details of the components in the Bayesian model, which is referred to as Bayesian Transfer Hawkes Model (BTHM).

The general idea of BTHM follows the framework as presented in Figure 5.1. We can observe that there are three groups of parameters in Hawkes process: μ_i's, α_{ij}'s and β_{ij}'s. We focus on the transfer of network structure, which is embodied in α_{ij}'s, from the source domain to the target domain. This is because the mutual influences between users are relatively constant and stable even if in distinctive domains. As illustrated before, μ_i's and β_{ij}'s control the arrival of immigrants and diffusion of events. They are highly dependent on the user viscosity, which is usually different cross platforms. Since μ_i's and β_{ij}'s are not what we care in the transfer learning task, we do not assume any prior distribution for them.

Next I discuss the generating procedures of our model.

5.2.2.1 Generating the infectivity parameter for both domains

We assume that the infectivity matrices $\boldsymbol{A}_S$ and $\boldsymbol{A}_T$ for both target and source domains are from the same prior, which is a Weibull distribution,

$$\boldsymbol{A}_S, \boldsymbol{A}_T \sim \text{Weibull}(\boldsymbol{\Theta}, \boldsymbol{\Pi}).$$

$\boldsymbol{\Theta}$ and $\boldsymbol{\Pi}$ are the parameters of the prior distribution, with the dimension of $M \times M$.

5.2.2.2 Generating the base intensity

We adopt the same assumption as [24] that $\boldsymbol{\mu}$ has a Rayleigh distribution

$$\begin{aligned}\boldsymbol{\mu}_S &\sim \text{Rayleigh}(\boldsymbol{\gamma}_S),\\ \boldsymbol{\mu}_T &\sim \text{Rayleigh}(\boldsymbol{\gamma}_T),\end{aligned}$$

where $\boldsymbol{\gamma}_S$ and $\boldsymbol{\gamma}_T$ are the parameters for source and target domains, respectively.

5.2.2.3 Generating event timestamps t's

We generate a series of event timestamps using Hawkes process:

$$t_i|\mathcal{H}_t \sim \text{HawkesProcess}(\lambda_i(t|\mathcal{H}_t)). \tag{5.11}$$

Applying the branching structure as illustrated in Section 2.2.1, the intensity function of the Hawkes process is defined by,

$$\begin{cases}\lambda(t_{ik}|\boldsymbol{\mu}, \boldsymbol{A}, \Phi = z_0) = \mu_i,\\ \lambda(t_{ik}|\boldsymbol{\mu}, \boldsymbol{A}, \Phi = t_{jk'}) = \alpha_{ij}\psi(t_{ik} - t_{jk'}).\end{cases} \tag{5.12}$$

Here we omit the subscript S and T since the relationships are symmetric for both domains. To recap the meaning of the notations, $\lambda_i(t|\boldsymbol{\mu}, \boldsymbol{A}, \Phi = t_{jk}) = \alpha_{ij}\psi(t - t_{jk})$ is the conditional intensity function of the subprocess $\mathcal{P}_{ijk}$ with process center t_{jk} over the i-th dimension. We generate timestamps from a Hawkes process using the parameters generated above. It is notable that the hidden variables Φ's are embedded in the generation of Hawkes process. After generating the timestamps, we generate the corresponding features.

5.2.2.4 Generating features

Suppose $\boldsymbol{f}_{ik}$, $\boldsymbol{f}_{jk'}$ are feature vectors corresponding to t_{ik}, $t_{jk'}$, respectively. We assume the distribution of a feature vector $\boldsymbol{f}$ depends on its ancestor event, but independent of time. It is intuitive that if event t_{ik} triggers another event at $t_{jk'}$, then $\boldsymbol{f}_{ik}$ should be similar to $\boldsymbol{f}_{jk'}$. Therefore we generate features from Gaussian distribution with the feature vector of its ancestor as the mean. Mathematically,

Gaussian prior is equivalent to the L_2 norm regularization of $\boldsymbol{f}_{ik} - \boldsymbol{f}_{jk'}$. Also, we assume independence of each variable and the covariance matrix is given by $\sigma^2 \boldsymbol{I}$, where σ is the scalar of variance and $\boldsymbol{I}$ is the identity matrix. σ_1 is for immigrant events and σ_0 for offspring events. For immigrant events, features are similarly generated from a Gaussian distribution with mean $\boldsymbol{\theta}$. Both σ and $\boldsymbol{\theta}$ are predefined. The probabilistic density function $\tau(\boldsymbol{f})$ is given by,

$$\begin{cases} \tau(\boldsymbol{f}|\theta, \sigma, \Phi = \psi_0) = \dfrac{1}{\sqrt{2\pi}\sigma_0^d} \exp\left(-\dfrac{\|\boldsymbol{f} - \boldsymbol{\theta}_0\|_2^2}{2\sigma_0^2}\right) & \text{(immigrant)}, \\ \tau(\boldsymbol{f}|\theta, \sigma, \Phi = t_k^j) = \dfrac{1}{\sqrt{2\pi}\sigma_1^d} \exp\left(-\dfrac{\|\boldsymbol{f} - \boldsymbol{f}_{jk}\|_2^2}{2\sigma_1^2}\right) & \text{(offspring of } t_{jk}\text{)}, \end{cases} \tag{5.13}$$

where d is the dimensionality of feature space.

5.2.3 Variational Inference of BTHM

Instead of using the arduous sampling methods such as Markov chain Monte Carlo (MCMC), we derive a Variational Expectation Maximization (VEM) learning algorithm for the inference. The key inferential problem is to find the joint distribution which is denoted by p^* below of all the variables:

$$\begin{aligned} p^* \coloneqq & p(\boldsymbol{S}_T, \boldsymbol{S}_S, \boldsymbol{\mu}_T, \boldsymbol{\mu}_S, \boldsymbol{A}_T, \boldsymbol{A}_S | \boldsymbol{\Theta}, \boldsymbol{\Pi}, \boldsymbol{\gamma}_T, \boldsymbol{\gamma}_S, \boldsymbol{\Phi}_T, \boldsymbol{\Phi}_S) \\ = & p(\boldsymbol{S}_T | \boldsymbol{\mu}_T, \boldsymbol{A}_T, \boldsymbol{\Phi}_T) p(\boldsymbol{S}_S | \boldsymbol{\mu}_S, \boldsymbol{A}_S, \boldsymbol{\Phi}_S) \times \\ & p(\boldsymbol{\mu}_T | \boldsymbol{\gamma}_T) p(\boldsymbol{\mu}_S | \boldsymbol{\gamma}_S) p(\boldsymbol{A}_T | \boldsymbol{\Theta}, \boldsymbol{\Pi}) p(\boldsymbol{A}_S | \boldsymbol{\Theta}, \boldsymbol{\Pi}). \end{aligned}$$

Here only $\boldsymbol{S}_T$ and $\boldsymbol{S}_S$ are observable, and $\boldsymbol{\mu}_T$, $\boldsymbol{\mu}_S$, $\boldsymbol{A}_T$, $\boldsymbol{A}_S$ are latent variables. To break down the joint distribution of all the observable and latent variables, the components are expressible in the specific form,

$$\begin{aligned} \log p(\boldsymbol{S}|\boldsymbol{\mu}, \boldsymbol{A}, \boldsymbol{\Phi}) = & \sum_{i=1}^{M} \sum_{k=1}^{n_i} \log \left\{ \mu_i + \sum_{j=1}^{M} \alpha_{ij} \sum_{k'} \psi(t_{ik} - t_{jk'}) \right\} - \sum_{i=1}^{M} \mu_i T \\ & - \sum_{i,j=1}^{M} \alpha_{ij} \sum_{k,k'} \int_0^T \psi(t_{ik} - t_{jk'}) + \sum_{i=1}^{M} \sum_{k=1}^{n_i} \log \tau(\boldsymbol{f}_{ik}|\theta, \sigma, \Phi_{ik}), \end{aligned}$$

In the experiment, we use a much larger σ_0 than σ_1, as we want the features of immigrant events to be more scattered.

$$\log p(\boldsymbol{\mu}|\boldsymbol{\gamma}) = \sum_{i=1}^{M} \left[\log \mu_i - {\mu_i}^2/(2{\gamma_i}^2) - 2\log \gamma_i\right],$$

and

$$\log p(\boldsymbol{A}|\boldsymbol{\Theta}, \boldsymbol{\Pi}) = \sum_{i=1}^{M}\sum_{j=1}^{M}\left[\log \Pi_{ij} + (\Pi_{ij}-1)\log a_{ij} - \Pi_{ij}\log \Theta_{ij} - \left(\frac{a_{ij}}{\Theta_{ij}}\right)^{\Pi_{ij}}\right].$$

It is worth noting that we omit the subscripts of S and T, since all the above equations hold on both source and target domain.

5.2.3.1 E-step

In the E-step, we use the variational distributions to approximate the posterior distribution of the latent variables. In this chapter, we consider a mean-field variational distribution factorization,

$$q(\boldsymbol{\mu}_T, \boldsymbol{\mu}_S, \boldsymbol{A}_T, \boldsymbol{A}_S) = q(\boldsymbol{\mu}_T|\tilde{\boldsymbol{\gamma}}_T)q(\boldsymbol{\mu}_S|\tilde{\boldsymbol{\gamma}}_S)q(\boldsymbol{A}_T|\tilde{\boldsymbol{\Theta}}, \tilde{\boldsymbol{\Gamma}})q(\boldsymbol{A}_S|\tilde{\boldsymbol{\Theta}}, \tilde{\boldsymbol{\Gamma}}),$$

where the tildes "$\sim$" denotes the parameters for variational distribution. The inference of the parameters turns into an optimization problem by maximizing the Evidence Lower Bound (ELBO):

$$\text{ELBO} = \mathbb{E}_q \log p^* - \mathbb{E}_q \log q(\boldsymbol{\mu}_T, \boldsymbol{\mu}_S, \boldsymbol{A}_T, \boldsymbol{A}_S).$$

We approximate expectation of the logarithm of a random variable by the Taylor expansion:

$$\mathbb{E}\log X = \log \mathbb{E}[X] - \frac{\mathbb{V}[X]}{2\mathbb{E}^2[X]} + \mathcal{O}\left(\frac{\mathbb{E}[(X-\mathbb{E}X)^3]}{\mathbb{E}^3[X]}\right).$$

The first term of the ELBO can be approximated by,

$$\begin{aligned}
\mathbb{E}_q \log p^* \approx & \sum_{S,T} \sum_{i=1}^{M} \sum_{k=1}^{n_i} \log \left\{ \mathbb{E}\mu_i + \sum_{j=1}^{M} \mathbb{E}\alpha_{ij} \sum_{k'} \psi(t_{ik} - t_{jk'}) \right\} \\
& - \frac{\left\{ \mathbb{V}\mu_i + \sum_{j=1}^{M} \mathbb{V}\alpha_{ij} \left(\sum_{k'} \psi(t_{ik} - t_{jk'}) \right)^2 \right\}}{2 \left\{ \mathbb{E}\mu_i + \sum_{j=1}^{M} \mathbb{E}\alpha_{ij} \sum_{k'} \psi(t_{ik} - t_{jk'}) \right\}^2} \\
& - \sum_{i=1}^{M} \mathbb{E}\mu_i T - \sum_{i,j=1}^{M} \mathbb{E}\alpha_{ij} \sum_{k'} \int_0^T \psi(t - t_{jk'}) dt \\
& + \sum_{i=1}^{M} \left[\log \mathbb{E}\mu_i - \frac{\mathbb{V}\mu_i}{2 \left[\mathbb{E}\mu_i \right]^2} - \frac{\mathbb{E}{\mu_i}^2}{(2\gamma^{i2})} - 2 \log \gamma_i \right] \\
& + \sum_{i=1}^{M} \sum_{j=1}^{M} \left\{ \log \Pi_{ij} + (\Pi_{ij} - 1) \left[\log \mathbb{E}\alpha_{ij} - \frac{\mathbb{V}\alpha_{ij}}{2 \left[\mathbb{E}\alpha_{ij} \right]^2} \right] \right. \\
& \left. - \Pi_{ij} \log \Theta_{ij} - \mathbb{E} \left(\frac{\alpha_{ij}}{\Theta_{ij}} \right)^{\Pi_{ij}} \right\}. \qquad (5.14)
\end{aligned}$$

Here the first summation is over both source and target domains. The parameter Π controls how failure rate varies over time. The expectation and variance of μ and α under the variational distributions are,

$$\begin{aligned}
\mathbb{E}_q \mu_i &= \sqrt{\frac{\pi}{2}} \tilde{\gamma}_i, \\
\mathbb{V}_q \mu_i &= \frac{4 - \pi}{2} \left(\tilde{\gamma}_i \right)^2, \\
\mathbb{E}_q \alpha_{ij} &= \tilde{\Theta} \Gamma(1 + \frac{1}{\tilde{\Pi}}), \\
\mathbb{V}_q \alpha_{ij} &= \tilde{\Theta}^2 \Big\{ \Gamma(1 + \frac{2}{\tilde{\Pi}}) - \left[\Gamma(1 + \frac{1}{\tilde{\Pi}}) \right]^2 \Big\}.
\end{aligned}$$

The second term in the ELBO can be written explicitly as,

$$\begin{aligned}
\mathbb{E}_q \log q(\boldsymbol{\mu} | \tilde{\boldsymbol{\gamma}}) &= \sum_{i=1}^{M} 1 + \log \left(\frac{\tilde{\gamma}_i}{\sqrt{2}} \right) + c, \\
\mathbb{E}_q \log q(\boldsymbol{A} | \tilde{\boldsymbol{\Theta}}, \tilde{\boldsymbol{\Gamma}}) &= \sum_{i=1}^{M} \sum_{j=1}^{M} c \left(1 - \frac{1}{\tilde{\Pi}_{ij}} \right) + \log \left(\frac{\tilde{\Theta}_{ij}}{\tilde{\Pi}_{ij}} \right) + 1,
\end{aligned}$$

where c is the Euler-Mascheroni constant.

5.2.3.2 M-step

In the M-step we update the branch structure $\mathbf{\Phi}$ and the hyper-parameters γ and Θ. The hyper-parameter γ and Θ is updated by maximizing the Q-function defined by,

$$Q(\gamma_i, \Theta_{ij}|\tilde{\gamma}_i, \tilde{\Theta}_{ij}) = \mathbb{E}_q \log p^*.$$

The Q-function can by decomposed by,

$$Q(\gamma_i, \Theta_{ij}|\tilde{\gamma}_i, \tilde{\Theta}_{ij}) = Q(\gamma_i|\tilde{\gamma}_i) + Q(\Theta_{ij}|\tilde{\Theta}_{ij}). \tag{5.15}$$

Updating γ. First, we maximize the first term of Eq. (5.15), which can be written explicitly as,

$$Q(\gamma_i|\tilde{\gamma}_i) = \frac{\left(\tilde{\gamma}_i\right)^2}{\left(\gamma_i\right)^2} - \log\left(\gamma_i\right)^2,$$

yielding,

$$\gamma_i \leftarrow \tilde{\gamma}_i.$$

Updating Θ. Similarly, we have the explicit formula of the second term of the Q-function,

$$Q(\Theta^{ij}) = -\Pi_{ij} \log \Theta_{ij} - \mathbb{E}\left(\frac{a_{ij}}{\Theta_{ij}}\right)^{\Pi_{ij}}.$$

Thus, the updating formula for Θ,

$$\Theta_{ij} \leftarrow \tilde{\Theta}_{ij}.$$

Updating the branch structure $\mathbf{\Phi}$. Instead of using the Metropolis-Hasting algorithm as [108] does, we directly sample from the posterior distribution, which is can be explicitly written as a multinomial distribution:

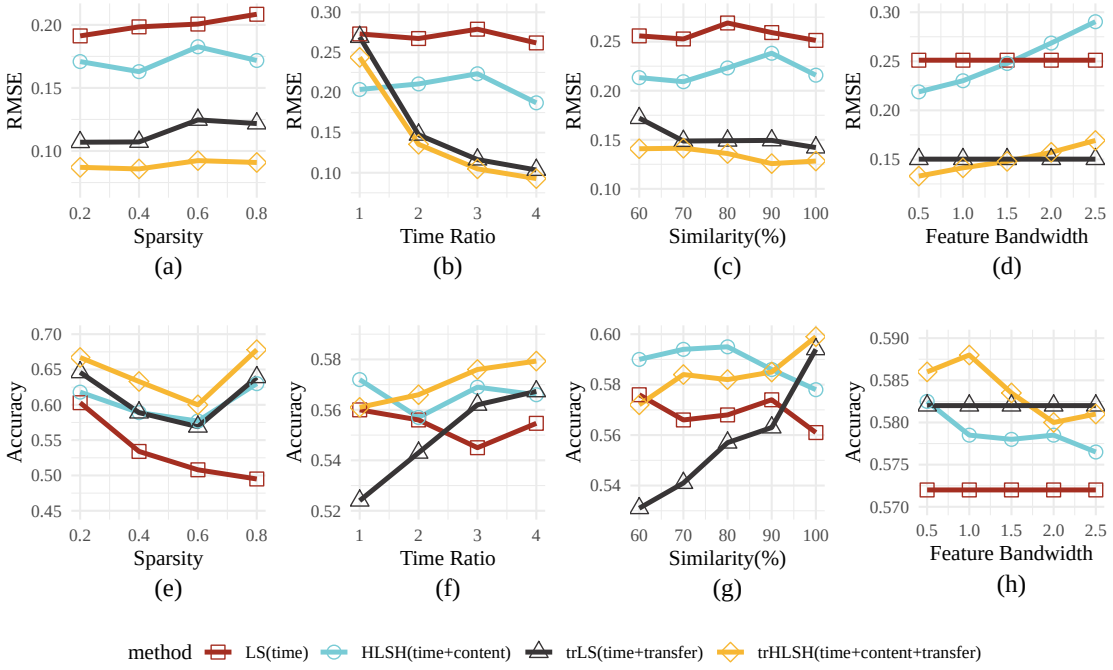


FIGURE 5.2: Performance of the parametric models on synthetic data.

$$Pr(\Phi_{t_{ik}} = t_{jk'}|\boldsymbol{\mu}, \boldsymbol{A}, \theta, \sigma, \boldsymbol{S}) = \frac{\lambda\left(t_{jk'}|\boldsymbol{\mu}, \boldsymbol{A}, \Phi\right)\tau\left(\boldsymbol{f}_{jk'}|\theta, \sigma, \Phi\right)}{\sum_{t_{jk'}}\lambda\left(t_{jk'}|\boldsymbol{\mu}, \boldsymbol{A}, \Phi\right)\tau\left(\boldsymbol{f}_{jk'}|\theta, \sigma, \Phi\right)}. \tag{5.16}$$

Note that this method for updating Z that we propose is more efficient compared with [108]. Proof is shown in Appendix A.

5.3 Experiments

In this section, we test our models on both synthetic and real-world datasets. On synthetic dataset, we aim to show how the parameters affect the model, as well as the capability of Bayesian model inferring the branching structure. Further, we evaluate our model on two tasks with real-world datasets, especially focus on accessing the ability of fitting future dynamics and predicting the arrival time of future events.

5.3.1 Synthetic Data

In this experiment, we test how the hyper-parameters affect the performance of the ability to infer the infectivity matrix $\boldsymbol{A}$, which represents the network structure.

Data Generation. We follow the method for generating synthetic data used in [56]. First, we generate an 10-dimensional Erdös-Rényi graph with `sparsity` parameter ρ as the adjacency matrix for target domain. If there is an edge, then we generate a weight from a uniform distribution. The weighted adjacency matrix is the α's (infectivity matrix) we use in Hawkes process. Before generating event cascades, the stability condition is checked [109] such that the simulation does not lead to infinite numbers of events. Then we randomly add or remove edges in the Erdös-Rényi graph that we have generated at the first step with a small probability to form the adjacency matrix for source domain and we generate a new weight for each newly-added edge. Next we generate base intensities μ's uniformly. We adopt exponential kernel function and set $\beta = 1$. After all the parameters needed are prepared, we use the thinning algorithm [110] and the brunching structure to simulate two event cascades for source and target domains respectively. Note that in brunching structure, the parent event can be indicated. When an event triggers an offspring, the corresponding feature vector is generated from Gaussian distribution with the parent feature vector as the mean. For each experiment, we generate 50 different samples and apply the proposed models to obtain the estimation of infectivity parameter α's. The performance results reported are the average over all the 50 runs.

Hyper-parameters. We summarize the hyper-parameters that we investigate in the experiments as below:

- `sparsity`: the parameter used in the Erdös-Rényi graph;
- `Time ratio`: the ratio between the observation windows of both source and target domains, which measures how much more information is in source domain than in target domain;
- `Similarity`: measures how many edges are the same in both adjacency matrices of source and target domains.
- `Feature bandwidth`: the variance parameter for the Gaussian distribution, which controls how alike the feature vectors will be between generations.

The evaluation metrics we use in parameter recovery are:

- `RMSE`: the root-mean-square error of α's.
- `Accuracy`: the percentage of edges that are correctly predicted by the estimated adjacency matrix. It measures the accuracy of the prediction of adjacency matrices. The estimated adjacency matrix is obtained by bisecting the infectivity matrix with a designated threshold. That is, if α is larger than the threshold, we regard it as an edge.

A lower RMSE and a higher accuracy indicate better performance for parameter recovery.

We involve 4 models for comparison:

- `HP (time)`. The original Hawkes process. Only temporal information is used.
- `HLSH (time+feature)`. The model utilizes both temporal and feature information.
- `trLS (time+transfer)`. The `trLS` algorithm is to set $\eta_1 = 0$, $\eta_2 = 0$, $\eta_4 = 0$ in Algorithm 3. These methods apply both temporal and cross-domain information, but not feature information.
- `trHLSH (time+feature+transfer)`. The method is shown in Algorithm 3, which leverages all of temporal, cross-domain and feature information.

Experimental Results. Figure 5.2 plots the results of the proposed parametric models when varying `sparsity`, `similarity`, `time ratio` and `feature bandwidth`. In general, `trHLSH` outperforms `trLS`, `HLSH`, and `LS` is the worst. In Figure 5.2(a) and 5.2(e) show how sparsity affects performance. Figure 5.2(b) and 5.2(f) demonstrate that the more informative source domain is, the more helpful transfer will be. Figure 5.2(c) and 5.2(g) illustrate how the similarity of the networks of source and target domains affects cross-domain knowledge transfer. The more alike two domains are, such information will be more useful for transfer. Figure 5.2(d) and 5.2(h) show the influence of feature information. The more alike between generations (a smaller feature bandwidth), the more conductive feature information will be.

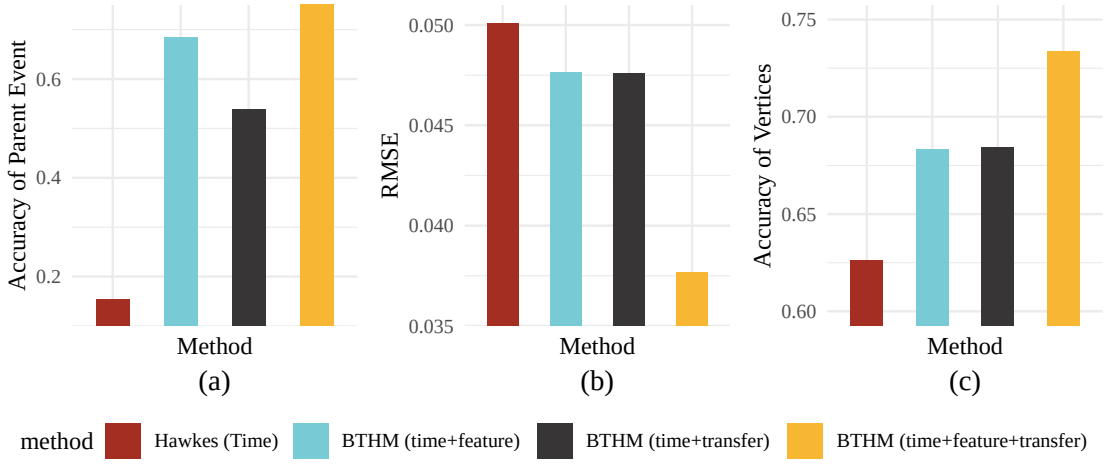


FIGURE 5.3: Performance of the BTHM on synthetic data. (`sparsity` = 0.5, `similarity` = 0.95, `time ratio` = 2 and `feature bandwidth` = 1)

Figure 5.3 presents the results for Bayesian models. Since we applied branching structure and involve the hidden variable Z, which denotes the parent event in Bayesian models, accuracy of finding "*which parent event triggers it*" is compared as shown in Figure 5.3 (a). It can be observed that both feature and cross-domain information do help for indicating parent events. In terms of recovery hidden network, BTHM outperforms the other methods.

5.3.2 The Check-in Dataset

We test our method in the check-in dataset, which consists of users' check-in activities from location-based social networks. The dataset is formed by two subsets:

- `Weeplace` (source domain) [107]: This dataset is collected from Weeplaces, a third party website that aims to visualize users' check-in activities in location-based social networks. We collect 15K event as source domain. Each event is categorized into one of the seven categories: *food*, *shop*, *art/entertainment*, *park/outdoor*, *travel*, *nightlife* and *home/work/other*.

- `Gowalla` (target domain) [111]: the dataset is collected from Gowalla, a popular location-based social network. The dataset records user's check-in history (timestamp, location). The locations are categorized into 500

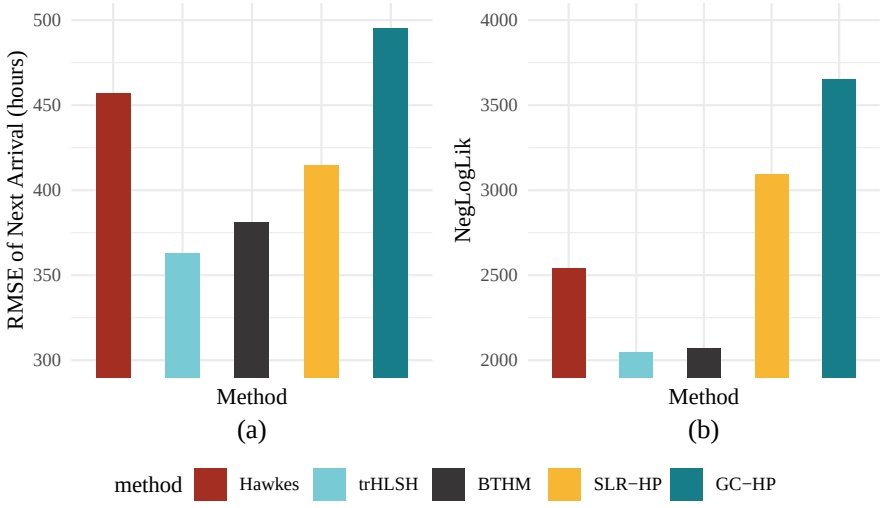


FIGURE 5.4: Performance on the check-in dataset.

subcategories. We re-classified the subcategories into the same 7 major subcategories as in the Weeplace dataset. The datasets contains 6K events.

We split the Gowalla dataset into 70% and 30% as training and test datasets, respectively. We train our model on the training dataset with `Weeplace` as the source domain. Then we evaluate the performance on the test dataset using two metrics:

- `RMSE`: the root of mean square error of prediction of the next event,

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{M}(\hat{t}^i_{N_i(T_0)+1} - t^i_{N_i(T_0)+1})^2}{n}},$$

- `NegLogLik`: the negative log-likelihood of the test dataset with the inferred parameters, which is defined by [38],

$$\text{NegLogLik} = -\sum_{i=1}^{M}\left\{\sum_{t_{ki}\in\text{TestSet}} \log\lambda_i(t_{ki}|\hat{\theta}_i) - \int_{T_0}^{T_1}\lambda_i(t|\hat{\theta})dt\right\}.$$

Here M is the dimensionality of the process, $t^i_{N_i(T_0)+1}$ is the first event in the test dataset, along with its estimation $\hat{t}^i_{N_i(T_0)+1}$. $\hat{\theta}$ is the parameter learned from the training dataset. $(0, T_0]$, $(T_0, T_1]$ are the time intervals of the training and test dataset, respectively. A lower `RMSE` or `NegLogLik` means a better performance.

Besides the 4 models we compared in the experiment on synthetic dataset, we involve two more state-of-the-art models on real-world dataset:

- `SLR-HP` [47]: the Sparse Low-Rank Hawkes Processes, where a nuclear and ℓ_1 norm of infectivity matrix is imposed to the general Hawkes process, which takes into account the prior knowledge of sparse and low-rank structure in social networks.
- `GC-HP` [27]: the Granger Causality Hawkes Processes, which replaces the commonly used exponential decay kernels with a series of Gaussian basis functions, which can flexibly capture the mutual influences.

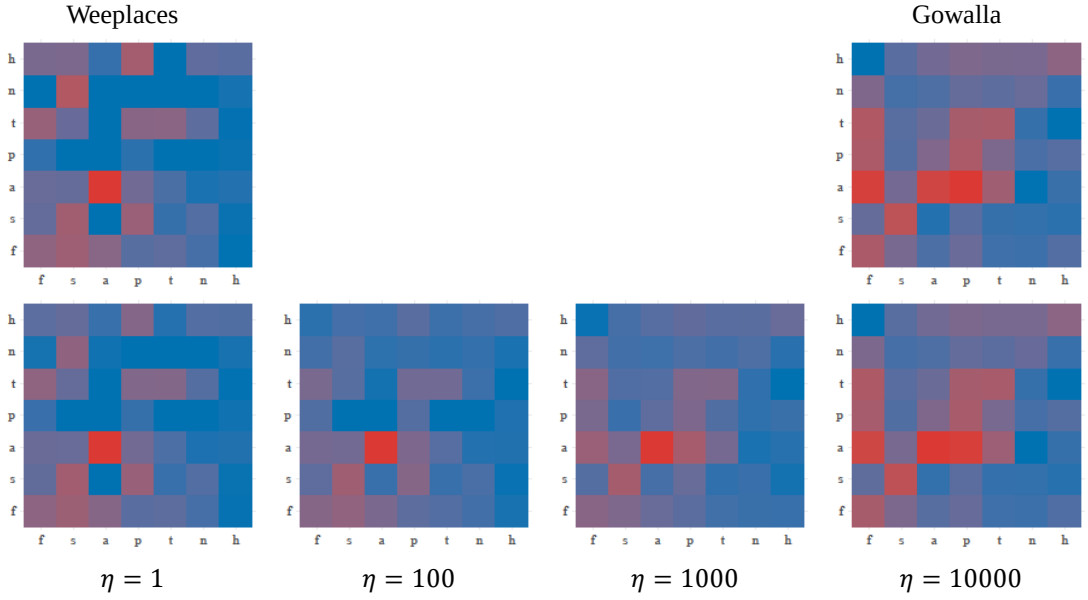


FIGURE 5.5: Illustration of the progressive transforming process of network transfer. Left-top: the network inferred on the target domain (`Weeplace`) without transfer. Right-top: the network inferred on the source domain (`Gowalla`). Bottom row: the network inferred by trHLSH with different regularization coefficients η. As η increases, the network of target domain will more resemble that of the source domain. The labels on both x- and y-axes are the initials of seven categories: **f**ood, **s**hop, **a**rt/entertainment, **p**ark/outdoor, **t**ravel, **n**ightlife and **h**ome/work/other.

Figure 5.4 presents the experimental results. It can be seen that `trHLSH` has the best `RMSE` and NegLogLik, while `BTHM` is slightly inferior to `trHLSH`. Both of our methods outperform the other baselines. Figure 5.5 gives an illustration of the network transfer process. It shows how the hyper parameter controls the transfer

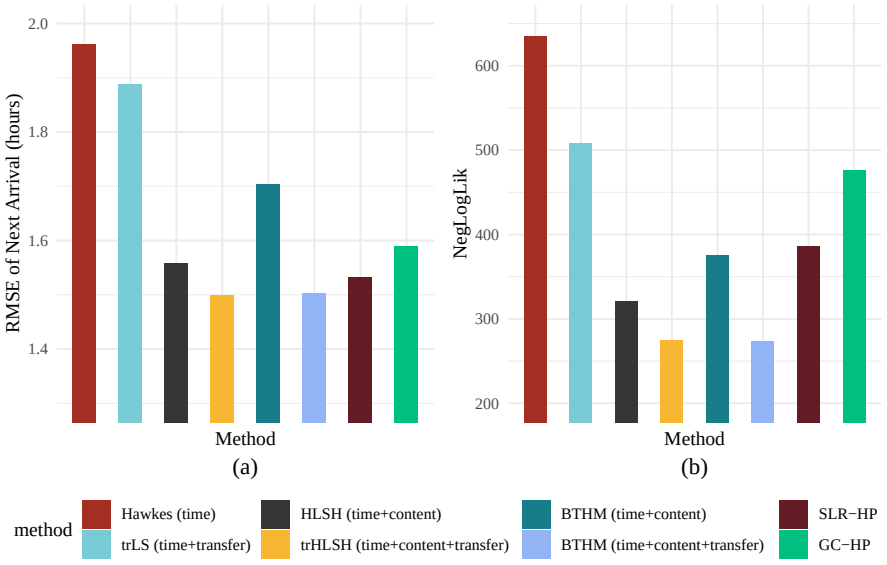


FIGURE 5.6: Performance on the SNS dataset. `trHLSH` has the best `RMSE` of next arrival and `NegLogLik`.

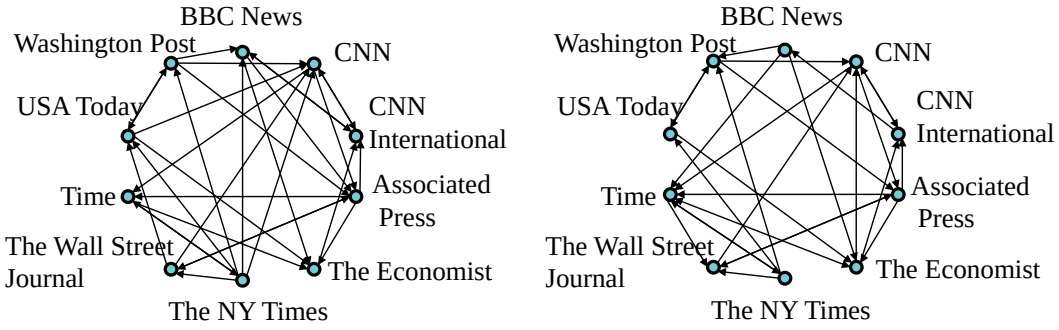
part of the `trHLSH` model. I can be seen that as the hyper parameter increases, the estimated network structure will be more like that of source domain, and vise versa.

5.3.3 The SNS Dataset

The dataset is crawled from Facebook and Twitter. We get 1482 posts from Facebook as the target domain and 1587 posts from Twitter as the source domain. The posts of both domains are from the same 10 major news agencies including CNN, BBC, Associated Press, the New York Times, the Wall Street Journal, Washington Post, etc. We divide Facebook dataset into 70% and 30% as training and testing dataset, respectively.

The textual contents are represented by bag-of-words. We extract 2000 most frequent words as features. Then we apply a simple PCA method for dimensionality reduction. Eventually we obtain the features with the first 200 principle components.

After that, we train our model and the baseline models on the training dataset, and evaluate the trained models on the test dataset, in terms of the `RMSE` of predicting next arrival time and the negative log likelihood `NegLogLik` of the test dataset. The results are shown in Figure 5.6. It can be seen that `trHLSH` outperforms the baselines in both metrics. It has the lowest `RMSE` on predicting the next arrival



(a) Facebook (b) Twitter

FIGURE 5.7: Network Structure learned from Facebook and Twitter, respectively. Some similarities can be observed.

time, and also the lowest `NegLogLik`. This result validates that feature and cross-domain information are instrumental in improving predictive performance of Hawkes processes. The network structures learned form Facebook and Twitter are presented in Figure 5.7. Some similarities can be seen between the two graphs.

5.4 Summary

In this Chapter, I present two manners of organically leveraging temporal, feature and cross-domain information: a parametric model trHLSH and a Bayesian model BTHM. The proposed models augment the basic Hawkes process by taking into account features associated with events and transferring the network structure inferred from a source domain. I present efficient learning algorithms for proposed models. The trHLSH can be viewed as a quadratic programming problem, which enjoys many computational merits. The BTHM is a mixture model, which allows flexible prior knowledge. The experimental results on both synthetic data and real-world data crawled from Facebook and Twitter suggest that our models have a better performance than the baseline models in terms of network structure recovery and prediction. One limitation is that the idea of network transfer for Hawkes processes is not applicable to neural-based Hawkes processes. It will also be interesting to explore the idea of transfer learning for neural Hawkes processes.

In the last two chapters, we see Hawkes-based models for respective tasks in social networks and recommender systems. As a generic probabilistic model, Hawkes

processes can be easily adjusted to specific tasks by applying domain knowledge as priors. Albeit exact inference may not be applicable for these Bayesian models, variational methods provide an alternative solution for the model inference. A limitation of these Bayesian models is that they are relatively difficult to be scaled. In the next part with two chapters, I present two methods for accelerating the learning process.

Part II

Accelerations

Chapter 6

Graph Convolutional Hawkes Processes: a Fast Neural Hawkes Process for Learning Feature Embeddings

6.1 Motivation

Previously we see two Bayesian models for incorporating domain knowledge to Hawkes processes. In this chapter, I propose a pure convolutional neural network for the learning of Hawkes processes. To endow the probabilistic methods with better flexibility and effectiveness, some researchers [32–35] have explored the idea of incorporating marked Hawkes processes with recurrent networks (RNNs). These models, however, suffer from three limitations. First, they only focus on the marked Hawkes processes with categorical features, which are equivalent to multi-dimensional Hawkes processes, but fail to consider continuous features. Therefore, a better feature embedding method for attributed event sequences is needed. Second, the recurrent architecture of these models makes it difficult to be accelerated by parallel mechanisms as convolutional networks (CNNs) do [36, 37]. Third, the

This chapter was published as Li, T., Luo, T., Ke, Y., & Pan, S. (2021). Mitigating Performance Saturation in Neural Marked PointProcesses: Architectures and Loss Functions. Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD, 2021 [3].

TABLE 6.1: Summary of some neural-based Hawkes processes.

Model	Ours	RMTPP [32]	IRNN [33]	NHPP [34]	FulNN [35]	GeoHP [69]	Transformer HP [113]
Marked point process	✓	✓	✓	✓	✗	✗	✓
Nonlinear intensity	✓	✓	✓	✓	✓	✗	✓
Unassigned form	✓	✗	✗	✗	✗	✗	✗
Continuous features	✓	✗	✗	✗	✗	✗	✗
GCN layers	✓	✗	✗	✗	✗	✓	✗
Non-recurrent architecture	✓	✗	✗	✗	✗	✗	✓

aforementioned models often assign particular forms of intensity function. For example, the RMTPP model [32] utilizes an exponential form whereas NHPP [34] utilizes a sigmoid function. These assumptions restrict the representation power of neural networks. In Table 6.1, I summarize some important models in terms of six aspects: is the intensify function nonlinear, do they belong to the category of marked point processes, do they assign a specific form of intensity, are they able to deal with continuous features, and do they contain GCN layers or recurrent units such as LSTM [112].

In this chapter, we propose a novel neural-based Hawkes process called Graph Convolutional Hawkes Process (GCHP) for learning attributed event sequences. Our model retains the intrinsic convolutional nature of the intensity function of Hawkes processes, and learns a trainable neural-like nonlinear intensity of a special type of processes—the nonlinear marked Hawkes processes with multiplicative kernels. The model provides a general framework for feature embedding in attributed event sequences, which can deal with both categorical and continuous features. The non-recurrent architecture of our model leverages the efficiency advantage of GCNs and has better flexibility and scalability. More specifically, the procedures are as follows: we first transform the input sequences into a set of temporal similarity graphs and feature matrices, which are then passed to graph convolutional layers. In this way, the time and the feature information of events can be naturally and seamlessly combined through graph convolutional networks (GCNs). We also utilize a moment matching mechanism, which eschews the problem of choosing proper intensity and form makes direct prediction more efficient. Meanwhile, a theoretic bound for the approximation quality is presented.

The **contributions** of this chapter are summarized as below.

- **Pure convolutional networks for Hawkes processes.** Our model links up Hawkes processes with GCNs, which provides an alternative perspective of learning a Hawkes process with neural networks. To the best of our knowledge, this is the first attempt to learn point processes with only convolutional layers and without recurrent architectures.
- **Better incorporation of features.** The GCHP model provides an elegant method for dealing with features without assuming any probability structure. Existing neural-based models [32–34] usually deal with categorical event types, but fail to consider continuous features.
- **Efficient prediction with the moment matching mechanism.** RNN-based Hawkes processes [32, 34] often pre-define a specific form of the intensity, which do not have analytic solutions and need to be solved by the time-consuming numerical method. By adopting the moment matching mechanism, our model can do prediction directly from the network, which is more efficient.
- **Theoretic development of GCHP.** We derive and prove the theoretic error bound for the approximation performed in GCHP with respect to the true process. We also show that GCHP belongs to a general class of nonlinear marked Hawkes processes with multiplicative kernels.
- **Better scalability and performance.** The experimental results show that our model runs faster and has better performance on the tasks of prediction and Granger causality inference. Compared with state-of-the-art baselines, our model achieves an average of 13% increase in the accuracy of predicting next event type on real-world datasets, with much less time.

6.2 Graph Convolutional Hawkes Processes

In this section, we introduce each component of our model. We first introduce the mathematical intuition of our model, followed by technical details. In the end, we illustrate a theoretic error bound for the approximation performed in GCHP with respect to the true process.

Graph convolutional networks. In recent years, GCNs [114, 115] have obtained great success as an efficient and effective model for graph-structured data. Given an input graph with an adjacency matrix $\mathcal{A}$ and a feature matrix $\boldsymbol{X}$, GCNs encode

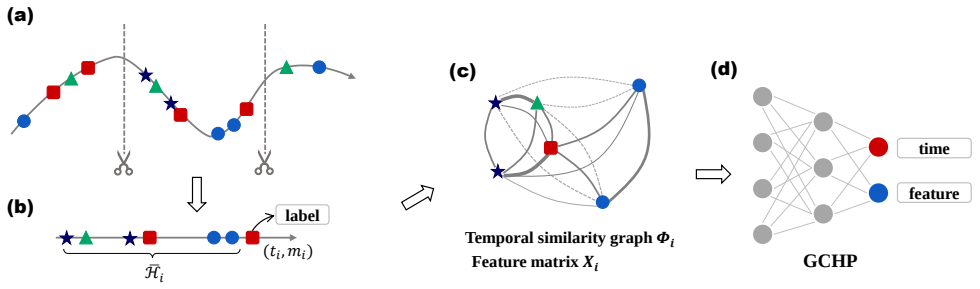


FIGURE 6.1: An illustration of the modeling flow of graph convolutional Hawkes processes (GCHP). (a)→(b): scan the attributed event sequence and obtain the trimmed history for each event. (b)→(c): transform into the attributed graph (Φ_i, X_i). (c)→(d): input the data into the GCHP model.

both the topological information and the node attributes and produce an output with node embeddings. The most representative model applies the new layer-wise propagation rule [114]:

$$H^{(l+1)} = \sigma(\tilde{A}H^{(l)}W^{(l)}),$$

where $\tilde{A}$ is a normalized adjacency matrix, $H^{(l)}$ is the output of the l-th layer, $W^{(l)}$ is a layer-specific trainable weight matrix, and σ is a non-linear activation function.

Nonliear marked Hawkes process with multiplicative kernels. We define a special type of marked Hawkes processes that incorporates multiplicative kernels for time and marks, whose intensity can be written as

$$\lambda(t, m) = \mu p(m) + \int_0^t \int_{\mathcal{M}} (\phi\kappa) * dN, \tag{6.1}$$

where μ is the base intensity and p is a deterministic density function w.r.t. the mark m. ϕ and κ are two positive definite kernel functions for arrival time and marks. $*$ denotes the convolution operation. It can be seen that such intensity is a linear convolution function. To relax the assumption of the linearity of intensity function, Eq. (6.1) can be extended to nonlinearity:

$$\lambda(t, m) = h\left(\mu p(m) + \int_0^t \int_{\mathcal{M}} (\phi\kappa) * dN\right), \tag{6.2}$$

where $h : \mathbb{R} \to \mathbb{R}^+$ is a non-negative function. It can be verified that the likelihood of such a process can be viewed as a function of matrices Φ and $\mathcal{K}$. The former matrix is composed of $\phi(t_i - t_j)$'s. We call it the *temporal similarity graph*, as it measures the similarity between every two events. It can be seen that the feature

kernel $\mathcal{K}$ provides an embedding method for the marks, in accordance with the theory of reproducing kernel Hilbert space. Therefore, the estimation of the next interarrival time τ and mark m, which is calculated from the estimated parameters by maximizing the likelihood, can be viewed as a function (denoted by g) of Φ and $\mathcal{K}$. The estimation of the next interarrival time $\hat{\tau}$ and mark $\widehat{m}$ can be written by

$$\hat{\tau}, \quad \widehat{m} = g(\Phi \odot \mathcal{K}),$$

where Φ is the temporal similarity graph, $\mathcal{K}$ the Gram matrix of the features (marks) and $\odot$ denotes the Hadamard product. The model assumes that the next event is determined by the similarity of features and time. The closer two events are, the more similar their corresponding features will be.

In this chapter, we propose a model called Graph convolutional Hawkes process (GCHP), which learns the function g as a trainable neural-like function. Instead of specifying a particular kernel for the features, we utilize GCN layers to learn the embedding of features. The target of the model is to learn the expected interarrival time and the feature of the i-th event (t_i, m_i) from the set of historical events $\mathcal{H}_i = \{(t_j, m_j) : 0 < j < i\}$:

$$\hat{\tau}_i, \quad \widehat{m}_i = g(\mathcal{H}_i),$$

where $\hat{\tau}_i$ and $\widehat{m}_i$ are the estimation of the interarrival time $\tau_i = t_i - t_{i-1}$ and the associated feature m_i, respectively.

The model. Figure 6.1 illustrates the modeling process of our GCHP. We first scan the input attributed event sequence. For each event (t_i, m_i), we obtain its trimmed history $\bar{\mathcal{H}}_i$ with a preset number of prior events. We then transform the trimmed history $\bar{\mathcal{H}}_i$ into a temporal similarity graph Φ_i and a feature matrix $\boldsymbol{X}_i$, which are then passed to graph convolutional layers. Unlike [32, 34] that assume a specific form of the intensity function, we use a moment matching strategy to approximate the intensity. To be specific, our GCHP model with two graph convolutional layers can be written as

$$\begin{aligned}
\hat{\tau}_i, \quad \widehat{m}_i &= F([H_i^{(2)} : \tilde{\Phi}_i]), \\
H_i^{(2)} &= \sigma(\tilde{\Phi}_i H_i^{(1)} W^{(1)}), \\
H_i^{(1)} &= \sigma(\tilde{\Phi}_i \boldsymbol{X_i} W^{(0)}),
\end{aligned}$$

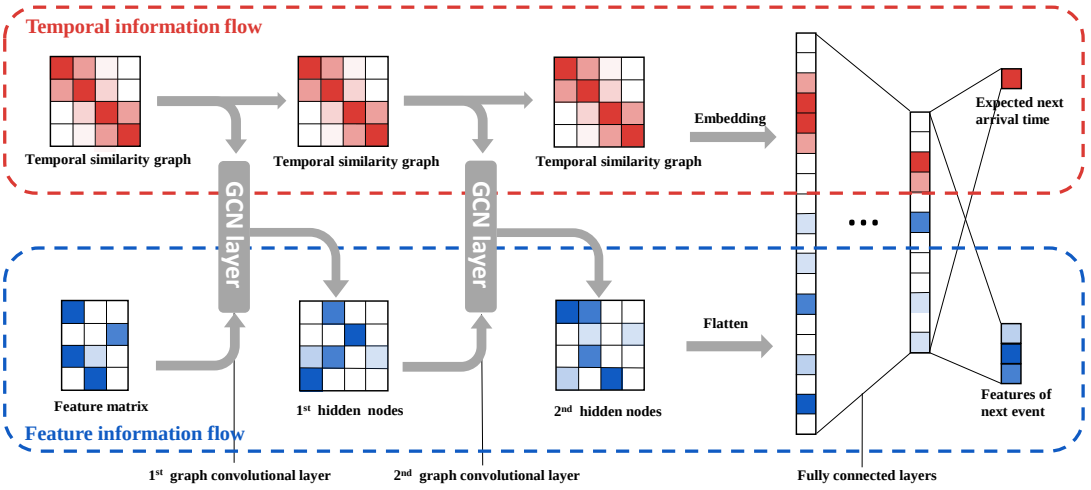


FIGURE 6.2: The network architecture of the GCHP. The feature matrices interact with the temporal information embodied in the temporal similarity graphs through the GCN layers.

where $\tilde{\Phi}_i = D_i^{-1/2}\Phi_i D_i^{-1/2}$, and D_i is a diagonal matrix of the degrees of Φ_i. ":" denotes the concatenation of the two matrices. F denotes fully connected layers, and σ is an activation function, such as the ReLU.

The temporal similarity graph Φ. The temporal similarity graph Φ plays a crucial role in our model. As suggested by its name, it measures the similarity between events in the time domain. The use of temporal similarity graph is not arbitrary. It is essentially an important component in the intensity function of the nonlinear marked Hawkes processes with multiplicative kernels.

Given a symmetric kernel ϕ, the weight between two events (t_i, m_i) and (t_j, m_j) can be defined by $\phi(t_i - t_j)$. The dimension of Φ is determined by the length of the trimmed history, which is preset by fixing the influential range, i.e., the number of past events that the next event is relevant to. Presetting the range makes the temporal similarity graphs and feature matrices of different events aligned. Trimmed history is also considered a reasonable approximation of the full history as the major influence comes from the closest events due to the decay of influence.

Self-supervised learning of GCHP. Generally, learning a stochastic process by maximizing the log-likelihood is viewed as an unsupervised task. In this chapter, we borrow the idea from task of predicting the next word given the previous sequence [116, 117] in natural language processing, and learn GCHP in the manner of self-supervised learning. That is, as shown in Figure 6.1(b), we treat each event (t_i, m_i)

in the input sequence as the label and compute the loss from the events right before it. The objective function can be rewritten as

$$\text{loss} = \sum_{i=1}^{N} \ell_m(m_i, \widehat{m}_i) + \theta \ell_t(\tau_i, \hat{\tau}_i).$$

Here $\widehat{m}_i$ and $\hat{\tau}_i$ are the outputs for the feature and interarrival time of the i-th event, given the trimmed history $\bar{\mathcal{H}}_i$. θ is a hyper-parameter controlling the weight of time. ℓ_m and ℓ_t are the respective loss functions. It is worth noting that, the maximum likelihood estimator of Hawkes processes also admits the form, as a result of the invariance property.

Moment Matching. A challenging part in modeling marked Hawkes processes using deep learning techniques is that the exact form of intensity $\lambda(t, m)$ is not known. There are several attempts made by researchers to designate some specific form for the intensity. [32] proposes an exponential form, whereas [34] uses sigmoid. Such choices, however, may restrict the expressive power of neural networks. Moreover, the calculation of the expectation of the next interarrival time usually does not have analytic solutions, and thus one has to turn to numerical methods, such as Monte Carlo simulation, which is computationally unfriendly. To address this issue, we propose to use a moment matching mechanism. We directly output the expectation of the next interarrival time $\hat{\tau}_i = \mathbb{E}(\tau_i|\bar{\mathcal{H}}_i)$ and feature $\hat{m}_i = \mathbb{E}(m_i|\bar{\mathcal{H}}_i)$. In fact, this is equivalent to adopting a piecewise constant intensity function—when an event occurs, the arrival time of the next event has an exponential distribution, whose mean is determined by the history.

Remark. It is worth noting that GCHP is still a well-defined point process, where the stochasticity is inherited from the background Poisson process. The intensity of GCHP is piecewisely determined by $\hat{\lambda}_i = 1/\hat{\tau}_i = 1/g(\mathcal{H}_i)$ between arrivals, and therefore GCHP is adapted to the filtration $\boldsymbol{\mathcal{H}} = \{\mathcal{H}_t : 0 \leqslant t \leqslant \infty\}$. The neural network learns g in this case.

Theorem 6.1 shows that the difference between the original process and our approximation can be bounded by $\mathcal{O}(e^{-t})$, which shows the error decays exponentially as the length of observation time increases.

Theorem 6.1 (Moment Matching)**.** Let $\lambda_g(t|\mathcal{H}_{t-})$ be the ground intensity function of a point process N. Let $\breve{N}$ be an adjoint process such that (1) $\breve{N}$ has piecewise constant intensity function; (2) the intensity function is continuous between

events; (3) its ground intensity function $\breve{\lambda}_g(t|\mathcal{H}_{t-})$ satisfying for any t, it has $\int_0^{T_{\breve{N}_g(t)}} \breve{\lambda}_g(t|\mathcal{H}_{t-})dt = \int_0^{T_{N_g(t)}} \lambda_g(t|\mathcal{H}_{t-})dt$, where $T_{N_g(t)}$ is the arrive time of the $N_g(t)$-th event. If λ_g is bounded by b, then

$$\mathbb{P}\left\{\sup_{\alpha\in[0,1]} \left\|\frac{1}{t}\int_{\mathcal{M}} \left(N(\alpha t, m) - \breve{N}(\alpha t, m)\right) dm\right\| \geqslant \epsilon\right\} \leqslant 2\exp\left\{-cb\min\left(\epsilon^2 t, \epsilon\right)\right\},$$

where c is a constant.

Proof Scratch. The theorem can be obtained by using the relationship of N and the arrival time. The bounded intensities yields sub-exponentially distributed interarrival time. Applying Berstein's inequality, the final risk bound can be obtained.

Complexity. Our model has a running time complexity of $\mathcal{O}(Nm^2p)$ for each epoch, where N is the number of events, m and p are the length of the trimmed history and the dimension of features, respectively. Note that $m \ll N$. This complexity is superior to [27, 30, 47] whose complexity is $\mathcal{O}(N^3p)$. It is also better than [60], which has a complexity of $\mathcal{O}(Np^2)$ for high-dimensional Hawkes processes where $p \gg m > 0$.

6.3 Experiments

In this section, we evaluate our model against some state-of-the-art baselines. We test our model on two synthetic and four real-world datasets for two tasks: prediction and Granger causality inference.

Datasets. The datasets we use are listed as follows. We summarize the statistics of the datasets in Table 6.2, 6.3.

– *Hawkes*: a synthetic dataset with categorical features. The event sequences are generated from a 10-dimensional Hawkes process with uniformly sampled parameters.
– *GMHP* [68]: a synthetic dataset with continuous features. The Gaussian marked Hawkes process is a hierarchical Bayesian model. Its infectivity parameters are Gaussian distributed with the mean parameter being a linear combination of the features.

– *IPTV* [20]: a real-world dataset with categorical features. The dataset consists of IPTV viewing events with timestamps and categories.
– *Weeplace* [107]: a real-world dataset with both categorical and continuous features. The dataset contains the check-in histories of users at different locations (longitudes and latitudes).
– *ATM* [33]: a real-world dataset with categorical features. The dataset is composed of the event logs of error reporting and failure tickets.
– *Auction* [118]: a real-world dataset with continuous features. The dataset contains eBay auction information on Cartier wristwatches, Xbox game consoles, etc.

TABLE 6.2: Statistics of datasets: categorical features

Dataset	**No. of events**		**No. of sequences**		**No. of event types K**
	Training	Testing	Training	Testing	
Hawkes	36k	7k	100	40	10
ATM	370k	182k	1085	469	7
Weeplace	98k	31k	21	8	8
IPTV	731k	243k	227	75	16

TABLE 6.3: Statistics of datasets: continuous features

Dataset	**No. of events**		**No. of sequences**		**Feature dimension**
	Train set	Test set	Train set	Test set	
GMHP	49k	21k	100	80	4
Auction	7K	2.5k	317	108	1
Weeplace	98k	31k	21	8	2

Experimental environment. All the experiments were conducted on a server with 64G RAM, a 16 logical cores CPU (AMD Ryzen Threadripper 1900X) and 4 GPUs (Nvidia GeForce GTX 1080 Ti) for acceleration.

6.3.1 Prediction

The first task is to predict the next event in terms of the interarrival time and features.

Baselines. We compare our model with five state-of-the-art neural-based methods: RMTPP [32], IRNN [33], NHPP [34], MAHP [66] and GeoHP [69].

Metrics. We assess the performance of each model in three aspects: time prediction, feature prediction, and training time. We use **RMSE** for time prediction and continuous features. For categorical features, we measure by the percentage of correct predictions (**Accuracy**). A higher accuracy and a lower RMSE indicate a better performance. Running time is also recorded.

Experimental setting. The hyper-parameters of all models were tuned for the best performance. The memory sizes for GCHP and MAHP are set five and eight. We use a single fully connected layer after the graph convolutional layers in our model. To process continuous features in the models, we replace the original cross entropy loss for event type with the MSE loss.

Discussion. We present the experimental results in Table 6.4 and 6.5. Full results can be found in the Appendix. In the case of categorical features, it can be seen that our model achieves the best accuracy and RMSE on almost all datasets. Compared with the second best model, the average improvement for the prediction accuracy on real-world datasets is 13%, and 18% for RMSE. Apart from the superior prediction quality, our model only uses 11% of the training time compared to the second best. Note that NHPP runs relatively slow, as it involves an MCMC step at each epoch. On the datasets with the continuous features, the GCHP also achieves similar performance, and analogous conclusions can be drawn on datasets with continuous features. These results show that the GCHP model is both effective and efficient in terms of prediction.

6.3.2 Granger Causality Inference

In this task, we perform experiments for inferring the *Granger causality graph* [27, 54, 60], which plays an important role in the study of attributed event sequences.

6.3.2.1 Synthetic Dataset

This experiment is designed to assess the capability of each model in recognizing the pattern in the Granger causality graph. As defined in [27], the Granger causality

graph of a multi-dimensional Hawkes process can be represented by its infectivity matrix $\mathcal{A}$. The (i, j) entry α_{ij} of $\mathcal{A}$ measures the probability that an event of type j will trigger a subsequent event of type i. Similarly, we define the Granger causality graph for GCHP. We input an event sequence of which the last event type is i, while other events are zero-padded, to the trained model. The output, which represents the distribution of the next event type, constitutes the i-th row of the Granger causality graph for GCHP.

Data generation. We first generate four datasets, each with 50 event sequences generated from a 10-dimensional Hawkes process. Each realization uses an exponential kernel and sets the base intensity $\mu = 0.02$. For different datasets, a different pattern is embedded in the infectivity matrix $\mathcal{A}$ of the corresponding process, which is shown in the first row of Figure 6.3.

Baselines. We compare our GCHP with three models: **ADM4** [47]. Granger causality for Hawkes processes (**GC**) [27], the non-parametric Hawkes cumulant method (**NPHC**) [60], and Transformer HP (**THP**) [113]. For **GC**, we use five Gaussian kernels, and the causality graph is obtained by averaging the amplitudes.

Discussion. Figure 6.3 shows the inferred causality graphs by the tested models. It can be seen that our model succeeds in recognizing the patterns of Granger causality graphs. ADM4 also finds the correct patterns. GC partially recognizes the patterns; while NHPC fails on three datasets. The possible reason is that ADM4 has the same assumptions as how the synthetic dataset is generated, except for the regularizer. The GC model adopts Gaussian decay kernels, which is able to learn the exponential decays, but not perfectly. We animate the training process in the supplementary material.

6.3.2.2 Real-world Datasets

We also perform an experiment on the real-world datasets. Since the true Granger causality graph is not available for the real-world datasets, we define the *empirical causality graph* $\mathcal{G}$ as the ground truth. The (i, j) entry of an empirical causality graph, given order k, is defined by

$$g_{ij}(k) = \frac{\sum_l \#\{\text{event of type } j \in \mathcal{H}_{il}^k\}}{\sum_l \#\{\text{event} \in \mathcal{H}_{il}^k\}}, \tag{6.3}$$

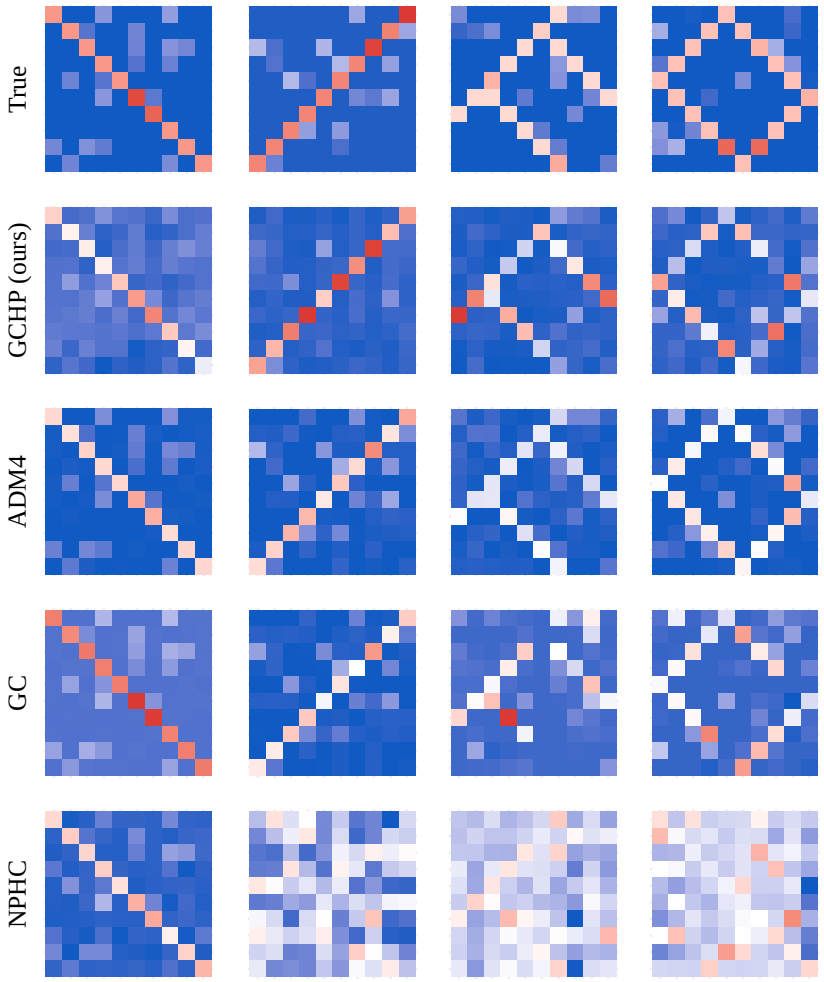


FIGURE 6.3: Comparasion of the true infectivity matrix $\mathcal{A}$ and the inferred Granger causality graph by various methods. Each column uses the same dataset that is generated from a 10-dimensional Hawkes process, whose infectivity matrix is shown in the top row.

where $\mathcal{H}_{il}^k$ is the history of size k (i.e., k events prior) for the l-th event of type i in the sequence.

Metrics. We use quality metrics including: running time (s), the mean Bhattacharyya distance (**BhatDist**), and the mean Kendall rank correlation (**KenRank**). The BhatDist and KenRank are defined by

$$\text{BhatDist}(\mathcal{A}, \mathcal{G}) = -\frac{1}{M}\sum_{i=1}^{M} \ln\left(\sum_{j=1}^{M}\sqrt{a_{ij}g_{ij}}\right),$$

$$\text{KenRank}(\mathcal{A}, \mathcal{G}) = \frac{1}{M}\sum_{j=1}^{M} \text{KenCorr}\left(a_{\cdot j}, g_{\cdot j}\right),$$

where a_{ij} and g_{ij} are the (i, j) entry of matrices $\mathcal{A}$ and $\mathcal{G}$, respectively. Note that a smaller BhatDist and a larger KenRank indicate that the estimation of the causality graph is closer to the ground truth, i.e., the empirical causality graph $\mathcal{G}$.

Discussion. We first obtain the empirical causality graph of the real-world datasets as defined in Eq. (6.3) with orders 1, 2 and 5. Then we calculate BhatDist and KenRank between the inferred causality graphs and empirical causality graphs. The results are shown in Table 6.6. It can be seen that our model achieves the best performance on the real-world datasets. These results demonstrate that our model is better at finding causality patterns from event sequences with much less time, especially for the short-term causality.

6.4 Summary

In this chapter, I propose a novel model called Graph Convolutional Hawkes Processes for modeling attributed event sequences. The proposed GCHP complies with the convolutional structure existing in marked Hawkes processes. We establish some theoretic results of GCHP, including its approximation bound with respect to the true process, and its relationship with nonlinear marked Hawkes processes with multiplicative kernels. Extensive experiments show that our model achieves significant improvements over the state-of-the-art baselines in terms of the prediction and the inference of the Granger causality graph. Our model also shows great efficiency with an average of 9 times speed-up over baselines. This chapter shows the potential of convolutional networks for modeling event sequences, albeit it seems counter-intuitive as RNNs match the nature of a sequence. In the next Chapter, we will see a generic downsampling method for accelerating the learning of Point processes including but not limited to Hawkes processes.

TABLE 6.4: Performance of prediction: categorical features.

Dataset	Model	Accuracy (feature)	RMSE (time)	Running time (s)
Hawkes	**Ours**	**33.67%**	**4.385**	**0.101**
	RMTPP	32.46%	5.565	0.451
	IRNN	33.40%	4.395	0.475
	NHPP	33.61%	4.480	46.47
	MAHP	10.01%	4.898	1.794
	GeoHP	22.91%	12.62	38.94
	Transformer	33.27%	35.01	122.7
ATM	**Ours**	**91.26%**	**2.612**	**0.356**
	RMTPP	76.64%	7.150	5.756
	IRNN	76.19%	2.793	6.299
	NHPP	33.78%	7.558	660.52
	MAHP	41.91%	3.202	24.876
	GeoHP	14.91%	9.268	872.40
	Transformer	68.76%	4.534	877.05
Weeplace	**Ours**	**31.81%**	6.493	**0.116**
	RMTPP	22.07%	7.162	1.400
	IRNN	23.37%	**6.448**	1.434
	NHPP	25.71%	6.773	140.26
	MAHP	15.13%	6.969	5.210
	GeoHP	17.74%	28.28	42.89
	Transformer	29.24%	51.78	51.15
IPTV	**Ours**	**75.35%**	**10.866**	**1.124**
	RMTPP	57.57%	34.382	11.281
	IRNN	58.63%	34.311	11.065
	NHPP	31.05%	19.929	1070.15
	MAHP	18.02%	36.738	28.213
	GeoHP	43.12%	25.421	907.91
	Transformer	71.94%	31.325	424.37

TABLE 6.5: Performance of prediction: continuous features.

Dataset	Model	RMSE (feature)	RMSE (time)	Running time (s)
GMHP	**Ours**	**1.855**	**4.340**	**0.208**
	RMTPP	2.338	5.641	1.116
	IRNN	2.498	4.349	1.375
Auction	**Ours**	**24.52**	**0.430**	**0.011**
	RMTPP	30.90	1.094	0.127
	IRNN	29.69	0.434	0.090
Weeplace	**Ours**	**0.211**	6.502	**0.085**
	RMTPP	0.537	7.070	1.577
	IRNN	0.528	**6.478**	1.435

TABLE 6.6: Performance of Granger causality inference.

Dataset (k)	**Metric**	**GCHP** (ours)	**ADM4**	**GC**	**NPHC**
ATM-1	BhatDist	**1.193**	1.550	2.216	1.544
	KenRank	0.314	0.136	0.108	**0.383**
ATM-2	BhatDist	**1.023**	1.318	2.021	1.301
	KenRank	**0.306**	0.183	0.156	0.211
ATM-5	BhatDist	**1.067**	1.156	1.783	1.141
	KenRank	**0.265**	0.170	0.143	0.170
Weeplace-1	BhatDist	**0.171**	0.521	0.652	0.960
	KenRank	**0.339**	0.250	0.312	0.330
Weeplace-2	BhatDist	**0.171**	0.498	0.620	0.933
	KenRank	**0.285**	0.214	0.258	0.276
Weeplace-5	BhatDist	**0.172**	0.472	0.580	0.883
	KenRank	0.223	0.151	0.214	**0.267**
IPTV-1	BhatDist	**0.302**	5.567	6.914	8.547
	KenRank	**0.217**	-0.026	0.189	0.144
IPTV-2	BhatDist	**0.303**	5.156	6.395	8.058
	KenRank	**0.199**	-0.043	0.181	0.159
IPTV-5	BhatDist	**0.399**	4.495	5.555	7.213
	KenRank	0.178	-0.071	0.156	**0.184**

TABLE 6.7: Running time of Granger causality inference.

Dataset	**GCHP** (ours)	**ADM4**	**GC**	**NPHC**
ATM	**20.98**	54.65	403.50	183.14
Weeplace	**2.20**	99.39	14.34	3.19
IPTV	**40.36**	375.21	80.89	52.49

Chapter 7

Accelerating the Learning Process via Thinning

7.1 Motivation

Despite the popularity of Hawkes processes and other point processes, related applications are often plagued by the scalability issue. Some state-of-the-art models [27, 29, 30] have a time complexity of $\mathcal{O}(d^2n^3)$, where n is the number of events and d is the dimension. As the number of events increases, learning such a model would be very time consuming, if not infeasible. This becomes a major obstacle when applying point processes.

A simple strategy to address this problem is to use part of the dataset in the learning. For instance, in mini-batch gradient descent, the gradient is computed at each iteration using a small batch instead of full data. For point processes, however, to find a suitable sampling method is not a easy task. This is due to the special input data — event sequences. First of all, event sequences are *posets*. An inappropriate sampling method may spoil the order structure of the temporal information. This is especially harmful when the intensity function depends on its history. Second, many models built upon point processes utilize the arrival intervals between two events. Such models are particularly useful as they take into account

The work in this chapter has been published as Li, T., & Ke, Y. (2019). Thinning for accelerating the learning of point processes. Advances in Neural Information Processing Systems, NeurIPS. Vancouver, Canada. [4]

the interactions between events or nodes. Examples include Hawkes processes and their variants [27, 29, 45]. An improper sampling method may change the lengths of arrival intervals, leading to a poor estimation of model parameters.

A commonly-used approach to the sampling of point processes is *sub-interval sampling* [64, 66]. Sub-interval sampling is a piecewise sampling method, which splits an event sequence into small pieces and learns the model on these sub-intervals. At each iteration, one or several sub-intervals are selected to compute the gradient. This method, however, has an intrinsic limitation: it cannot capture the panoramic view of a point process. Take self-excited event sequences as an example. An important characteristic of such sequences is that events are not evenly distributed across the time axis, but tend to be clumping in a short period of time. Sub-interval sampling, in this circumstance, is like "a blind man appraising an elephant" — it can only see part of the information at each iteration, prone to a large variance of the gradient.

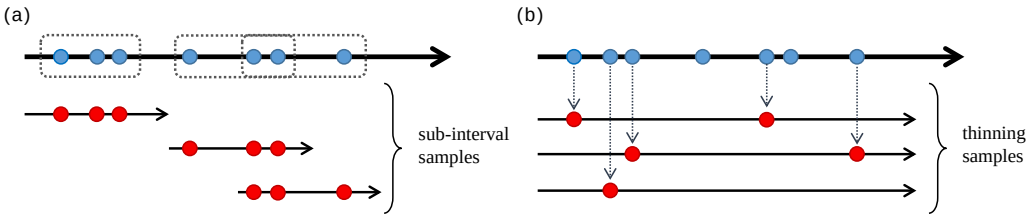


FIGURE 7.1: Comparison of sub-interval sampling (a) and thinning sampling (b).

In this chapter, we discuss "thinning sampling" as a downsampling method for accelerating the learning of point processes. A comparison between sub-interval and thinning are shown in Figure 7.1. Conventionally, thinning is a classic technique for simulating point processes [119]. We borrow the idea and adopt it as a downsampling method for fast learning of point processes. It is convenient to implement and able to capture the entire landscape over the observation timeline.

The main contributions of this chapter are summarized as follows.

- To the best of our knowledge, we are the first to employ thinning as a downsampling method to accelerate the learning of point processes.

- We present theoretical results for intensity, parameter and gradient estimation on the thinned history of a point process with decouplable intensities. We also apply thinning in stochastic optimization for the learning of point processes.
- Experiments verify that thinning sampling can significantly reduce the model learning time without much loss of accuracy, and achieve the best performance when training a Hawkes process on both synthetic and real-world datasets.

7.2 Point Processes

A point process $N(t)$ can be viewed as a random measure from a probability space $(\Omega, \mathfrak{F}, P)$ onto a simple point measure space $(\mathcal{N}, \mathfrak{B}_{\mathcal{N}})$. We define a point process $N(t)$ as follows.

Definition 7.1 (Point process)**.** Let t_i be the i-th arrival time of a point process $N(t)$ defined by, $t_i = \inf_t\{N(t) \geq i\}$. A point process $N(t)$ on $\mathbb{R}^+$ is defined by $N(t) = \sum_i \delta_{t_i}(t)$, where δ_ω is the Dirac measure at ω.

The "information" available at time t is represented by a sub-σ-algebra $\mathcal{H}_t = \sigma(N(t) : t \in \mathbb{R}^+)$. The filtration $\boldsymbol{\mathcal{H}} = (\mathcal{H}_t)_{0 \leq t < \infty}$ is called the *internal history*. A point process can be characterized by its *intensity function*. It measures the probability that a point will arrive in an infinitesimal period of time given the history up to the current time. Herein we follow the definition of stochastic intensity introduced in [120, 121].

Definition 7.2 (Stochastic intensity)**.** Let $N(t)$ be an $\boldsymbol{\mathcal{H}}$-adapted point process and $\lambda(t)$ be a nonnegative $\boldsymbol{\mathcal{H}}$-predictable process. $\lambda(t)$ is called the $\boldsymbol{\mathcal{H}}$-intensity of $N(t)$, if for any nonegative $\boldsymbol{\mathcal{H}}$-predictable process $C(t)$, the equality below holds,

$$\mathbb{E}\left[\int_0^\infty C(s)dN(s)\right] = \mathbb{E}\left[\int_0^\infty C(s)\lambda(s)ds\right]. \tag{7.1}$$

The expectation of $N(t)$ is called the $\boldsymbol{\mathcal{H}}$-*compensator*, which is the cumulative intensity $\Lambda(t) = \int_0^t \lambda(s)ds$. Doob-Meyer decomposition yields that $N(t) - \Lambda(t)$ is an $\boldsymbol{\mathcal{H}}$-adapted martingale. Another important result is that the conditional intensity function uniquely determines the probability structure of a point process [122]. Similar results can be extended to compensators [121, 123].

M-estimator. Commonly-used parameter estimation methods for point processes include maximum likelihood estimation (MLE) [27, 29, 56], intensity-based least square estimation (LSE) [124, 125] and counting-based LSE [66]. These methods fall into a wider class called *martingale estimator* (M-estimator) [43, 121]. The gradient $\nabla R(\theta)$, $\theta \in \mathbb{R}^d$ can be expressed as a stochastic integral: $N(t)$,

$$\textit{(full gradient)} \qquad \nabla R(\theta) = \int_0^T H(t;\theta)\left[dN(t) - \lambda(t;\theta)dt\right]. \tag{7.2}$$

Here $R(\theta)$ is the loss function, which may be the log-likelihood, or the sum of the squares of the residuals. $\lambda(t;\theta)$ is the intensity function to be estimated. $[0,T]$ is the observation window. $H(t;\theta)$ is a vector-valued function, or more generally, a predictable, boundedly finite, and square integrable process associated with $\lambda(t;\theta)$. Different choices of $H(t;\theta)$ instantiate different estimators: $H(t;\theta) = -\nabla \log \lambda(t;\theta)$ for MLE, $H(t;\theta) = \nabla\lambda(t;\theta)$ for intensity-based LSE, and $H(t;\theta) = 1$ for counting-based LSE. We write $\sum_{i=1} \nabla R(\theta;\omega_i), \omega_i \in \Omega$ as the empirical gradient given realizations $\{\omega_i\}$. Under the true parameter θ^*, $\nabla R(\theta^*)$ is a martingale and its expectation is 0. Gradient descent methods are often used to find $\hat{\theta}$ by solving $\sum_{i=1} \nabla R(\hat{\theta};\omega_i) = 0$. Note that in this chapter, LSE refers to intensity-based LSE, as all the results can be easily extended to counting-based LSE.

7.3 Thinned Point Processes

In this section, we introduce a derivative process called *p-thinned process*. Intuitively, the p-thinned process is obtained from a point process by retaining each point in it with probability p, and dropping with probability $1-p$. We formally define the p-thinned process as follows.

Definition 7.3 (p-thinned process)**.** The p-thinned process $N_p(t)$ associated with a point process $N(t)$ (called the ground process) is defined by summing up the Dirac measure on the product space $\Omega \times \mathcal{K}$:

$$N_p(t) = \sum_{i=1} \delta_{(t_i, B_i=1)}(t),$$

where B_i's are independent Bernoulli distributed random variables with parameter p.

Alternatively, the p-thinned process can be written in the form of a compound process as $N_p(t) = \sum_{i=1}^{N(t)} B_i$ and its differential can be expressed as $dN_p(t) = B_{N(t)}dN(t)$. In this way, the Riemann–Stieltjes integral of a stochastic process with respect to a p-thinned process can be defined by,

$$\int_0^T H(t)dN_p(t) = \int_0^T H(t)B_{N(t)}dN(t) = \sum_{i=1}^{N(T)} H(t_i|\mathcal{H}_{t_i-})B_i.$$

Two types of histories. The differential defined above implies that the intensity of thinned process can be written as $\lambda_p(t) = p\lambda(t)$. This relation between the intensities of a thinned process and its ground process is intuitively plausible. The implicit condition, however, is that $\lambda_p(t)$ must be measurable with respect to the full history of $N(t)$ and all the thinning marks B_i prior to the current time t. Such a history can be expressed by the filtration $\boldsymbol{\mathcal{F}} = (\mathcal{F}_t)$, where $\mathcal{F}_t = \mathcal{H}_t \otimes \mathcal{K}_t$ and $\mathcal{K}_t$ is the cylindrical σ-algebra of the markers. This history is called the *full history*, and its corresponding $\boldsymbol{\mathcal{F}}$-intensity is denoted by $\lambda_p^{\boldsymbol{\mathcal{F}}}(t)$. When computing $\lambda_p^{\boldsymbol{\mathcal{F}}}(t)$, we need to take into account all the points prior to t, including those dropped ones.

The other type of history, called *the thinned history*, is the internal history of the thinned process, denoted by $\boldsymbol{\mathcal{G}} = (\mathcal{G}_t)$, where $\mathcal{G}_t = \sigma(N_p(t))$. In computing its $\boldsymbol{\mathcal{G}}$-intensity $\lambda_p^{\boldsymbol{\mathcal{G}}}(t)$, we only need to consider all the retained points of the thinned process.

The following lemma describes the relationship between different intensities.

Lemma 7.4 (Relationship of intensities)**.** Let $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{G}}$ be the full history and thinned history with respect to a p-thinned process $N_p(t)$. Let $\boldsymbol{\mathcal{H}}$ be the internal history of $N(t)$. The following equalities hold:

(1) $\lambda_p^{\boldsymbol{\mathcal{F}}}(t) = p\lambda^{\boldsymbol{\mathcal{H}}}(t)$;

(2) $\lambda_p^{\boldsymbol{\mathcal{G}}}(t) = p\mathbb{E}\left[\lambda^{\boldsymbol{\mathcal{H}}}(t)|\boldsymbol{\mathcal{G}}\right]$.

Due to space limit, we defer all the proofs to the supplementary material. Lemma 7.4 tells that the intensity of a p-thinned process is a version of the conditional expectation $\mathbb{E}\left[\lambda^{\boldsymbol{\mathcal{H}}}(t)|\boldsymbol{\mathcal{G}}\right]$. Provided the information of the p-thinned process, the p-thinned intensity is an orthogonal projection of that of the ground point process on $\mathcal{L}^2$. Therefore, $1/p\lambda_p^{\boldsymbol{\mathcal{G}}}(t)$ is an unbiased estimation of $\lambda^{\boldsymbol{\mathcal{H}}}(t)$.

p-thinned and sub-interval gradient. We define the *p-thinned gradient*, which is the stochastic integral with respect to a p-thinned process:

$$\textit{(p-thinned gradient)} \qquad \nabla R_p(\theta) = \frac{1}{p}\int_0^T H_p^{\boldsymbol{\mathcal{G}}}(t;\theta)\left[dN_p(t) - \lambda_p^{\boldsymbol{\mathcal{G}}}(t;\theta)dt\right]. \tag{7.3}$$

Here $H_p^{\boldsymbol{\mathcal{G}}}(t;\theta)$ is related to $\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\theta)$.

Sub-interval gradient is defined as follows. Let τ_0, τ_1, τ_2, $\ldots$, $\tau_{\lceil 1/p\rceil}$ be a partition of $[0,T)$, where $\tau_0 = 0$, and $\tau_{\lceil 1/p\rceil} = T$. We cut the interval into $\lceil 1/p\rceil$ pieces so that the batch size is comparable to that in the thinned gradient. At every step, one interval is selected with probability p. We define the *sub-interval gradient* on $[\tau_i, \tau_{i+1})$ by,

(sub-interval gradient)

$$\nabla R_\ell(\theta) = \frac{1}{p}\int_0^T I\{t\in[\tau_i,\tau_{i+1})\}H^{\boldsymbol{\mathcal{F}}}(t;\theta)\big[dN(t) - \lambda^{\boldsymbol{\mathcal{F}}}(t;\theta)dt\big],$$

where I is an indicator representing whether the sub-interval is selected or not. Here we consider the full history. It can be easily verified that $\nabla R_\ell(\theta)$ is an unbiased estimation of the full gradient in Eq. (7.2). The thinned gradient can be used as an estimator of full gradient, which will be illustrated in Section 5. This definition also generalizes the stochastic optimization method proposed in [66], which splits the observation timeline at the arrival of each event.

7.4 Thinning for Parameter Estimation

In this section, we discuss how to estimate the parameter $\theta \in \mathbb{R}^d$ of the intensity function $\lambda(t;\theta)$ given the thinned history $\boldsymbol{\mathcal{G}}$. We first define the notations used.

- $\theta^*_{\boldsymbol{\mathcal{H}}}$: true parameter of $\boldsymbol{\mathcal{H}}$-intensity $\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta)$, such that $\mathbb{E}\nabla R(\theta^*_{\boldsymbol{\mathcal{H}}}) = 0$;
- $\theta^*_{\boldsymbol{\mathcal{G}}}$: true parameter of $\boldsymbol{\mathcal{G}}$-intensity $\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\theta)$, such that $\mathbb{E}\nabla R_p(\theta^*_{\boldsymbol{\mathcal{G}}}) = 0$;
- $\hat{\theta}_{\boldsymbol{\mathcal{H}}}$: estimate of $\theta^*_{\boldsymbol{\mathcal{H}}}$, such that $\sum_i \nabla R(\hat{\theta}_{\boldsymbol{\mathcal{H}}};\omega_i) = 0$;
- $\hat{\theta}_{\boldsymbol{\mathcal{G}}}$: estimate of $\theta^*_{\boldsymbol{\mathcal{G}}}$, such that $\sum_i \nabla R_p(\hat{\theta}_{\boldsymbol{\mathcal{G}}};\omega'_i) = 0$, where ω'_i is a realization of the p-thinned process.

The task of parameter estimation on a thinned history is to find $\tilde{\theta}_{\boldsymbol{\mathcal{H}}}$, such that $\mathbb{E}\big[\nabla R(\tilde{\theta}_{\boldsymbol{\mathcal{H}}})|\boldsymbol{\mathcal{G}}\big]$ is close enough to 0. We refer to $\tilde{\theta}_{\boldsymbol{\mathcal{H}}}$ as *the M-estimator on thinned history*. Here the expectation is over the thinning operation. The tilde is used to indicate that $\tilde{\theta}_{\boldsymbol{\mathcal{H}}}$ is a $\boldsymbol{\mathcal{G}}$-measurable estimator for the parameter of $\boldsymbol{\mathcal{H}}$-intensity $\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta)$, whereas $\hat{\theta}_{\boldsymbol{\mathcal{H}}}$, with a hat on it, is $\boldsymbol{\mathcal{H}}$-measurable. A notable result is that M-estimators have asymptotic normality [43], thus we have $\hat{\theta}_{\boldsymbol{\mathcal{H}}} \xrightarrow{P} \theta^*_{\boldsymbol{\mathcal{H}}}$ and $\hat{\theta}_{\boldsymbol{\mathcal{G}}} \xrightarrow{P} \theta^*_{\boldsymbol{\mathcal{G}}}$, as the number of realizations $n \to \infty$.

In the following, we first present a method for parameter estimation of a non-homogeneous Poisson process (NHPP) whose intensity is deterministic. We then derive a theorem that works for a more general type of intensities.

Lemma 7.5 (Thinning for parameter estimation of NHPP)**.** Consider an NHPP $N(t)$ with deterministic intensity $\lambda(t;\theta)$, $t > 0$, $\theta \in \mathbb{R}^d$. If there exists an invertible linear operator $\mathcal{A} : \mathbb{R}^d \to \mathbb{R}^d$ satisfying $\lambda(t;\mathcal{A}\theta) = p\lambda(t;\theta)$, then the M-estimator on thinned history can be written as $\tilde{\theta}_{\boldsymbol{\mathcal{H}}} = \mathcal{A}^{-1}\hat{\theta}_{\boldsymbol{\mathcal{G}}}$ such that $\mathbb{E}\big[\nabla R(\tilde{\theta}_{\boldsymbol{\mathcal{H}}})|\boldsymbol{\mathcal{G}}\big] \xrightarrow{P} 0$, as the number of realizations $n \to \infty$.

Example (Parameter estimation for NHPP). Let consider an NHPP with intensity $\lambda(t;a,b,c,d) = a + b\sin(ct+d)$. We can find a diagonal matrix $\mathcal{A} = \mathrm{diag}(p,p,1,1)$ such that $\lambda(t;\mathcal{A}(a,b,c,d)) = pa+pb\sin(ct+d) = p\lambda(t;a,b,c,d)$. Thus the parameter given the thinned history can be estimated by $\mathcal{A}^{-1}(\hat{a},\hat{b},\hat{c},\hat{d}) = (1/p\hat{a}, 1/p\hat{b}, \hat{c}, \hat{d})$, where $\hat{a},\hat{b},\hat{c},\hat{d}$ are estimated on the thinned history.

Next, we focus on a more general type of intensities, called *decouplable intensity*. Most commonly-used point processes have decouplable intensities, including NHPPs, linear Hawkes processes, compound Poisson process, etc.

Definition 7.6 (Decouplable intensity)**.** An intensity function is said to be decouplable, if it can be written in such a form:

$$\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta) = g(t;\theta)^T m^{\boldsymbol{\mathcal{H}}}(t). \tag{7.4}$$

Here $g(t;\theta)$ is a deterministic vector-valued function that is continuous with respect to θ and does not contain any information regarding $\mathcal{H}_t$. $m^{\boldsymbol{\mathcal{H}}}(t)$ is an $\boldsymbol{\mathcal{H}}$-predictable vector-valued measure that does not contain any information regarding θ. Particularly, $\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta)$ is said to be linear if $g(t;\theta) = \theta$.

This category covers a multitude of state-of-the-art models, including Netcodec [17], parametric Hawkes [45], MMEL model [30], Granger causality for Hawkes [27], and the sparse low-rank Hawkes [47]. The next theorem demonstrates a similar result with Lemma 7.5 for decouplable intensities.

Theorem 7.7 (Thinning for parameter estimation of decouplable intensities)**.** Consider a point process $N(t)$ with decouplable intensity. If there exist invertible linear operators $\mathcal{A}$ and $\mathcal{B}$ satisfying $\mathcal{B}\mathbb{E}\big[m^{\mathcal{H}}(t)|\mathcal{G}\big] = m_p^{\mathcal{G}}(t)$, where $m_p^{\mathcal{G}}(t)$ is the component of thinned intensity $\lambda_p^{\mathcal{G}}(t)$, and $p\mathcal{B}^{-1}g(t;\theta) = g(t;\mathcal{A}\theta)$, then the M-estimator on thinned history can be written as $\tilde{\theta}_{\mathcal{H}} = \mathcal{A}^{-1}\hat{\theta}_{\mathcal{G}}$ such that $\mathbb{E}\big[\nabla R(\tilde{\theta}_{\mathcal{H}})|\mathcal{G}\big] \xrightarrow{P} 0$, as the number of realizations $n \to \infty$. Particularly, if $\lambda^{\mathcal{H}}(t;\theta)$ is linear, then $\mathcal{A} = p\mathcal{B}^{-1}$.

Example (Parameter estimation for Hawkes processes). Consider a one-dimensional Hawkes process with intensity $\lambda^{\mathcal{H}}(t;\mu,\alpha) = (\mu,\alpha)^T\big(1, m^{\mathcal{H}}(t)\big)$, where $m^{\mathcal{H}}(t) = \sum_{i=1}\phi(t-t_i)$. From the fact that $\mathbb{E}\big[m^{\mathcal{H}}(t)|\mathcal{G}\big] = 1/p\, m_p^{\mathcal{G}}(t)$, we obtain $\mathcal{B} = \mathrm{diag}(1,p)$. Thus Theorem 7.7 yields $\mathcal{A} = p\mathcal{B}^{-1} = \mathrm{diag}(p,1)$ and consequently μ and α can be estimated by $p\hat{\mu}$ and $\hat{\alpha}$, where $\hat{\mu}$ and $\hat{\alpha}$ are estimated on the thinned history. Similar results can be obtained on multi-dimensional linear Hawkes processes. This result reveals that the thinning operation does not change the endogenous triggering pattern in linear Hawkes processes.

Remark (Parameter estimation for multi-dimensional Hawkes processes). The thinning estimator is also valid for multi-dimensional Hawkes processes. Consider the i-th dimension of an m-dimensional Hawkes process. Its intensity function can be written as $\lambda_i^{\mathcal{H}}(t;\mu_i,\alpha_{i1},\cdots,\alpha_{im}) = (\mu_i,\alpha_{i1},\cdots,\alpha_{im})^T\big(1, m_1^{\mathcal{H}}(t),\cdots,m_m^{\mathcal{H}}(t)\big)$, which complies with the definition of decouplable intensity. Theorem 7.7 again yields a thinning estimator with the linear operator $\mathcal{A} = \mathrm{diag}(p,1,...,1)$.

7.5 Thinning for Gradient Estimation and Stochastic Optimization

So far we have discussed how to estimate the parameter given the thinned history. In fact, the gradient at any θ can also be recovered without knowing all the information about a point process. The following theorem describes the gradient estimation on the thinned history for decoupleable intensity.

Theorem 7.8 (Thinning for gradient estimation)**.** Let $N(t)$ be a point process with decouplable intensity $\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta) = g(t;\theta)^T m^{\boldsymbol{\mathcal{H}}}(t)$ in Eq. (7.4). If there exist invertible linear operators $\mathcal{A}$ and $\mathcal{B}$ satisfying $\mathcal{B}\mathbb{E}\big[m^{\boldsymbol{\mathcal{H}}}(t)|\boldsymbol{\mathcal{G}}\big] = m_p^{\boldsymbol{\mathcal{G}}}(t)$, where $m_p^{\boldsymbol{\mathcal{G}}}(t)$ is the component of thinned intensity $\lambda_p^{\boldsymbol{\mathcal{G}}}(t)$, and $p\mathcal{B}^{-1}g(t;\theta) = g(t;\mathcal{A}\theta)$, then

(1) $\mathbb{E}\big[\nabla R(\theta)|\boldsymbol{\mathcal{G}}\big] \leq 1/p\mathcal{A}^{-1}\nabla R_p(\mathcal{A}\theta)$, for R is LSE;

(2) $\mathbb{E}\big[\nabla R(\theta)|\boldsymbol{\mathcal{G}}\big] \leq \mathcal{A}^{-1}\nabla R_p(\mathcal{A}\theta)$, for R is MLE.

Particularly, if the intensity is deterministic, i.e., $m^{\boldsymbol{\mathcal{H}}}(t) = 1$, both equalities hold.

Remark. The thinned gradient can be transformed to a larger estimation of the full gradient, and an unbiased estimation for deterministic intensity. More specifically, the gradient estimation is unbiased if and only if $\mathbb{E}[\boldsymbol{\mathcal{H}}(t;\theta)\lambda(t;\theta)] = \mathbb{E}\boldsymbol{\mathcal{H}}(t;\theta)\mathbb{E}\lambda(t;\theta)$, as shown in the proof of Theorem 7.8. Here $\boldsymbol{\mathcal{H}}$ usually depends on the intensity function $\lambda(t;\theta)$, such as MLE estimator has $\boldsymbol{\mathcal{H}}(t;\theta) = -\nabla \log \lambda(t;\theta)$. The condition may not hold under such circumstances. For stochastic intensities, the thinned gradient may be biased, yielding an estimation larger than the ground truth. Some empirical results on Hawkes processes are shown in Figure 7.3. The next theorem shows that the thinned gradient has a smaller variance compared with the sub-interval gradient.

Theorem 7.9 (Variance comparison)**.** Let $\nabla\tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta)$ and $\nabla R_\ell(\theta)$ be the p-thinned and sub-interval gradient at θ, where $\nabla\tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta) = 1/p\mathcal{A}^{-1}\nabla R_p(\mathcal{A}\theta)$ for LSE and $\nabla\tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta) = \mathcal{A}^{-1}\nabla R_p(\mathcal{A}\theta)$ for MLE. The variance of the p-thinned gradient is no greater than that of the sub-interval gradient, i.e.,

$$\mathbb{V}\Big[\nabla\tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta)\Big] \leq \mathbb{V}\Big[\nabla R_\ell(\theta)\Big].$$

Remark. A Chebyshev error bound can be easily obtained, as a result of Theorems 7.8 and 7.9:

$$\begin{aligned}\mathbb{P}\Big(\big|\nabla\tilde{R}^{\boldsymbol{\mathcal{G}}} - \mathbb{E}\nabla\tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta)\big| > \epsilon\Big) &\leq \frac{\mathbb{V}\Big[\nabla\tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta)\Big]}{\epsilon^2} \\ &\leq \frac{\mathbb{V}\Big[\nabla R_\ell(\theta)\Big]}{\epsilon^2} = \frac{\dfrac{1-p}{p}\Big[\mathbb{E}\nabla R(\theta)\Big]^2 + \dfrac{1}{p}\mathbb{V}\big[\nabla R(\theta)\big]}{\epsilon^2},\end{aligned}$$

for any $\epsilon > 0$. Since $\nabla R(\theta)$ is a martingale integral (Eq. 7.2), we have $\mathbb{E}\nabla R(\theta) \to 0$, as the number of realizations increases. Hence, the left-hand side probability is bounded by $\mathcal{O}(\epsilon^{-2}p^{-1}\mathbb{V}\big[\nabla R(\theta)\big])$, which shows that the gradient estimation of

deterministic intensities will not be far from its true one, if the number of realizations is sufficiently large. Unfortunately, the result does not apply to stochastic intensities. Nonetheless, its effectiveness on stochastic intensities is empirically validated on real datasets with Hawkes processes in our experiments (See Figure 7.4).

Thinning for stochastic optimization. We have shown that thinning can be used for estimating parameters and gradients with less data.This inspires us to employ it to stochastic optimization. We propose a novel Thinning-SGD (TSGD) method for learning a point process with a parametric intensity function, as shown in Algorithm 4. At each iteration, a thinned dataset is used for computing the gradient. Compared with sub-interval variance, thinned gradient has a smaller variance, so that the convergence curve may have less fluctuations and find a path to the optimal solution faster. Thinning is also applicable to other gradient-based optimization algorithms such as Adam [126].

Algorithm 4: TSGD: Thinning Stochastic Gradient Descent

Input : Event sequences $\{\boldsymbol{t_i}\}$, learning rate α, thinning size p, convergence criterion,
the objective function of a parametric point process model $R(\theta)$.
Output : Optimal parameter θ^*.
Initialize θ;
Find $\mathcal{A}$ according to Theorem 7.7;
repeat
  Sample a p-thinning batch t_i' from one of the sequences t_i;
  Compute the thinned gradient $\tilde{R}^{\mathcal{G}}(\theta)$, where $\tilde{R}^{\mathcal{G}}(\theta)$ is defined in Theorem 7.9;
  $\theta \leftarrow \theta - \alpha \tilde{R}^{\mathcal{G}}(\theta)$;
until *Convergence criterion is satisfied*;

7.6 Experiments

In this section, we assess the performance of our proposed thinning sampling in three tasks: parameter estimation, gradient estimation, and stochastic optimization. All the experiments were conducted on a server with Intel Xeon CPU E5-2680 (2.80GHz) and 250GB RAM.

Parameter estimation. We conduct two experiments for this task on synthetic datasets. The first experiment is to test thinning on Hawkes processes. We simulate

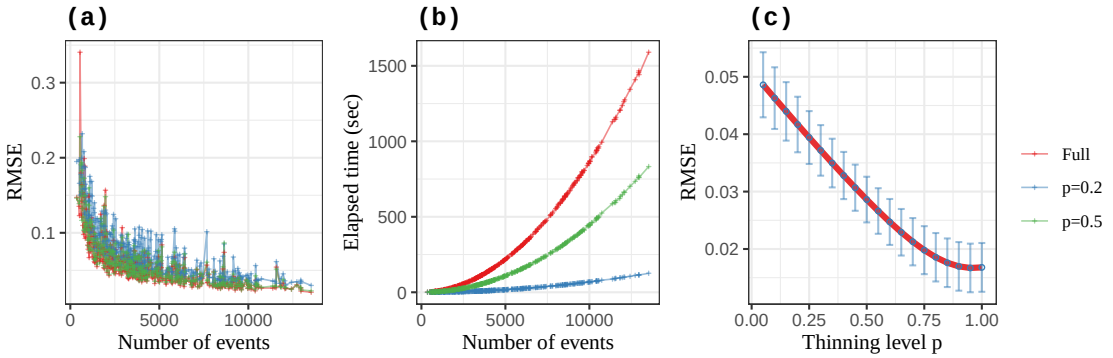


FIGURE 7.2: Parameter estimation on a 10-dimensional linear Hawkes process with LSE. (a): the RMSE of estimated parameters. (b): trainning time. (c): RMSE v.s. thinning level p.

100 sequences of 10-dimensional linear Hawkes processes and use different number of events for training. The longest sequence has around 14k events. The parameters of the process are randomly generated from a uniform distribution. For each dataset, we perform LSE with different histories: full data and p-thinned data with p = 0.2 and p = 0.5.

The results are shown in Figure 7.2. We can see that as the number of events training increases, the error (measured by RMSE) in parameter estimation decreases, at the cost of longer running time. A larger p value yields better estimations but also runs slower. When the number of events is large enough, the estimation with 0.2-thinning is as accurate as that with full data, but runs an order of magnitude faster. For a dataset of 14k events, 0.2-thinning only took 2 minutes, whereas the

TABLE 7.1: Parameter estimation on state-of-the-art models.

	Model	**RMSE/Accuracy**	**Training time (s)**
MMEL [29]	Full	0.0568 (0.0013)	38.03 (4.19)
	Thinned (p=0.5)	0.0569 (0.0012)	8.68 (1.06)
	Thinned (p=0.2)	0.0570 (0.0012)	3.94 (0.47)
Granger Causality for Hawkes [27]	Full	0.0161 (0.0078)	229.56 (17.87)
	Thinned (p=0.5)	0.0163 (0.0022)	65.68 (4.67)
	Thinned (p=0.2)	0.0167 (0.0010)	3.96 (1.80)
Sparse Low-rank Hawkes [30]	Full	97.46% (0.0133)	73.76 (42.24)
	Thinned (p=0.5)	97.60% (0.0166)	27.45 (17.51)
	Thinned (p=0.2)	96.63% (0.0243)	4.51 (2.65)

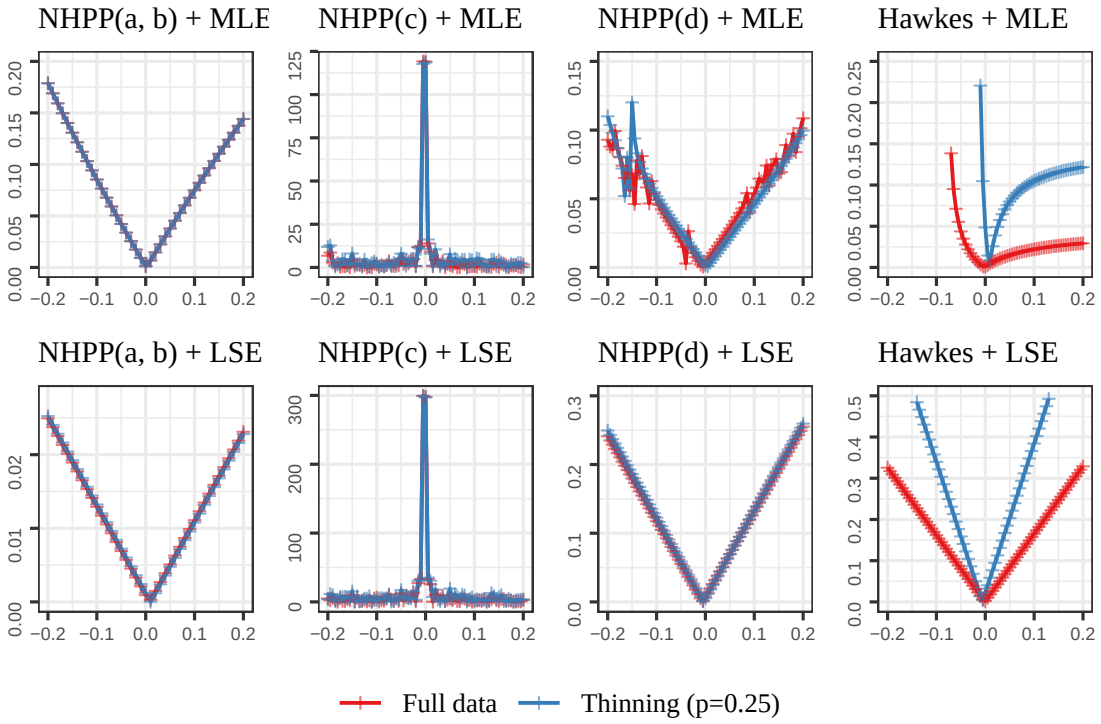


FIGURE 7.3: Gradient estimation for an NHPP and a linear Hawkes process using MLE and LSE. X-axes represent the RMSE of the parameters, and Y-axes the l_2-norm of gradient with corresponding parameters.

training on full data took 26.5 minutes, and the decrease of RMSE is less than 0.01. Figure 7.2 (c) shows that RMSE decreases as the thinning level p increases.

The second experiment is to test thinning for learning various state-of-the-art models: MMEL [29], Granger Causality for Hawkes [27] and Sparse Low-rank Hawkes [30]. We generate 30 sequences for each model and perform parameter estimation on different histories. The averages and standard deviations of the quality metric and training time are presented in Table 7.1. We use RMSE as the metric for MMEL and Granger Causality, and the accuracy of non-zero entries in the adjacency matrix for Sparse Low-rank Hawkes. It can be seen that thinning significantly reduces the training time of all models without compromising much estimation quality.

Gradient estimation. We consider two types of point process: a non-homogeneous point process with deterministic intensity $\lambda(t; a, b, c, d) = a + b\sin(ct + d)$; and a linear Hawkes process with $\boldsymbol{\mathcal{H}}$-intensity $\lambda^{\boldsymbol{\mathcal{H}}}(t) = (\mu + \alpha \sum \phi(t - t_i))$. The gradient at different values of parameters is computed and depicted in Figure 7.3.

The result shows three facts. First, every line in the figure touches X-axis at the origin, except for `NHPP(c)` (indifferentiable). This phenomenon demonstrates that thinning sampling yields asymptotically unbiased parameter estimation, no matter for LSE or MLE. Second, we can see that red and blue lines in the results of first 6 sub-figures overlap significantly, which confirms that thinning gives unbiased gradient estimation for deterministic intensities. Third, in the last two sub-figures, blue lines tend to be on or above the red ones, which demonstrates that thinning makes gradient estimation larger or equal to the ground truth for stochastic intensities.

Stochastic optimization. We test thinning sampling for stochastic optimization algorithms, including *SGD* and *Adam*. The task is to learn a linear Hawkes process. We test *Thinning* (p=0.1), sub-interval sampling (*SubInt*), the stochastic optimization learning algorithm (*StoOpt*) proposed in [66], combined with *SGD*, *ADAM* and the typical gradient descent (*GD*). We test on 4 datasets:

- **Synthetic dataset**: We simulate 10 realizations of a 5-dimensional linear Hawkes process, with parameters generated from a uniform distribution. The dataset contains 20k events. We train the model using the entire dataset and the RMSE between the estimated parameters and the ground truth is shown as test error.
- **IPTV dataset** [20]: The dataset consists of IPTV viewing events, which records the timestamps for multiple users watching a video, and the category that the video belongs to. Each user is treated as a realization and each category as a dimension. We select 7 and 3 realizations with 22k and 9k events as training and test datasets, respectively. The number of categories is 16.
- **NYC taxi dataset**: The data is from The New York City Taxi and Limousine Commission, which records fields capturing pick-up time, location and payment information of green taxis' orders. We select those trips starting from Manhattan district in the first 10 days of January 2018 and use the 14 areas as dimensions. The training and test datasets contain 60k and 12k events, respectively.
- **Weeplace dataset** [107]: This dataset contains the check-in histories of users at different locations. The categories of events include food, education, outdoors, shops, and 10 others. The check-in histories of 46 and 10 users are selected as training and test dataset, respectively. The sizes of the datasets are 50k and 11k.

https://www1.nyc.gov/site/tlc/index.page

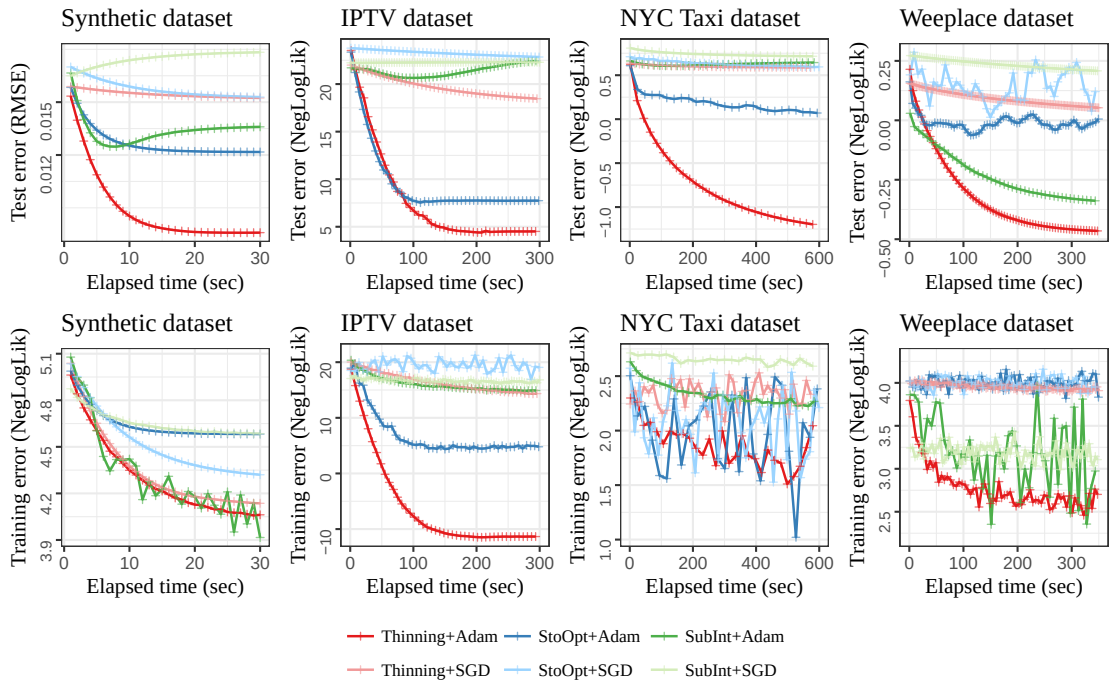


FIGURE 7.4: The average convergence curves of different learning algorithms on different datasets.

We ran each method on each dataset for 10 times. Figure 7.4 presents the average convergence curves of each method on different datasets. Training of *GD* failed to finish the first iteration given the maximum time shown in Figure 7.4 for each dataset and thus its results are not presented. From the learning curves, we can see that *Thinning+Adam* outperforms all the competitors in terms of test error on all the datasets. When looking at the *SGD* group alone, *Thinning* also achieves the lowest test error. From the bottom row, we see that *Thinning+Adam* tends to have less fluctuated learning curves. Especially on Weeplace and NYC taxi datasets, the fluctuations of *StoOpt* and *SubInt* are dramatic. This is due to the fact that thinning sampling can better capture the information of the whole timeline, whereas other methods are prone to a zigzag of searching path.

7.7 Summary

In this chapter, we discussed thinning as a downsampling method for point processes. Thinning operation uniformly compresses the intensity on time axis, but its structure is completely preserved. In this way, for parameter estimation, similar performance

can be achieved with less input data, as shown in the experiments. We also demonstrated how to estimate gradient on the thinned history, which leads to a novel stochastic optimization algorithm, called TSGD. Experimental results show that TSGD converges faster and has a learning curve with less fluctuations, which can be explained by the theorem that the thinning estimator for gradient has a smaller variance.

In previous chapters, special focus is given on incorporating features/marks/attribute associated with events. The thinning sampling method is also applicable when dealing with some marked Hawkes processes, ie., marked Hawkes processes with conditional independent marks and multi-dimensional Hawkes processes. For general marked point processes with complex mark distribution when it depends on time, thinning may fail in some circumstances.

Chapter 8

Conclusion

In this thesis, I study the problem of event sequence prediction and analysis using Hawkes processes. I pay particular attention to the modeling and acceleration aspects of Hawkes processes. This thesis contains several attempts to enhance Hawkes processes with better efficacy, flexibility, and efficiency. Some of the recent advancements and cutting-edge tools in the study of machine learning, such as transfer learning, probabilistic graphical models, variational inference, and deep learning, have been applied for enhancing Hawkes processes. In particular, this thesis has made the following contributions:

- **Tweedie Hawkes Process: Linking Features with Heavy-tailed Excitations.** Self-exciting event sequences, in which the occurrence of an event increases the probability of triggering subsequent ones, are common in many disciplines. In this chapter, I introduce a Bayesian model called Tweedie-Hawkes Processes (THP), which is able to model the outbreaks of events and find out the dominant factors behind. The model parameterizes the excitation parameter in Hawkes process with a Tweedie regression over event features. THP leverages on the Tweedie distribution in capturing various excitation effects. A variational EM algorithm is developed for model inference. Some theoretical properties of THP, including the sub-criticality and convergence of the learning algorithm, are discussed. Besides, a novel kernel bandwidth selection method for Hawkes processes is presented.

- **Network Transfer for Hawkes processes: Learning from both Cross-domain Temporal and Feature Information.** One of the most important

characteristics of Hawkes processes is that they link the occurrence of events up to the network structure, which makes it possible to infer the network structure from nothing but the events dynamics. However, cross-domain and feature information, which is also instrumental in modeling, is always neglected in existing works. In this chapter, I explore the idea of network transfer for Hawkes processes to leverage cross-domain information. I especially incorporate features in order to enhance the performance of transfer learning. I instantiate the idea by two models *trHLSH* and *BTHM*, from parametric and Bayesian perspectives, respectively. Both models augment Hawkes processes with feature and cross-domain information. I also present effective learning algorithms for each model. Evaluation on both synthetic and real-world datasets demonstrates that the proposed models can jointly learn knowledge from temporal, feature and cross-domain information, and have better performance in terms of network recovery and prediction.

- **Graph Convolutional Hawkes Processes: A Fast Graph-based Neural Model for Event Sequences.** A recent research line focuses on incorporating RNNs with the statistical model—marked Hawkes processes, which is the conventional tool for dealing with attributed event sequences. Existing methods, however, suffer from the limitations of failing to consider continuous features, relatively time-consuming training process, and restricted intensity assumptions. This chapter introduces a novel model called Graph Convolutional Hawkes Processes (GCHP) that eschews recurrent units and is able to learn a non-linear marked Hawkes process via graph convolutional layers. The model provides a general framework for feature embedding in attributed event sequences, and learns a nonlinear intensity without pre-defined form. Our model learns point processes with only graph convolutional layers and therefore it can be easily accelerated by the parallel mechanism. The model shows great prediction accuracy and efficiency in the experiment.

- **Accelerating the Learning Process via Thinning**. This chapter discusses one of the most fundamental issues about point processes that what is the best sampling method for point processes. I propose thinning as a downsampling method for accelerating the learning of point processes. I find that the thinning operation preserves the structure of intensity, and is able to estimate parameters with less time and without much loss of accuracy. Theoretical results including intensity, parameter and gradient estimation on

a thinned history are presented for point processes with decouplable intensities. A stochastic optimization algorithm based on the thinned gradient is proposed. Experimental results on synthetic and real-world datasets validate the effectiveness of thinning in the tasks of parameter and gradient estimation, as well as stochastic optimization.

It can be seen that, event sequences, which are one of the most ubiquitous categories of data, will be seen more and more applications in the future. Potential applications includes:

- **Sequential recommendation.** In e-commercial platforms, users purchasing history can be viewed as an event sequence. The task to predict what the user will buy in the future is the core of the industrial recommendation. Hawkes processes-related model can capture the dynamics and can predict not only what the user will buy, but also when. Some successful applications in recommendation have been seen in practice.
- **Social network analysis.** Hawkes processes are naturally embedded with the graph structure. One of the most successful applications of Hawkes processes in social network analysis is network structure inference. In this application, the network structure of users is unknown, and the task is to find the network structure from the observable event sequences, such as users' timelines, browsing records, etc. Hawkes processes can model the information diffusion process. Potential applications in community detection, misinformation mitigation and population prediction have also been explored by some researches [24].
- **Financial engineering**. Financial event sequences often exhibit the "chasing the market" behavior, that is, the selling or buying action can successively trigger followers to sell or buy, leading to the non-random fluctuation of stock price. Hawkes processes can model the self-/mutually triggering pattern among these events. Therefore, equipped with out methods, Hawkes processes can be more powerful when it comes to the applications in financial engineering.

In the future, I believe this study will inspire new research directions towards more complex application scenarios and better understanding of point processes. For

future work, modeling and algorithmic acceleration will be everlasting topics for event sequence prediction and analysis. In the following, I will give four directions that would be interesting to explore in the future:

- For the **modeling** aspect:

  Hawkes processes on manifolds. Currently, Hawkes processes are only considered on Euclidean space in the literature. An intrinsic assumption is that the system progresses linearly. However, many real-world phenomena have shown more complex characteristics, eg., the temporal information is not evolving in an Euclidean space but rather a hyperbolic one. In this case, current analysis methods may fail. Considering Hawkes processes and other point processes on a manifold can provide a more flexible model if the geometry of the processes do not comply with the Euclidean space assumption.

  Graph learning and Hawkes processes. Hawkes processes are naturally embedded with graph structures through the infectivity matrix $\boldsymbol{A}$. As the developing of graph representation learning, more and more complex tasks have been successfully solved by the newest technologies. In these tasks, graphs are often considered to be static. Hawkes processes is able to take the dynamics of the systems into account and may provide more flexible alternatives for the existing methods. It would be interesting to explore the direction of aggregating the information in both the graph structure and the temporal axis. In this direction, I believe recent development in the area of spatial-temporal graph neural networks can help achieve promising results.

  Better incorporation of auxiliary information. An event sequence is often associated with auxiliary information in real-world applications. In practice, it is often accessible to some features regarding each event or sequence, i.e., event and sequence features. The GCHP model proposed in Chapter 6 provides a solution for learning the embedding representations for event features. It would be interesting when node and edge features are considered.

- For the **algorithm** aspect:

  Large-scale learning of high-dimensional Hawkes processes. In Chapter 7, I have proposed a downsampling method for the acceleration of a uni-dimensional temporal Hawkes process. This method is valid when the sequence contains a large number of events. In real-world applications, an

often-encountered scenario is that there are a great number of nodes, which makes the Hawkes process high-dimensional. For example, in recommender systems, the numbers of users and items can be massive. Fast algorithms for the inference of high-dimensional Hawkes processes are always in demand. I believe some recent developments in high-dimensional statistics and randomized algorithms may serve as a potential solution.

Bias-corrected estimator for Hawkes processes. During the experiments, a phenomenon that I have encountered multiple times is that the MLE and LSE tend to overestimate the base intensity $\boldsymbol{\mu}$ and underestimate the infectivity $\boldsymbol{A}$. In part, one possible reason might be the inappropriate selection of the kernel and its bandwidth. A bias-corrected estimator for Hawkes processes would be valuable.

Appendices

Appendix A

Some Basic Concepts

A point process can be viewed as a random measure: a point process N is a measurable mapping of a probability space $(\Omega, \mathfrak{F}, P)$ onto a *simple* point measure space $(\mathcal{N}, \mathfrak{B}_{\mathcal{N}})$, where Ω is usually taking non-negative real half-line $\mathbb{R}^+$, $\mathfrak{F}$ is the Borel σ-algebra of Ω, $\mathcal{N}$ is the space of all boundedly finite integer-valued measures on $\mathbb{N}$, and equipped with the σ-algebra $\mathfrak{B}_{\mathcal{N}}$. The measure is said to be simple if at most one point arrives at a single point of time, i.e. $N(\{t\}) \leq 1$, $\forall t \in \mathbb{R}^+$, more specifically, the arrivals $t_1, t_2, \ldots$ are distinct (a.s.). The differences between arrival times is called the *interarrival times*, denoted by $\delta_i = t_i - t_{i-1}$. The fundamental relation between the point process $N(t)$ and the arrival time t_n is that for each n and s, $\{N(s) \leq n\} = \{t_i < s\}$.

We formally define the point process $N(t) = N((0, t])$ in the manner of a measurable enumeration as follows.

Definition A.1 (Point process)**.** Let t_i be the i-th the arrival times of a point process N defined by, $t_i = \inf_t \{N(t) \geq i\}$. A point process process N on $\mathbb{R}^+$ is defined by,

$$N(t) = \sum_i \delta_{t_i}(t), \tag{A.0}$$

where δ_ω is the Dirac measure at ω.

We see that if t is fixed, $N(t, \cdot)$ becomes a random variable. Given a realization ω, $N(t, \omega)$ is a *càdlàg* (right continuous with left limits) step function. The "information" available at time t is represented by a sub-σ-algebra $\mathcal{H}_t = \sigma(N(t) : t \in \mathbb{R}^+)$. The

filtration $\boldsymbol{\mathcal{H}} = (\mathcal{H}_t)_{0 \leq t < \infty}$ is called the *(internal) history*. A point process $N(t)$ is $\boldsymbol{\mathcal{H}}$*-predictable* if $N(t)$ is $\mathcal{H}_t$-measurable for all $0 \leq t < \infty$.

What plays an essential role in modeling point processes is the *intensity function*, aka, *intensity measure*. It measures the probability that a point will arrive in an infinitesimal period of time given all the historical information up to the current time. To define the intensity of a point process in the general case we follow the definition of stochastic intensity introduced in [120, 121].

Definition A.2 (Stochastic intensity)**.** Let $N(t)$ be a $\boldsymbol{\mathcal{H}}$-adapted point process, and let $\lambda(t)$ is a nonnegative $\boldsymbol{\mathcal{H}}$-predictable process such that for all $t \leq 0$, $\int_o^t \lambda(s)ds < 0$, a.s. Then $\lambda(t)$ is the $\boldsymbol{\mathcal{H}}$-intensity of $N(t)$ if for any nonegative $\boldsymbol{\mathcal{H}}$-predictable processes $C(t)$, the equality below is satisfied,

$$\mathbb{E}\left[\int_0^\infty C(s)dN(s)\right] = \mathbb{E}\left[\int_0^\infty C(s)\lambda(s)ds\right]. \tag{A.0}$$

The process $N(t)$ is said to be directed by the $\boldsymbol{\mathcal{H}}$-intensity $\lambda^{\boldsymbol{\mathcal{H}}}(t)$. We loosely write $\lambda(t)dt = \mathbb{E}dN(t)$, where $dN(t) = N((t, t+dt])$.

The expectation of a point process $N(t)$ is called the $\boldsymbol{\mathcal{H}}$*-compensator*, which is the cumulative intensity $\Lambda(t) = \int_0^t \lambda(s)ds$. Doob-Meyer decomposition yields that $N(t) - \Lambda(t)$ is a $\boldsymbol{\mathcal{H}}$-adapted martingale. Another important result is that the intensity measure determines the probability structure of the point process uniquely (Proposition 7.2.IV, [122]). Similar results can be extend to the compensator, as stated in (Proposition 14.1.VI, [123] and Theorem 2.19, [121]). Therefore, the core of learning a point process model is to learn the intensity measure or the compensator.

Definition A.3 (Poisson process)**.** Let λ be an intensity measure of a point process $N(t)$. $N(t)$ is a Poisson process if for any finite family of disjoint intervals $\{\Delta_i\}_{i=1,\ldots,k}$:

$$\mathbb{P}(N(\Delta_i) = n_i, i = 1, \ldots, k) = \prod_{i=1}^{k} \frac{[\Lambda(\Delta_i)]_i^n}{n_i!} e^{-\Lambda(\Delta_i)}$$

where $\Lambda(\Delta_i) = \int_{\Delta_i} \lambda(s)ds$.

Poisson process is the most important class of point processes and has several subclasses:

- *Homogeneous Poisson process*, when $\lambda(t) = \lambda$ is a constant;
- *Non-homogeneous Poisson process*, when $\lambda(t)$ is a deterministic function of t;
- *Doubly-stochastic Poisson process* (a.k.a. *Cox process)*, when $\lambda(*)$ is a random measure.

Next we define one of the most used doubly-stochastic Poisson processes, where Hawkes processes is an illustrative example.

Definition A.4 (One-dimensional Hawkes process)**.** A Hawkes process $N(t)$ is a doubly-stochastic point process such that the its intensity measure can be written as

$$\lambda(t) = \mu + \int_0^t \phi(s) dN(s), \tag{A.0}$$

where μ is a constant, called exogenous intensity, and $\phi(s)$ is a decreasing function whose support is on the positive half real line.

We can see from Eq. (A.4) that its intensity measure $\lambda(t)$ depends on the process itself. A more intuitive definition of the intensity measure is called the *conditional intensity function*:

$$\lambda^*(t) = \lim_{\Delta \to 0} \frac{\mathbb{E}[N(t+\Delta) - N(t) | \mathcal{H}_t]}{\Delta}.$$

Alternatively, Hawkes processes can be defined by a specific form of conditional intensity function:

$$\lambda^*(t) = \mu + \sum_{t_i < t} \phi(t - t_i).$$

Next theorem gives the log-likelihood function in terms of the conditional intensity function.

Theorem A.5 (Likelihood with condistional intensity function)**.** Let $N(t)$ be a Poisson process with conditional intensity function $\lambda^*(t)$. Given an observation window $[0, T]$ and a realization denoted by $t_1, \ldots, t_{N(T)}$, the log-likelihood function is expressible in the form,

$$\ell = \sum_{i=1}^{N(T)} \log(\lambda^*(t_i)) - \int_0^T \lambda^*(s) ds.$$

The proof can be found in [122]. An important consequence of the theorem is the maximum likelihood estimators, which are to maximize the log-likelihood function with respect to the parameters.

Appendix B

Proofs

B.1 Proof of Lemma 4.1

Lemma 4.1 (Concavity). The ELBO defined by Eq.(4.15) is concave in $\tilde{\eta}_i$ for each i, if $p < 2$.

Proof. This proof is achieved by showing the second derivative of the ELBO is non-positive. Before we start the proof, We introduce some notations for simplification. Let,

$$\begin{aligned} a_k &= \phi(t_k - t_i) > 0, \\ b_k &= \sum_{j \neq i}^{k-1} \eta_j^p \phi^2(t_k - t_j) > 0, \\ c_k &= \mu + \sum_{j \neq i}^{k-1} \eta_j \phi(t_k - t_j) > 0, \\ x &= \tilde{\eta}_i. \end{aligned}$$

Then the ELBO function can be written by,

$$\text{ELBO} = \sum_{k=i+1}^{n} \left[\ln(a_k x + c_k) - \frac{x^p a_k^2 + b_k}{2(a_k x + c_k)^2} \right] + f(x),$$

where

$$f(x) = -\tilde{\eta}_i \int_0^T \phi(t - t_i)dt + \frac{\tilde{\eta}_i \left(\eta_i^{1-p} - \tilde{\eta}_i^{1-p}\right)}{\psi(1-p)} + \frac{\tilde{\eta}_i^{2-p}}{\psi(2-p)}.$$

It suffices to show that $g(x)$, which is defined by,

$$g(x) = \ln(ax + c) - \frac{x^p a^2 + b}{2(ax + c)^2},$$

is concave. Given $1 < p < 2$ and $0 < x < 1$, we have,

$$0 < x^2 < x^p < x^{2-p} < 1.$$

The first and second derivative of $g(x)$ can be written as,

$$\begin{aligned}
g'(x) &= \frac{a}{ax + c} - \frac{a^2 p x^{p-1}}{2(ax + c)^2} + \frac{a(a^2 x^p + b)}{(ax + c)^3} - \frac{x^{1-p}}{(n - i)(1 - p)}, \\
g''(x) &= -\frac{a^2}{(ax + c)^2} - \frac{p(p - 1)a^2 x^{p-2}}{2(ax + c)^2} + \frac{2a^3 p x^{p-1}}{(ax + c)^3} - \frac{3a^2(b + a^2 x^p)}{(ax + c)^4} \\
&= \frac{-a^2 x^p \left(c^2(p - 1)p + 2ac(p - 3)px + a^2(p - 3)(p - 2)x^2\right) - 2a^2 x^{2-p}(3b + (c + ax)^2)}{2x^2(c + ax)^4} \\
&< \frac{-a^2 x \left(c^2(p - 1)p + 2ac(p - 3)px + a^2(p - 3)(p - 2)x^2 + 2(3b + (c + ax)^2)\right)}{2x^2(c + ax)^4}.
\end{aligned}$$

To show $g''(x) < 0$, we only need to show that

$$c^2(p - 1)p + 2ac(p - 3)px + a^2(p - 3)(p - 2)x^2 + 2(3b + (c + ax)^2) > 0.$$

Denote the left hind side by $m(x)$. It can be reformulated as a quadratic function of x,

$$\begin{aligned}
m(x) &= a^2(p^2 - 5p + 8)x^2 + 2ac(p - 2)(p - 1)x + c^2(p - 1) * p + 2c^2 + 6b \\
&> a^2(p^2 - 4p + 4)x^2 + 2ac(p - 2)(p - 1)x + (p^2 - 2p + 1)c^2 + 6b \\
&= [a(p - 2)x + (p - 1)c]^2 + 6b > 0.
\end{aligned}$$

Thus $g''(x) < 0$, and $g(x)$ is concave. It is easy to see that $f(x)$ is concave. Therefore the proof of the lemma is complete. □

Remark. In practice, especially in insurance pricing, p is always assumed to smaller than 2 when applying Tweedie regression. This is because the Tweedie distribution has a positive mass at zero when $1 < p < 2$. [127] states that Tweedie distribution

is zero-inflated when $1 < p < 2$ that the probability of a Tweedie random variable taking value zero is $\exp\left(-\frac{\eta^{2-p}}{\psi(2-p)}\right)$.

B.2 Proof of Theorem 4.3

theorem 4.3(Sub-criticality) The THP model consists of sub-critical Galton-Watson branching processes if the link function is, (1) invertible, and (2) mapping $(0, 1)$ onto $\mathbb{R}$. The number of the Galton-Watson branching processes is a Poisson process of rate μ.

Proof. We need to clarify a key assumption about the regression before we start the proof. According to the assumption that feature vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are deterministic or random, regression models can be categorized into fixed design and random design. In this thesis we only consider fixed design. In fixed design, it is assumed that all the feature vectors are drawn from a finite feature set $\mathbb{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. The reason for adopting this assumption is that the expectation of each α_i (i.e. η_i) is bounded by some less than 1,

$$\eta_i = g^{-1}(\boldsymbol{x}_i^T \boldsymbol{\beta}) \leqslant \eta^*,$$

where

$$\eta^* = \sup_{\boldsymbol{x}_i \in \mathbb{X}} g^{-1}(\boldsymbol{x}_i^T \boldsymbol{\beta}).$$

Since $\mathbb{X}$ is finite, we have $\eta^* < 1$. Then,

$$\begin{aligned}
\mathbb{E}N_{j+1}^{\ddagger} &= \mathbb{E}\left[\mathbb{E}\left(N_{j+1}^{\ddagger} | N_j^{\ddagger}\right)\right] && \text{(conditional expectation)} \\
&= \mathbb{E}\sum_{k=1}^{N_j^{\ddagger}} \int_0^{+\infty} \alpha_{kj}\phi(t - t_{kj})dt && \text{(expectation of non-homogeneous Poisson process)} \\
&= \mathbb{E}\sum_{k=1}^{N_j^{\ddagger}} \alpha_{kj} \\
&\leqslant \eta^* \mathbb{E}N_j^{\ddagger} && \text{(expecation of tweedie-distributed } \boldsymbol{\alpha}\text{'s)} \\
&< \mathbb{E}N_j^{\ddagger} && \text{(fixed design and condition (2)).}
\end{aligned}$$

□

Remark. This result shows that the expected number of a Galton-Watson braching process can be bounded,

$$\mathbb{E}\sum_{j=1}^{\infty} \boldsymbol{N}_{ij}^{\ddagger} \leqslant \lim_{n\to+\infty} \frac{1-(\eta^*)^n}{1-\eta^*} = \frac{1}{1-\eta^*}.$$

However, for some β's in PHP such that $\eta^{\S} = \inf_{\boldsymbol{x}_i \in \mathbb{X}} \boldsymbol{x}_i^T \boldsymbol{\beta} > 1$, then for a Galton-Watson branching process, the expectation of total number of events would be,

$$\mathbb{E}\sum_{j=1}^{\infty} \boldsymbol{N}_{ij}^{\ddagger} > \lim_{n\to+\infty} \frac{1-(\eta^{\S})^n}{1-\eta^{\S}} = \infty.$$

This explains why when simulating event sequences, the PHP may not necessarily generate finite number of events.

B.3 Proof of Theorem 4.4

Theorem 4.4 (Local Optima). For any $k = 1, 2, \cdots$, we have,

$$\mathcal{L}^{(k+1)} \geq \mathcal{L}^{(k)}, \tag{B.0}$$

where $\mathcal{L}^{(k)} = \ln p(\boldsymbol{t}|\mu^{(k)}, \boldsymbol{\beta}^{(k)})$ denotes the incomplete log-likelihood of k-th iteration in the learning algorithm of THP.

Proof. Theorem 4.4 states that the learning algorithm can generate a monotonically increasing log-likelihood sequence such that a local optima will be achieved. This result is obtained by applying Theorem 3 in [128]. It states that if the Q function $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}')$ is continuous in both $\boldsymbol{\Theta}$ and $\boldsymbol{\Theta}'$, and monotonically increasing in each M-step, then the incomplete log-likelihood $\mathcal{L}(\boldsymbol{\Theta})$ converges to a local optima at some stationary points $\boldsymbol{\Theta}^*$. The continuity of the Q function can be easily seen. On the other hand, the variational method in E-step guarantees the log likelihood to increase by at least as much as the ELBO does, otherwise it stagnates at a local optima. Next we give the detailed proof.

Applying Theorem 3 in [128], it suffices to show two facts:

(i) $Q\left(\mu^{(k+1)}, \boldsymbol{\beta}^{(k+1)} \middle| \mu^{(k)}, \boldsymbol{\beta}^{(k)}\right)$ is continuous in $\mu^{(k+1)}$ and $\boldsymbol{\beta}^{(k+1)}$. And,

$$\text{(ii)} \quad \sup_{\mu^{(k+1)},\boldsymbol{\beta}^{(k+1)}} Q\left(\mu^{(k+1)},\boldsymbol{\beta}^{(k+1)}\middle|\mu^{(k)},\boldsymbol{\beta}^{(k)}\right) \geqslant Q\left(\mu^{(k)},\boldsymbol{\beta}^{(k)}\middle|\mu^{(k)},\boldsymbol{\beta}^{(k)}\right).$$

For (i), as all the components of it are continuous, the Q-funtion defined by Eq. (19) and (20) is consequently continuous. For (ii), since we maximize the Q function regarding $\mu^{(k+1)}$, $\boldsymbol{\beta}^{(k+1)}$, it must be larger than $\mu^{(k)}$, $\boldsymbol{\beta}^{(k)}$, otherwise μ and $\boldsymbol{\beta}$ is converged.

Alternatively, the learning algorithm can also be interpreted as a mean-field variational Bayesian inference framework, only if we assume degenerate distribution for μ and $\boldsymbol{\beta}$. [129] has shown that, for the exponential family models with missing values, the mean-field variational Bayesian estimator, mean of the variational posterior distribution, converges locally to the true value with probability 1 as the sample size becomes infinitely large. □

B.4 Proof of Theorem 4.5

Theorem 4.5 (Convergence). If the updating method for $Q(\mu|\mu')$ and $Q(\boldsymbol{\beta}|\boldsymbol{\beta}')$ is gradient descent (or Newton-like methods), then as $k \to \infty$,

$$\|\mu^{(k+1)} - \mu^{(k)}\| \to 0,$$

$$\|\boldsymbol{\eta}^{(k+1)} - \boldsymbol{\eta}^{(k)}\| \to 0.$$

In particular, the convergence holds for $\boldsymbol{\beta}$ that as $k \to +\infty$,

$$\|\boldsymbol{\beta}^{(k+1)} - \boldsymbol{\beta}^{(k)}\| \to 0,$$

if either of the following conditions is satisfied: (1) the link function g is uniformly continuous, and (2) the link function g satisfies the sub-critical conditions illustrated in Theorem 4.3 and $\eta_i^{(k)} \not\to 0$ or 1 for all i.

Proof. [129] states a sufficient condition for the convergence of paramters. Here we utilize the result of Eq. (18) in [129] and show that,

$$Q\left(\mu^{(k+1)}|\mu^{(k)}\right) - Q\left(\mu^{(k)}|\mu^{(k)}\right) \geqslant CQ'\left(\mu^{(k)}|\mu^{(k)}\right)\left(\mu^{(k+1)} - \mu^{(k)}\right), \tag{B.0}$$

$$Q\left(\boldsymbol{\eta}^{(k+1)}|\boldsymbol{\eta}^{(k)}\right) - Q\left(\boldsymbol{\eta}^{(k)}|\boldsymbol{\eta}^{(k)}\right) \geqslant CQ'\left(\boldsymbol{\eta}^{(k)}|\boldsymbol{\eta}^{(k)}\right)\left(\boldsymbol{\eta}^{(k+1)} - \boldsymbol{\eta}^{(k)}\right). \tag{B.0}$$

where C is a constant. When $\mu^{(k+1)} > \mu^{(k)}$,

$$\begin{aligned}
Q\left(\mu^{(k+1)}|\mu^{(k)}\right) - Q\left(\mu^{(k)}|\mu^{(k)}\right) &= \sum_{i=1}^{n} \ln\left[\frac{\mu^{(k+1)} + \sum_{j}^{i-1}\tilde{\eta}_i\phi(t_i - t_j)}{\mu^{(k)} + \sum_{j}^{i-1}\tilde{\eta}_i\phi(t_i - t_j)}\right] - \left(\mu^{(k+1)} - \mu^{(k)}\right)T \\
&= \sum_{i=1}^{n} \ln\left[1 - \frac{\mu^{(k)} - \mu^{(k+1)}}{\mu^{(k)} + \sum_{j}^{i-1}\tilde{\eta}_i\phi(t_i - t_j)}\right] - \left(\mu^{(k+1)} - \mu^{(k)}\right)T \\
&\geqslant \sum_{i=1}^{n} \frac{\mu^{(k)} - \mu^{(k+1)}}{\mu^{(k)} + \sum_{j}^{i-1}\tilde{\eta}_i\phi(t_i - t_j)} - \left(\mu^{(k+1)} - \mu^{(k)}\right)T \\
&= \sum_{i=1}^{n}\left(\frac{1}{\mu^{(k)} + \sum_{j}^{i-1}\tilde{\eta}_i\phi(t_i - t_j)} - T\right)\left(\mu^{(k+1)} - \mu^{(k)}\right) \\
&= Q'\left(\mu^{(k)}|\mu^{(k)}\right)\left(\mu^{(k+1)} - \mu^{(k)}\right).
\end{aligned}$$

When $\mu^{(k+1)} < \mu^{(k)}$,

$$
\begin{aligned}
& Q\left(\mu^{(k+1)}|\mu^{(k)}\right) - Q\left(\mu^{(k)}|\mu^{(k)}\right) \\
&= \sum_{i=1}^{n} \ln\left[\frac{\mu^{(k+1)} + \sum_{j}^{i-1} \tilde{\eta}_i \phi(t_i - t_j)}{\mu^{(k)} + \sum_{j}^{i-1} \tilde{\eta}_i \phi(t_i - t_j)}\right] - \left(\mu^{(k+1)} - \mu^{(k)}\right) T \\
&= -\sum_{i=1}^{n} \ln\left[1 + \frac{\mu^{(k)} - \mu^{(k+1)}}{\mu^{(k+1)} + \sum_{j}^{i-1} \tilde{\eta}_i \phi(t_i - t_j)}\right] - \left(\mu^{(k+1)} - \mu^{(k)}\right) T \\
&\geqslant -\sum_{i=1}^{n} \frac{\mu^{(k)} - \mu^{(k+1)}}{\mu^{(k)} + \sum_{j}^{i-1} \tilde{\eta}_i \phi(t_i - t_j)} - \left(\mu^{(k+1)} - \mu^{(k)}\right) T \\
&= \sum_{i=1}^{n} \left(\frac{1}{\mu^{(k)} + \sum_{j}^{i-1} \tilde{\eta}_i \phi(t_i - t_j)} - T\right) \left(\mu^{(k+1)} - \mu^{(k)}\right) \\
&= Q'\left(\mu^{(k)}|\mu^{(k)}\right) \left(\mu^{(k+1)} - \mu^{(k)}\right).
\end{aligned}
$$

For $\boldsymbol{\eta}$, we have

$$
\begin{aligned}
& Q(\boldsymbol{\eta}^{(k+1)}|\boldsymbol{\eta}^{(k)}) - Q(\boldsymbol{\eta}^{(k)}|\boldsymbol{\eta}^{(k)}) \\
&= \frac{1}{\psi} \sum_{i=1}^{n} \left[\frac{\tilde{\eta}_i^{(k)} \left\{\eta_i^{(k+1)}\right\}^{1-p}}{1-p} - \frac{\left\{\eta_i^{(k+1)}\right\}^{2-p}}{2-p} - \frac{\tilde{\eta}_i^{(k)} \left\{\eta_i^{(k)}\right\}^{1-p}}{1-p} + \frac{\left\{\eta_i^{(k)}\right\}^{2-p}}{2-p}\right].
\end{aligned}
$$

Without loss of generality, we assume $0 < \eta_i^{(k)} < \eta_i^{(k+1)} < 1$. Given $1 < p < 2$, we have,

$$
\left\{\eta_i^{(k+1)}\right\}^{-p} < \left\{\eta_i^{(k)}\right\}^{-p},
$$

yielding,

$$
\left\{\eta_i^{(k+1)}\right\}^{1-p} < \eta_i^{(k+1)} \left\{\eta_i^{(k)}\right\}^{-p}, \tag{B.0}
$$

and

$$
\left\{\eta_i^{(k+1)}\right\}^{2-p} < \eta_i^{(k+1)} \left\{\eta_i^{(k)}\right\}^{1-p}, \tag{B.0}
$$

Then, applying the above equations, the left hand side of Eq. (B.4) can be written as,

$$
\begin{aligned}
& Q\left(\boldsymbol{\eta}^{(k+1)}|\boldsymbol{\eta}^{(k)}\right) - Q\left(\boldsymbol{\eta}^{(k)}|\boldsymbol{\eta}^{(k)}\right) \\
=& \frac{1}{\psi(1-p)(2-p)} \sum_{i=1}^{n} \left\{ (2-p)\tilde{\eta}_i^{(k)} \left[\left\{\eta_i^{(k+1)}\right\}^{1-p} - \left\{\eta_i^{(k)}\right\}^{1-p} \right] \right\} \\
& - \sum_{i=1}^{n} \left\{ (1-p) \left[\left\{\eta_i^{(k+1)}\right\}^{2-p} - \left\{\eta_i^{(k)}\right\}^{2-p} \right] \right\} \\
\geqslant& \frac{1}{\psi(1-p)(2-p)} \sum_{i=1}^{n} \left\{ (1-p)\tilde{\eta}_i^{(k)} \left[\eta_i^{(k+1)} \left\{\eta_i^{(k)}\right\}^{-p} - \left\{\eta_i^{(k)}\right\}^{1-p} \right] \right\} \\
& - \sum_{i=1}^{n} \left\{ (1-p) \left[\eta_i^{(k+1)} \left\{\eta_i^{(k)}\right\}^{1-p} - \left\{\eta_i^{(k)}\right\}^{2-p} \right] \right\} \\
\geqslant& \frac{1}{\psi(2-p)} \sum_{i=1}^{n} \left\{ \left[\tilde{\eta}_i^{(k)} - \left\{\eta_i^{(k)}\right\} \right] \left\{\eta_i^{(k)}\right\}^{-p} \left[\eta_i^{(k+1)} - \eta_i^{(k)} \right] \right\} \\
=& \frac{1}{(2-p)} Q'\left(\boldsymbol{\eta}^{(k)}|\boldsymbol{\eta}^{(k)}\right)\left(\boldsymbol{\eta}^{(k+1)} - \boldsymbol{\eta}^{(k)}\right).
\end{aligned}
$$

The proof of the other case when $p > 2$ is similar. Therefore we omit the remaining parts of the proof.

Last but not the least, we solve $\boldsymbol{\beta}$ after we obtain $\boldsymbol{\eta}$. As we can see from the learning algorithm, $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$ are connected through the link function g. If we apply logit as the link function, for example, then finding $\boldsymbol{\beta}$ will become a logistic regression problem. It is worth noting that, in this circumstances, the estimation of $\boldsymbol{\beta}$ may not necessarily exist (C.f. Chapter 4.4 [130]). Indeed many studies have given some examples, such as [131]. This problem is due to the fact that the link function g is not uniformly continuous. One sufficient condition for $\boldsymbol{\beta}$ to converge is to use uniformly continuous link function. In fact, the convergence of $\boldsymbol{\beta}$ does not affect the convergence of the log-likelihood and the other parameters. □

B.5 Proof of Theorem 6.1

Theorem 6.1 (Moment matching). Let $\lambda_g(t|\mathcal{H}_{t-})$ be the ground intensity function of a point process $\mathcal{N}$. Let $\breve{\mathcal{N}}$ be an adjoint process such that (1) $\breve{\mathcal{N}}$ has piecewise constant intensity function; (2) the intensity function is continuous between

events; (3) its ground intensity function $\breve{\lambda}_g(t|\mathcal{H}_{t-})$ satisfying for any t, it has $\int_0^{T_{\breve{\mathcal{N}}_g(t)}} \breve{\lambda}_g(t|\mathcal{H}_{t-})dt = \int_0^{T_{\mathcal{N}_g(t)}} \lambda_g(t|\mathcal{H}_{t-})dt$, where $T_{\mathcal{N}_g(t)}$ is the arrive time of the $\mathcal{N}_g(t)$-th event. If λ_g is bounded by b, then

$$\mathbb{P}\left\{\sup_{\alpha\in[0,1]}\left\|\frac{1}{t}\int_{\mathcal{M}}\left(\mathcal{N}(\alpha t,m)-\breve{\mathcal{N}}(\alpha t,m)\right)dm\right\|\geqslant\epsilon\right\}\leqslant 2\exp\left\{-cb\min\left(\epsilon^2 t,\epsilon\right)\right\},$$

where c is a constant.

Proof. From the conditions above we can see that $\breve{N}_g((T_{i-1},T_i])$ has a Poisson distribution, for each period of time $(T_{i-1},T_i]$ where T_i is the i-th arrival time of N_g. We denote $N_g((T_{i-1},T_i])$ and $\breve{N}_g((T_{i-1},T_i])$ as N_i and $\breve{N}_i$, respectively. The expectation of $\breve{N}_i$ is $\int_{T_{i-1}}^{T_i}\lambda_g(t|\mathcal{H}_{t-})dt$. Thus, we have

$$\left\|\int_{\mathcal{M}}\left(N(\alpha t,m)-\breve{N}(\alpha t,m)\right)dm\right\| = \left\|\sum_{i=1}^{N_g(\alpha t)}(N_i-\breve{N}_i)\right\|.$$

We see that $N_i-\breve{N}_i$ is the difference of two Poisson-like distributions. Condition (3) yields that $\mathbb{E}\left(N_i-\breve{N}_i\right)=0$.

$N_i-\breve{N}_i$ is sub-exponential, as

$$\mathbb{E}\left[\exp\left(\lambda\left(N_i-\breve{N}_i\right)\right)\right]<\frac{\mathbb{E}\left[\exp\left(\lambda\left(N_i\right)\right)\right]}{\mathbb{E}\left[\exp\left(\lambda\left(\breve{N}_i\right)\right)\right]}<1.$$

As a result, we obtain that $\left\|N_i-\breve{N}_i\right\|_{\psi_1}<b$, where $\|\cdot\|_{\psi_1}$ is the sub-exponential norm. Applying the Bernstein's inequality for sub-exponential distributions, we

have

$$
\begin{aligned}
&\mathbb{P}\left\{\sup_{\alpha\in[0,1]}\left\|\frac{1}{t}\int_{\mathcal{M}}\left(N(\alpha t,m)-\breve{N}(\alpha t,m)\right)dm\right\|\geqslant\epsilon\right\}\\
=&\mathbb{P}\left\{\sup_{\alpha\in[0,1]}\left\|\sum_{i=1}^{N_g(\alpha t)}(N_i-\breve{N}_i)\right\|\geqslant\epsilon t\right\}\\
\lesssim&\mathbb{P}\left\{\sup_{\alpha\in[0,1]}\left\|\sum_{i=1}^{\lfloor at/b\rfloor}(N_i-\breve{N}_i)\right\|\geqslant\epsilon t\right\}\\
=&\mathbb{P}\left\{\left\|\sum_{i=1}^{\lfloor t/b\rfloor}(N_i-\breve{N}_i)\right\|\geqslant\epsilon t\right\}\\
\leqslant&2\exp\left\{-cb\min\left(\epsilon^2t,\epsilon\right)\right\},
\end{aligned}
$$

where the last step is a result of Bernstein inequality. □

B.6 Proof of Theorem 7.4

Theorem 7.4 (Thinned intensities). Let $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{G}}$ be the full history and thinned history with respect to a p-thinned process $N_p(t)$. Let $\boldsymbol{\mathcal{H}}$ be the internal history of $N(t)$. The following equalities hold:

(1) $\lambda_p^{\boldsymbol{\mathcal{F}}}(t)=p\lambda^{\boldsymbol{\mathcal{H}}}(t)$;

(2) $\lambda_p^{\boldsymbol{\mathcal{G}}}(t)=p\mathbb{E}\left[\lambda^{\boldsymbol{\mathcal{H}}}(t)|\boldsymbol{\mathcal{G}}\right]$.

Proof. For (1), it can be obtained by taking expectation on both side of $dN_p(t)=B_{N(t)}dN(t)$:

$$\lambda_p^{\boldsymbol{\mathcal{F}}}(t)=\mathbb{E}dN_p(t)=\mathbb{E}B_{N(t)}dN(t)=\lambda^{\boldsymbol{\mathcal{H}}}(t). \tag{B.0}$$

For (2), Theorem 7.13 in [121] gives a solution to recover the point process given the thinned history be the following conditional expectation:

$$\mathbb{E}\left[N(t)|\boldsymbol{\mathcal{G}}\right]=N_p(t)+\frac{1-p}{p}\int_0^t d\Lambda_p^{\boldsymbol{\mathcal{G}}}(s). \tag{B.0}$$

Here, $\Lambda_p^{\mathcal{G}}$ is the $\mathcal{G}$-compensator of the p-thinned process, which equals to $\Lambda_p^{\mathcal{G}}(t) = \int_0^t \lambda_p^{\mathcal{G}}(s)ds$. Further,

$$\begin{aligned}
\mathbb{E}\left[\lambda^{\mathcal{H}}(t)|\mathcal{G}\right] &= \lim_{s\to 0} \frac{\mathbb{E}\left[N(t+s) - N(t)|\mathcal{G}\right]}{ds} \\
&= \lim_{s\to 0} \frac{\mathbb{E}\left[N_p(t+s) - N_p(t)|\mathcal{G}\right]}{ds} + \frac{1-p}{p}\lambda_p^{\mathcal{G}}(t) \\
&= \lambda_p^{\mathcal{G}}(t) + \frac{1-p}{p}\lambda_p^{\mathcal{G}}(t) \\
&= \frac{1}{p}\lambda_p^{\mathcal{G}}(t).
\end{aligned}$$

where the desired result follows. □

B.7 Proof of Theorem 7.5

Theorem 7.5 (Thinning for parameter estimation of NHPP). Consider an NHPP $N(t)$ with deterministic intensity $\lambda(t;\theta)$, $t > 0$, $\theta \in \mathbb{R}^d$. If there exists an invertible linear operator $\mathcal{A} : \mathbb{R}^d \to \mathbb{R}^d$ satisfying $\lambda(t;\mathcal{A}\theta) = p\lambda(t;\theta)$, then the M-estimator on thinned history can be written as $\tilde{\theta}_{\mathcal{H}} = \mathcal{A}^{-1}\hat{\theta}_{\mathcal{G}}$ such that $\mathbb{E}\big[\nabla R(\tilde{\theta}_{\mathcal{H}})|\mathcal{G}\big] \xrightarrow{P} 0$, as the number of realizations $n \to \infty$.

Proof. From Theorem 7.13 in [121] we have,

$$\begin{aligned}
\mathbb{E}\big[\nabla R(\tilde{\theta}_{\mathcal{H}})|\mathcal{G}\big] &= \mathbb{E}\Big\{\frac{1}{p}\int_0^T H(t;\tilde{\theta}_{\mathcal{H}})\Big[dN_p(t) - p\lambda(t;\tilde{\theta}_{\mathcal{H}})dt\Big|\mathcal{G}\Big]\Big\} \\
&= \frac{1}{p}\int_0^T H(t;\tilde{\theta}_{\mathcal{H}})\Big[dN_p(t) - \lambda(t;\hat{\theta}_{\mathcal{G}})dt\Big] \\
&= \frac{1}{p}\int_0^T H(t;\tilde{\theta}_{\mathcal{H}})\Big[dN_p(t) - \lambda(t;\theta^*_{\mathcal{G}})dt + \lambda(t;\theta^*_{\mathcal{G}})dt - \lambda(t;\hat{\theta}_{\mathcal{G}})dt\Big] \\
&= \frac{1}{p}\int_0^T H(t;\tilde{\theta}_{\mathcal{H}})\Big[\lambda(t;\theta^*_{\mathcal{G}})dt - \lambda(t;\hat{\theta}_{\mathcal{G}})dt\Big] \xrightarrow{P} 0.
\end{aligned}$$

The last step is due to the asymptotic normality of M-estimator ([43]) that $\hat{\theta}_{\mathcal{G}} \xrightarrow{P} \theta^*_{\mathcal{G}}$ as the number of realizations $n \to \infty$. □

B.8 Proof of Theorem 7.7

Theorem 7.7 (Thinning for parameter estimation of decouplable intensities). Consider a point process $N(t)$ with decouplable intensity. If there exist invertible linear operators $\mathcal{A}$ and $\mathcal{B}$ satisfying $\mathcal{B}\mathbb{E}\big[m^{\mathcal{H}}(t)|\mathcal{G}\big] = m_p^{\mathcal{G}}(t)$, where $m_p^{\mathcal{G}}(t)$ is the component of thinned intensity $\lambda_p^{\mathcal{G}}(t)$, and $p\mathcal{B}^{-1}g(t;\theta) = g(t;\mathcal{A}\theta)$, then the M-estimator on thinned history can be written as $\tilde{\theta}_{\mathcal{H}} = \mathcal{A}^{-1}\hat{\theta}_{\mathcal{G}}$ such that $\mathbb{E}\big[\nabla R(\tilde{\theta}_{\mathcal{H}})|\mathcal{G}\big] \xrightarrow{P} 0$, as the number of realizations $n \to \infty$. Particularly, if $\lambda^{\mathcal{H}}(t;\theta)$ is linear, then $\mathcal{A} = p\mathcal{B}^{-1}$.

Proof. The proof is similar with the NHPP one. Be definition we have,

$$\begin{aligned}\mathbb{E}\big[\nabla R(\tilde{\theta}_{\mathcal{H}})|\mathcal{G}\big] &= \frac{1}{p}\mathbb{E}\int_0^T \Big\{H^{\mathcal{H}}(t;\tilde{\theta}_{\mathcal{H}})\big[dN_p(t) - p\lambda^{\mathcal{H}}(t;\tilde{\theta}_{\mathcal{H}})dt\big]\Big|\mathcal{G}\Big\} \\ &= \frac{1}{p}\int_0^T \mathbb{E}\Big\{H^{\mathcal{H}}(t;\tilde{\theta}_{\mathcal{H}})|\mathcal{G}\Big\}\,\mathbb{E}\Big\{dN_p(t) - p\lambda^{\mathcal{H}}(t;\tilde{\theta}_{\mathcal{H}})dt\Big|\mathcal{G}\Big\}\end{aligned}$$

By the definition of stochastic integral, it suffices to show that $N_p(t) - \int p\lambda^{\mathcal{H}}(t;\tilde{\theta}_{\mathcal{H}})dt$ asymptotically converges to a martingale in probability.

$$\begin{aligned}\mathbb{E}\Big\{dN_p(t) - p\lambda^{\mathcal{H}}(t;\tilde{\theta}_{\mathcal{H}})dt\Big|\mathcal{G}\Big\} &= g(t;\theta^*)^T m^{\mathcal{G}}(t) - pg(t;\tilde{\theta}_{\mathcal{H}})^T\mathbb{E}\left[m^{\mathcal{H}}(t)|\mathcal{G}\right] \\ &= \Big[g(t;\theta^*) - p\mathcal{B}^{-1}g(t;\tilde{\theta}_{\mathcal{H}})\Big]^T m^{\mathcal{G}}(t) \\ &= \Big[g(t;\theta^*) - g(t;\mathcal{A}\tilde{\theta}_{\mathcal{H}})\Big]^T m^{\mathcal{G}}(t) \\ &= \Big[g(t;\theta^*) - g(t;\hat{\theta}_{\mathcal{G}})\Big]^T m^{\mathcal{G}}(t)\end{aligned}$$

Since $\hat{\theta}_{\mathcal{G}} \xrightarrow{P} \theta^*_{\mathcal{G}}$ as the number of realizations $n \to \infty$, and g is continuous with respect to θ, $\Big[g(t;\theta^*) - g(t;\hat{\theta}_{\mathcal{G}})\Big]^T m^{\mathcal{G}}(t) \xrightarrow{P} 0$, which is the desired result. □

B.9 Proof of Theorem 7.8

Theorem 7.8 (Thinning for gradient estimation). Let $N(t)$ be a point process with decouplable intensity $\lambda^{\mathcal{H}}(t;\theta) = g(t;\theta)^T m^{\mathcal{H}}(t)$ in Eq. (4). If there exist invertible linear operators $\mathcal{A}$ and $\mathcal{B}$ satisfying $\mathcal{B}\mathbb{E}\big[m^{\mathcal{H}}(t)|\mathcal{G}\big] = m_p^{\mathcal{G}}(t)$, where $m_p^{\mathcal{G}}(t)$ is the component of thinned intensity $\lambda_p^{\mathcal{G}}(t)$, and $p\mathcal{B}^{-1}g(t;\theta) = g(t;\mathcal{A}\theta)$, then

(1) $\mathbb{E}\big[\nabla R(\theta)|\mathcal{G}\big] \le 1/p\mathcal{A}^{-1}\nabla R_p(\mathcal{A}\theta)$, for R is LSE;

(2) $\mathbb{E}\big[\nabla R(\theta)|\boldsymbol{\mathcal{G}}\big] \leq \mathcal{A}^{-1}\nabla R_p(\mathcal{A}\theta)$, for R is MLE.

Particularly, if the intensity is deterministic, i.e., $m^{\boldsymbol{\mathcal{H}}}(t) = 1$, both equalities hold.

Proof. By definition of stochastic integral, we have

$$\begin{aligned}\mathbb{E}\big[\nabla R(\theta)|\boldsymbol{\mathcal{G}}\big] &= \mathbb{E}\left\{\int_0^T H^{\boldsymbol{\mathcal{H}}}(t;\theta)\left[dN(t) - \lambda^{\boldsymbol{\mathcal{H}}}(t;\theta)dt\right]\Big|\boldsymbol{\mathcal{G}}\right\} \\ &= \mathbb{E}\left\{\int_0^T H^{\boldsymbol{\mathcal{H}}}(t;\theta)dN(t)\Big|\boldsymbol{\mathcal{G}}\right\} - \mathbb{E}\left\{\int_0^T H^{\boldsymbol{\mathcal{H}}}(t;\theta)\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta)dt\Big|\boldsymbol{\mathcal{G}}\right\}\end{aligned}$$

Here the second term can be bounded by,

$$\mathbb{E}\left\{H^{\boldsymbol{\mathcal{H}}}(t;\theta)\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta)dt\big|\boldsymbol{\mathcal{G}}\right\} \geq \mathbb{E}\left\{H^{\boldsymbol{\mathcal{H}}}(t;\theta)\big|\boldsymbol{\mathcal{G}}\right\}\mathbb{E}\left\{\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta)dt\big|\boldsymbol{\mathcal{G}}\right\}$$

According to the definition of forward stochastic integral, the first term can be written as,

$$\mathbb{E}\left\{\int_0^T H^{\boldsymbol{\mathcal{H}}}(t;\theta)dN(t)|\boldsymbol{\mathcal{G}}\right\} = \int_0^T \mathbb{E}\left\{H^{\boldsymbol{\mathcal{H}}}(t;\theta)|\boldsymbol{\mathcal{G}}\right\}\mathbb{E}\left\{dN(t)|\boldsymbol{\mathcal{G}}\right\}$$

Let's look at these components one by one. The condition of the theorem yields,

$$\begin{aligned}\mathbb{E}\left\{\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta)dt\big|\boldsymbol{\mathcal{G}}\right\} &= g(t;\theta)^T\mathbb{E}\big[m^{\boldsymbol{\mathcal{H}}}(t)|\boldsymbol{\mathcal{G}}\big]dt \\ &= p\mathcal{B}^{-1}g(t;\theta)^T m^{\boldsymbol{\mathcal{G}}}(t)dt \\ &= g(t;\mathcal{A}\theta)^T m^{\boldsymbol{\mathcal{G}}}(t)dt \\ &= \lambda^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)dt \qquad \text{(B.1)}\end{aligned}$$

and,

$$\mathbb{E}\left\{dN(t)|\boldsymbol{\mathcal{G}}\right\} = \frac{1}{p}dN_p(t).$$

If R is LSE, then we have,

$$\begin{aligned}\mathbb{E}\Big[H^{\boldsymbol{\mathcal{H}}}(t;\theta)\big|\boldsymbol{\mathcal{G}}\Big] &= \nabla\mathbb{E}\Big[\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta)|\boldsymbol{\mathcal{G}}\Big] \\ &= \nabla_\theta\frac{1}{p}g(t;\mathcal{A}\theta)^T m_p^{\boldsymbol{\mathcal{G}}}(t) \\ &= \mathcal{A}^{-1}\nabla\frac{1}{p}g(t;\mathcal{A}\theta)^T m_p^{\boldsymbol{\mathcal{G}}}(t) \\ &= \mathcal{A}^{-1}H_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta) \qquad \text{(B.2)}\end{aligned}$$

Thus, combining Eq.(B.1),(B.9) and (B.2) yields,

$$\begin{aligned}\mathbb{E}\big[\nabla R(\theta)|\boldsymbol{\mathcal{G}}\big] &\leq \frac{1}{p^2}\mathcal{A}^{-1}\int_0^T H_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)\big[dN_p(t)-\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)dt\big]\\ &= \frac{1}{p}\mathcal{A}^{-1}\nabla R_p(\mathcal{A}\theta).\end{aligned}$$

If R is MSE, then we have,

$$\begin{aligned}\mathbb{E}\Big[H^{\boldsymbol{\mathcal{H}}}(t;\theta)\big|\boldsymbol{\mathcal{G}}\Big] &= \nabla\mathbb{E}\Big[\log\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta)|\boldsymbol{\mathcal{G}}\Big]\\ &\geq \nabla_\theta\log\Big[g(t;\mathcal{A}\theta)^T m_p^{\boldsymbol{\mathcal{G}}}(t)\Big]\\ &= \mathcal{A}^{-1}\nabla\log\Big[g(t;\mathcal{A}\theta)^T m_p^{\boldsymbol{\mathcal{G}}}(t)\Big]\\ &= \mathcal{A}^{-1}H_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)\end{aligned}\tag{B.3}$$

Combining Eq.(B.1) and Eq.(B.3) yields the second conclusion,

$$\begin{aligned}\mathbb{E}\big[\nabla R(\theta)|\boldsymbol{\mathcal{G}}\big] &\leq \frac{1}{p}\mathcal{A}^{-1}\int_0^T H_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)\big[dN_p(t)-\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)dt\big]\\ &= \mathcal{A}^{-1}\nabla R_p(\mathcal{A}\theta).\end{aligned}$$

The proof ends here. □

B.10 Proof of Theorem 7.9

Theorem 7.9 (Variance of gradient estimation). Let $\nabla\tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta)$ and $\nabla R_\ell(\theta)$ be the p-thinned and sub-interval gradient at θ, where $\nabla\tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta) = 1/p\mathcal{A}^{-1}\nabla R_p(\mathcal{A}\theta)$ for LSE and $\nabla\tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta) = \mathcal{A}^{-1}\nabla R_p(\mathcal{A}\theta)$ for MLE. The variance of p-thinned gradient is no greater than that of sub-interval gradient:

$$\mathbb{V}\Big[\nabla\tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta)\Big] \leq \mathbb{V}\Big[\nabla R_\ell(\theta)\Big].\tag{B.3}$$

Proof. For the RHS, using the law of total variance yields,

$$\mathbb{V}\Big[\nabla R_\ell(\theta)\Big] = \mathbb{E}\Big\{\mathbb{V}\big[\nabla R_\ell(\theta)|\boldsymbol{\mathcal{F}}\big]\Big\} + \mathbb{V}\Big\{\mathbb{E}\big[\nabla R_\ell(\theta)|\boldsymbol{\mathcal{F}}\big]\Big\}.$$

The first term can be rewritten as,

$$\begin{aligned}\mathbb{E}\Big\{\mathbb{V}\big[\nabla R_\ell(\theta)|\boldsymbol{\mathcal{F}}\big]\Big\} &= \frac{1-p}{p}\mathbb{E}\left\{\int_0^T H^{\boldsymbol{\mathcal{H}}}(t;\theta)\big[dN(t)-\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta)dt\big]\right\}^2 \\ &= \frac{1-p}{p}\mathbb{E}\Big[\nabla R(\theta)\Big]^2,\end{aligned}$$

The second term can be written as,

$$\mathbb{V}\Big\{\mathbb{E}\big[\nabla R_\ell(\theta)|\boldsymbol{\mathcal{F}}\big]\Big\} = \mathbb{V}\big[\nabla R(\theta)\big].$$

Thus, the total variance of $\nabla R_\ell(\theta)$ can be written as,

$$\mathbb{V}\Big[\nabla R_\ell(\theta)\Big] = \frac{1-p}{p}\mathbb{E}\Big[\nabla R(\theta)\Big]^2 + \mathbb{V}\big[\nabla R(\theta)\big].$$

Then we consider the LHS, by the definition of variance,

$$\mathbb{V}\Big[\nabla \tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta)\Big] = \mathbb{E}\Big[\nabla \tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta)\Big]^2 - \Big[\mathbb{E}\nabla \tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta)\Big]^2. \tag{B.3}$$

Apply Theorem 5.1, we have,

$$\Big[\mathbb{E}\nabla \tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta)\Big]^2 \geq \Big[\mathbb{E}\nabla R(\theta)\Big]^2. \tag{B.3}$$

For LSE, since quadratic function is convex, we obtain $\mathbb{E}\big[H_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)\big]^2 \leq \mathbb{E}\big[H^{\boldsymbol{\mathcal{H}}}(t;\theta)\big]^2$. This equivalence also holds for MLE, we omit the proof, since it can be proved similarly. Further, we obtain,

$$\begin{aligned}&\mathbb{E}\Big[\nabla \tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta)\Big]^2 \\ =&\frac{1}{p^2}\mathcal{A}^{-1}\mathbb{E}\Big[\int_0^T H_p^{\boldsymbol{\mathcal{G}}}(t;(\mathcal{A}\theta)\left[dN_p(t)-\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)dt\right]\Big]^2(\mathcal{A}^{-1})^T \\ =&\frac{1}{p^2}\mathcal{A}^{-1}\mathbb{E}\Big[\int_0^T H_p^{\boldsymbol{\mathcal{G}}}(t;(\mathcal{A}\theta)\left[dN_p(t)-\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\theta^*_{\boldsymbol{\mathcal{G}}})dt+\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\theta^*_{\boldsymbol{\mathcal{G}}})dt-\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)dt\right]\Big]^2(\mathcal{A}^{-1})^T \\ =&\frac{1}{p^2}\mathcal{A}^{-1}\mathbb{E}\left\{\int_0^T H_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)\left[dN_p(t)-\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\theta^*_{\boldsymbol{\mathcal{G}}})dt\right]\right\}^2(\mathcal{A}^{-1})^T + \\ &\frac{1}{p^2}\mathcal{A}^{-1}\mathbb{E}\left\{\int_0^T H_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)\left[\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta^*_{\boldsymbol{\mathcal{G}}})-\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)\right]dt\right\}^2(\mathcal{A}^{-1})^T\end{aligned} \tag{B.4}$$

The first term,

$$\begin{aligned}
&\frac{1}{p^2}\mathcal{A}^{-1}\mathbb{E}\left\{\int_0^T H_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)\left[dN_p(t)-\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\theta_{\boldsymbol{\mathcal{G}}}^*)dt\right]\right\}^2(\mathcal{A}^{-1})^T\\
=&\frac{1}{p^2}\mathcal{A}^{-1}\mathbb{E}\left\{\int_0^T\left[H_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)\right]^2 dN_p(t)\right\}(\mathcal{A}^{-1})^T\\
\leq&\frac{1}{p^2}\mathbb{E}\left\{\int_0^T\left[H^{\boldsymbol{\mathcal{H}}}(t;\theta)\right]^2 dN_p(t)\right\}\\
=&\frac{1}{p}\mathbb{E}\int_0^T\left[H^{\boldsymbol{\mathcal{H}}}(t;\theta)\right]^2 dN(t)
\end{aligned}\tag{B.5}$$

The second term,

$$\begin{aligned}
&\frac{1}{p^2}\mathcal{A}^{-1}\mathbb{E}\left\{\int_0^T H_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)\left[\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta_{\boldsymbol{\mathcal{G}}}^*)-\lambda_p^{\boldsymbol{\mathcal{G}}}(t;\mathcal{A}\theta)\right]dt\right\}^2(\mathcal{A}^{-1})^T\\
\leq&\frac{1}{p}\mathbb{E}\left\{\int_0^T H^{\boldsymbol{\mathcal{H}}}(t;\theta)\left[\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta_{\boldsymbol{\mathcal{H}}}^*)-\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta)\right]dt\right\}^2
\end{aligned}\tag{B.6}$$

Substituting Eq. (B.5) and (B.6) to Eq. (B.4) yields

$$\begin{aligned}
\mathbb{E}\Big[\nabla\tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta)\Big]^2 &\leq \frac{1}{p}\mathbb{E}\int_0^T\left[H^{\boldsymbol{\mathcal{H}}}(t;\theta)\right]^2 dN(t)+\frac{1}{p}\mathbb{E}\left\{\int_0^T H^{\boldsymbol{\mathcal{H}}}(t;\theta)\left[\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta_{\boldsymbol{\mathcal{H}}}^*)-\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta)\right]dt\right\}^2\\
&=\frac{1}{p}\mathbb{E}\left\{\int_0^T H^{\boldsymbol{\mathcal{H}}}(t;\theta)\left[dN(t)-\lambda^{\boldsymbol{\mathcal{H}}}(t;\theta_{\boldsymbol{\mathcal{H}}}^*)dt\right]\right\}^2\\
&=\mathbb{E}\Big[\nabla R(\theta)\Big]^2.
\end{aligned}\tag{B.7}$$

Combine Eq.(B.7) and Eq.(B.4) to Eq.(B.10),

$$\begin{aligned}
\mathbb{V}\Big[\nabla\tilde{R}^{\boldsymbol{\mathcal{G}}}(\theta)\Big] &\leq \frac{1}{p}\mathbb{E}\Big[\nabla R(\theta)\Big]^2-\Big[\mathbb{E}\nabla R(\theta)\Big]^2\\
&=\frac{1-p}{p}\mathbb{E}\Big[\nabla R(\theta)\Big]^2+\mathbb{E}\Big[\nabla R(\theta)\Big]^2-\Big[\mathbb{E}\nabla R(\theta)\Big]^2\\
&=\frac{1-p}{p}\mathbb{E}\Big[\nabla R(\theta)\Big]^2+\mathbb{V}\Big[\nabla R(\theta)\Big]\\
&=\mathbb{V}\Big[\nabla R_\ell(\theta)\Big],
\end{aligned}$$

which is the desired result. □

List of Author's Publications

- **Tianbo Li**, Tianze Luo, Yiping Ke, Sinno Jialin Pan. *Learning Complex Stochastic Systems via Transformer*. (Under review).
- **Tianbo Li**, Tianze Luo, Yiping Ke, Sinno Jialin Pan. *Graph Convolutional Hawkes Processes for Learning Attributed Event Sequences*. (Under review).
- **Tianbo Li**, Yiping Ke, Pengfei Wei. *Network Transfer for Hawkes Processes*. (Under review).
- **Tianbo Li**, Yiping Ke. (2020). *Tweedie-Hawkes Processes: Interpreting the Phenomena of Outbreaks*. The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20). New York, USA.
- **Tianbo Li**, Yiping Ke. (2019). *Thinning for Accelerating the Learning of Point Processes*, Advances in Neural Information Processing Systems (NeurIPS-19). Vancouver, Canada.
- **Tianbo Li**, Pengfei Wei, Yiping Ke. (2018). *Transfer Hawkes Processes with Content Information*, the 2018 IEEE International Conference on Data Mining (ICDM-18), Singapore.

Bibliography

[1] Tianbo Li and Yiping Ke. Thinning for accelerating the learning of point processes. In *Advances in Neural Information Processing Systems*, volume 32, pages 4091–4101. Curran Associates, Inc., 2019. iii, 30

[2] Tianbo Li and Yiping Ke. Tweedie-hawkes processes: Interpreting the phenomena of outbreaks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4699–4706, Apr. 2020. URL `https://ojs.aaai.org/index.php/AAAI/article/view/5902`. iii, 53

[3] Yiping Ke Sinno Jialin Pan Tianbo Li, Tianze Luo. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2021. iii, 75

[4] Tianbo Li and Yiping Ke. Thinning for accelerating the learning of point processes. In *Advances in Neural Information Processing Systems*, pages 4091–4101, 2019. iv, 91

[5] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *WWW'10*. ACM, 2010. xiii, 31

[6] Manlio De Domenico, Antonio Lima, Paul Mougel, and Mirco Musolesi. The anatomy of a scientific rumor. *Scientific reports*, 2013. xiii, 31

[7] Piet De Jong, Gillian Z Heller, et al. Generalized linear models for insurance data. *Cambridge Books*, 2008. xiii, 31

[8] Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. Time-sensitive recommendation from recurrent user activities. In *NIPS'15*, 2015. 1, 24

[9] Yichen Wang, Nan Du, Rakshit Trivedi, and Le Song. Coevolutionary latent feature processes for continuous-time user-item interactions. In *Advances in Neural Information Processing Systems*, pages 4547–4555, 2016. 23

[10] Hanjun Dai, Yichen Wang, Rakshit Trivedi, and Le Song. Recurrent coevolutionary latent feature processes for continuous-time recommendation. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pages 29–34. ACM, 2016.

[11] Wenming Xiao, Xiao Xu, Kang Liang, Junkang Mao, and Jun Wang. Job recommendation with hawkes process: an effective solution for recsys challenge 2016. In *Proceedings of the Recommender Systems Challenge*, page 11. ACM, 2016. 27

[12] Seyed Abbas Hosseini, Keivan Alizadeh, Ali Khodadadi, Ali Arabzadeh, Mehrdad Farajtabar, Hongyuan Zha, and Hamid R. Rabiee. Recurrent poisson factorization for temporal recommendation. In *KDD '17*, pages 847–855, New York, NY, USA, 2017. ACM.

[13] Hongteng Xu, Dixin Luo, Xu Chen, and Lawrence Carin. Benefits from superposed hawkes processes. *arXiv preprint arXiv:1710.05115*, 2017. 1, 23, 24

[14] Baichuan Yuan, Hao Li, Andrea L Bertozzi, P Jeffrey Brantingham, and Mason A Porter. Multivariate spatiotemporal hawkes processes and network reconstruction. *SIAM Journal on Mathematics of Data Science*, 1(2):356–382, 2019. 1

[15] Jiancang Zhuang and Jorge Mateu. A semiparametric spatiotemporal hawkes-type point process model with periodic background for crime data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 182(3):919–942, 2019.

[16] Pengfei Wang, Yanjie Fu, Guannan Liu, Wenqing Hu, and Charu Aggarwal. Human mobility synchronization and trip purpose detection with mixture of hawkes processes. In *KDD'17*, 2017. 1, 21

[17] Long Tran, Mehrdad Farajtabar, Le Song, and Hongyuan Zha. Netcodec: Community detection from individual activities. In *SDM'15*, 2015. 1, 19, 21, 23, 98

[18] Charles Blundell, Jeff Beck, and Katherine A Heller. Modelling reciprocating relationships with hawkes processes. In *NIPS'12*, 2012. 20, 21

[19] Seyed Abbas Hosseini, Ali Khodadadi, Ali Arabzadeh, and Hamid R Rabiee. Hnp3: A hierarchical nonparametric point process for modeling content diffusion over social media. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 943–948. IEEE, 2016. 23

[20] Dixin Luo, Hongteng Xu, Yi Zhen, Xia Ning, Hongyuan Zha, Xiaokang Yang, and Wenjun Zhang. Multi-task multi-dimensional hawkes processes for modeling event sequences. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015. 23, 83, 103

[21] Shuang-Hong Yang and Hongyuan Zha. Mixture of mutually exciting processes for viral diffusion. In *ICML'13*, 2013. 21, 23

[22] Liangda Li, Hongbo Deng, Anlei Dong, Yi Chang, and Hongyuan Zha. Identifying and labeling search tasks via query-based hawkes processes. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 731–740. ACM, 2014.

[23] Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *KDD'15*. ACM, 2015. 20, 21, 28

[24] Hongteng Xu and Hongyuan Zha. A dirichlet mixture model of hawkes processes for event sequence clustering. In *NIPS'17*, 2017. 2, 21, 28, 60, 108

[25] Xinran He, Theodoros Rekatsinas, James Foulds, Lise Getoor, and Yan Liu. Hawkestopic: A joint model for network inference and topic modeling from text-based cascades. In *ICML'15*, 2015. 21, 46

[26] Wanying Ding, Yue Zhang, Chaomei Chen, and Xiaohua Hu. Semi-supervised dirichlet-hawkes process with applications of topic detection and tracking in twitter. In *Big Data (Big Data), 2016 IEEE International Conference on*, pages 869–874. IEEE, 2016. 1, 23

[27] Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *ICML'16*, 2016. 2, 3, 4, 19, 20, 23, 50, 70, 82, 84, 85, 91, 92, 94, 98, 101, 102

[28] Yichen Wang, Xiaojing Ye, Haomin Zhou, Hongyuan Zha, and Le Song. Linking micro event history to macro prediction in point process models. In *Artificial Intelligence and Statistics*, pages 1375–1384, 2017. 2, 23, 27

[29] Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional hawkes processes. In *International Conference on Machine Learning*, pages 1301–1309, 2013. 3, 4, 20, 91, 92, 94, 101, 102

[30] Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional hawkes processes. In *International Conference on Machine Learning*, pages 1301–1309, 2013. 3, 4, 82, 91, 98, 101, 102

[31] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010. 4, 53, 55

[32] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le 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*, pages 1555–1564. ACM, 2016. 4, 21, 22, 75, 76, 77, 79, 81, 84

[33] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen M Chu. Modeling the intensity function of point process via recurrent neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 22, 76, 83, 84

[34] Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *NIPS'17*, 2017. 21, 22, 76, 77, 79, 81, 84

[35] Takahiro Omi, Naonori Ueda, and Kazuyuki Aihara. Fully neural network based model for general temporal point processes. In *Advances in Neural Information Processing Systems*, 2019. 4, 22, 75, 76

[36] James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. Quasi-recurrent neural networks. *arXiv preprint arXiv:1611.01576*, 2016. 4, 75

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 4, 75

[38] Daryl J Daley and David Vere-Jones. An introduction to the theory of point processes, volume 1: Elementary theory and methods. *Verlag New York Berlin Heidelberg: Springer*, 2003. 11, 69

[39] Alex Reinhart et al. A review of self-exciting spatio-temporal point processes and their applications. *Statistical Science*, 33(3):299–318, 2018. 11

[40] Alan G Hawkes and David Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, 11(3):493–503, 1974. 12, 40

[41] Emmanuel Bacry, Sylvain Delattre, Marc Hoffmann, and Jean-Francois Muzy. Some limit theorems for hawkes processes and application to financial statistics. *Stochastic Processes and their Applications*, 123(7):2475–2499, 2013. 13

[42] Alexandre Boumezoued. Population viewpoint on hawkes processes. *Advances in Applied Probability*, 48(2):463–480, 2016. 14

[43] Per K Andersen, Ornulf Borgan, Richard D Gill, and Niels Keiding. *Statistical models based on counting processes*. Springer Science & Business Media, 2012. 14, 15, 43, 94, 97, 126

[44] Remi Lemonnier and Nicolas Vayatis. Nonparametric markovian learning of triggering kernels for mutually exciting and mutually inhibiting multivariate hawkes processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 161–176. Springer, 2014. 19

[45] Liangda Li and Hongyuan Zha. Learning parametric models for social infectivity in multi-dimensional hawkes processes. In *AAAI'14*, 2014. 20, 21, 49, 92, 98

[46] Aleksandr Simma and Michael I. Jordan. Modeling events with cascades of poisson processes. UAI 10, pages 546–555, 2010.

[47] Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional hawkes processes. In *ICML'13*, 2013. 50, 70, 82, 85, 98

[48] Ali Zarezade, Ali Khodadadi, Mehrdad Farajtabar, Hamid R Rabiee, and Hongyuan Zha. Correlated cascades: compete or cooperate. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 238–244, 2017. 19, 23

[49] Spencer Wheatley, Vladimir Filimonov, and Didier Sornette. The hawkes process with renewal immigration & its estimation with an em algorithm. *Computational Statistics & Data Analysis*, 94:120–135, 2016. 20

[50] Young Lee, Kar Wai Lim, and Cheng Soon Ong. Hawkes processes with stochastic excitations. In *ICML'16*, 2016. 20

[51] Mehrdad Farajtabar, Xiaojing Ye, Sahar Harati, Le Song, and Hongyuan Zha. Multistage campaigning in social networks. In *Advances in Neural Information Processing Systems*, pages 4718–4726, 2016. 23, 26

[52] Erik Lewis, George Mohler, P Jeffrey Brantingham, and Andrea L Bertozzi. Self-exciting point process models of civilian deaths in iraq. *Security Journal*, 25(3):244–264, 2012.

[53] Wanying Ding, Yue Shang, Lifan Guo, Xiaohua Hu, Rui Yan, and Tingting He. Video popularity prediction by sentiment propagation via implicit network. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1621–1630, 2015. 23, 27

[54] Michael Eichler, Rainer Dahlhaus, and Johannes Dueck. Graphical modeling for multivariate hawkes processes with nonparametric link functions. *Journal of Time Series Analysis*, 38(2):225–242, 2017. 23, 25, 84

[55] Yichen Wang, Bo Xie, Nan Du, and Le Song. Isotonic hawkes processes. In *ICML'18*, 2016. 20

[56] Scott Linderman and Ryan Adams. Discovering latent network structure in point process data. In *ICML'14*, 2014. 20, 23, 66, 94

[57] Pierre Brémaud and Laurent Massoulié. Power spectra of general shot noises and hawkes point processes with a random excitation. *Advances in Applied Probability*, 34(1):205–222, 2002.

[58] Angelos Dassios and Hongbiao Zhao. A dynamic contagion process. *Advances in applied probability*, 43(3):814–846, 2011.

[59] Abir De, Isabel Valera, Niloy Ganguly, Sourangshu Bhattacharya, and Manuel Gomez Rodriguez. Learning and forecasting opinion dynamics in social networks. In *Advances in Neural Information Processing Systems*, pages 397–405, 2016. 20, 23

[60] Massil Achab, Emmanuel Bacry, Stéphane Gaïffas, Iacopo Mastromatteo, and Jean-François Muzy. Uncovering causality from multivariate hawkes integrated cumulants. *JRML*, 2017. 20, 82, 84, 85

[61] Charalampos Mavroforakis, Isabel Valera, and Manuel Gomez-Rodriguez. Modeling the dynamics of learning activity on the web. In *WWW'17*, 2017. 20

[62] Shuang Li, Shuai Xiao, Shixiang Zhu, Nan Du, Yao Xie, and Le Song. Learning temporal point processes via reinforcement learning. In *NIPS'18*, 2018. 21

[63] Rémi Lemonnier, Kevin Scaman, and Argyris Kalogeratos. Multivariate hawkes processes for large-scale inference. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 21

[64] Yingxiang Yang, Jalal Etesami, Niao He, and Negar Kiyavash. Online learning for multivariate hawkes processes. In *Advances in Neural Information Processing Systems*, pages 4937–4946, 2017. 21, 92

[65] Ce Guo and Wayne Luk. Accelerating maximum likelihood estimation for hawkes point processes. In *2013 23rd International Conference on Field programmable Logic and Applications*, pages 1–6. IEEE, 2013. 21

[66] Hongteng Xu, Xu Chen, and Lawrence Carin. Superposition-assisted stochastic optimization for hawkes processes. *arXiv preprint arXiv:1802.04725*, 2018. 21, 84, 92, 94, 96, 103

[67] Aleksandr Simma and Michael I Jordan. Modeling events with cascades of poisson processes. In *UAI'10*, 2010. 21

[68] Yeon Seonwoo, Alice Oh, and Sungjoon Park. Hierarchical dirichlet gaussian marked hawkes process for narrative reconstruction in continuous time domain. In *EMNLP'18*, 2018. 21, 28, 46, 50, 82

[69] Jin Shang and Mingxuan Sun. Geometric hawkes processes with graph convolutional recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4878–4885, 2019. 22, 76, 84

[70] Federico Monti, Michael Bronstein, and Xavier Bresson. Geometric matrix completion with recurrent multi-graph neural networks. In *Advances in Neural Information Processing Systems*, pages 3697–3707, 2017. 22

[71] Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *AISTATS*, pages 641–649, 2013. 23

[72] Nan Du, Le Song, Ming Yuan, and Alex J Smola. Learning networks of heterogeneous influence. In *Advances in Neural Information Processing Systems*, pages 2780–2788, 2012.

[73] Tomoharu Iwata, Amar Shah, and Zoubin 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*, pages 266–274. ACM, 2013.

[74] Mehrdad Farajtabar, Nan Du, Manuel Gomez Rodriguez, Isabel Valera, Hongyuan Zha, and Le Song. Shaping social activity by incentivizing users. In *Advances in neural information processing systems*, pages 2474–2482, 2014. 26, 27

[75] Martin Jankowiak and Manuel Gomez-Rodriguez. Uncovering the spatiotemporal patterns of collective social activity. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 822–830. SIAM, 2017.

[76] Jalal Etesami, Negar Kiyavash, Kun Zhang, and Kushagra Singhal. Learning network of multivariate hawkes processes: A time series approach. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16, pages 162–171, Arlington, Virginia, United States, 2016. AUAI Press. ISBN 978-0-9966431-1-5. URL `http://dl.acm.org/citation.cfm?id=3020948.3020966`.

[77] Ryota Kobayashi and Renaud Lambiotte. Tideh: Time-dependent hawkes process for predicting retweet dynamics. In *ICWSM*, pages 191–200, 2016. 23

[78] Mehrdad Farajtabar, Jiachen Yang, Xiaojing Ye, Huan Xu, Rakshit Trivedi, Elias Khalil, Shuang Li, Le Song, and Hongyuan Zha. Fake news mitigation via point process based intervention. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1097–1106. JMLR. org, 2017. 23, 26

[79] J.C. Louzada Pinto, T. Chahed, and E. Altman. A framework for information dissemination in social networks using hawkes processes. *Performance Evaluation*, 103(Supplement C):86 – 107, 2016. ISSN 0166-5316. Performance Evaluation Methodologies and Tools: Selected Papers from ValueTools 2014. 23

[80] Mehrdad Farajtabar, Yichen Wang, Manuel Gomez Rodriguez, Shuang Li, Hongyuan Zha, and Le Song. Coevolve: A joint point process model for information diffusion and network co-evolution. In *Advances in Neural Information Processing Systems*, pages 1954–1962, 2015. 23

[81] Vasanthan Raghavan, Greg Ver Steeg, Aram Galstyan, and Alexander G Tartakovsky. Modeling temporal activity patterns in dynamic social networks. *IEEE Transactions on Computational Social Systems*, 1(1):89–107, 2014. 23

[82] Qingyuan Zhao, Murat A Erdogdu, Hera Y He, Anand Rajaraman, and Jure Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1513–1522. ACM, 2015. 23, 26, 27

[83] Shuai Gao, Jun Ma, and Zhumin Chen. Modeling and predicting retweeting dynamics on microblogging platforms. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 107–116. ACM, 2015.

[84] Swapnil Mishra, Marian-Andrei Rizoiu, and Lexing Xie. Feature driven and point process approaches for popularity prediction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1069–1078. ACM, 2016. 26, 27

[85] Xin Liu, Junchi Yan, Shuai Xiao, Xiangfeng Wang, Hongyuan Zha, and Stephen M Chu. On predictive patent valuation: Forecasting patent citations and their types. In *AAAI*, pages 1438–1444, 2017. 27

[86] Marian-Andrei Rizoiu, Lexing Xie, Scott Sanner, Manuel Cebrian, Honglin Yu, and Pascal Van Hentenryck. Expecting to be hip: Hawkes intensity processes for social media popularity. In *Proceedings of the 26th International Conference on World Wide Web*, pages 735–744. International World Wide Web Conferences Steering Committee, 2017. 27

[87] Wang Yichen, Ye Xiaojing, Zha Hongyuan, and SongJ Le. Predicting user activity level in point processes with mass transport equation. In *Advances in Neural Information Processing Systems*, 2017. 27

[88] Wanying Ding, Yue Shang, Lifan Guo, Xiaohua Hu, Rui Yan, and Tingting He. Video popularity prediction by sentiment propagation via implicit network. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 1621–1630, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3794-6.

[89] Yi Wei, Ke Zhou, Ya Zhang, and Hongyuan Zha. Learning the hotness of information diffusions with multi-dimensional hawkes processes. In *Revised Selected Papers of the 9th International Workshop on Agents and Data Mining Interaction - Volume 8316*, ADMI 2013, pages 92–110, New York, NY, USA, 2014. Springer-Verlag New York, Inc. ISBN 978-3-642-55191-8. 23

[90] Vanessa Didelez. Graphical models for marked point processes based on local independence. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):245–264, 2008. 24

[91] LR Taylor. Aggregation, variance and the mean. *Nature*, 1961. 31, 33

[92] Bent Jorgensen. Exponential dispersion models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1987. 31, 33

[93] DG Harcourt. Population dynamics of leptinotarsa decemlineata (say) in eastern ontario: I. spatial pattern and transformation of field counts. *The Canadian Entomologist*, 1963. 32

[94] Agata Fronczak and Piotr Fronczak. Origins of taylor's power law for fluctuation scaling in complex systems. *Physical Review E*, 81(6):066112, 2010. 32

[95] Gal Oestreicher-Singer and Arun Sundararajan. Recommendation networks and the long tail of electronic commerce. *Mis quarterly*, pages 65–83, 2012. 32

[96] Stuart A Klugman, Harry H Panjer, and Gordon E Willmot. *Loss models: from data to decisions*. 2012. 32

[97] John Ashworth Nelder and R Jacob Baker. *Generalized linear models*. Wiley Online Library, 1972. 33

[98] MCK Tweedie. An index which distinguishes between some important exponential families. In *Statistics: Applications and new directions: Proc. Indian statistical institute golden Jubilee International conference*, volume 579, page 604, 1984. 33

[99] Harold P Benson. Fractional programming with convex quadratic forms and functions. *European Journal of Operational Research*, 2006. 38

[100] Mordecai Avriel. *Nonlinear programming: analysis and methods*. Courier Corporation, 2003. 39

[101] David Vere-Jones. *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*. Springer, 2003. 40, 43

[102] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD'09*. ACM, 2009. 46, 50

[103] Alan G Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 438–443, 1971. 49

[104] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'96*, 1996. 49

[105] Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *AISTATS'13*, 2013. 50

[106] Dixin Luo, Hongteng Xu, Hongyuan Zha, Jun Du, Rong Xie, Xiaokang Yang, and Wenjun Zhang. You are what you watch and when you watch: Inferring household structures from iptv viewing data. *IEEE Transactions on Broadcasting*, 2014. 50

[107] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. Learning geographical preferences for point-of-interest recommendation. In *KDD'13*. ACM. 50, 68, 83, 103

[108] Jakob Gulddahl Rasmussen. Bayesian inference for hawkes processes. *Methodology and Computing in Applied Probability*, 15(3):623–642, 2013. 64, 65

[109] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media, 2007. 66

[110] Yosihiko Ogata. On lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31, 1981. 66

[111] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. Exploiting geographical neighborhood characteristics for location recommendation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 739–748. ACM, 2014. 68

[112] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 76

[113] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International Conference on Machine Learning*, 2020. 76, 85

[114] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. 77, 78

[115] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016. 77

[116] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics*, June 2018. 80

[117] Jiawei Wu, Xin Wang, and William Yang Wang. Self-supervised dialogue learning. *arXiv preprint arXiv:1907.00448*, 2019. 80

[118] Wolfgang Jank and Galit Shmueli. *Modeling online auctions*, volume 91. John Wiley & Sons, 2010. 83

[119] PA W Lewis and Gerald S Shedler. Simulation of nonhomogeneous poisson processes by thinning. *Naval research logistics quarterly*, 26(3):403–413, 1979. 92

[120] Pierre Brémaud. *Point processes and queues: martingale dynamics*, volume 50. Springer, 1981. 93, 113

[121] Alan Karr. *Point Processes and Their Statistical Inference*, volume 7. CRC Press, 1991. 93, 94, 113, 125, 126

[122] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume I: Elementary theory and methods*. Springer Science & Business Media, 2002. 93, 113, 115

[123] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media, 2007. 93, 113

[124] Emmanuel Bacry, Stéphane Gaïffas, and Jean-François Muzy. A generalization error bound for sparse and low-rank multivariate hawkes processes. *arXiv preprint arXiv:1501.00725*, 2015. 94

[125] Tianbo Li, Pengfei Wei, and Yiping Ke. Transfer hawkes processes with content information. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1116–1121. IEEE, 2018. 94

[126] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 100

[127] Gordon K Smyth and Bent Jørgensen. Fitting tweedie's compound poisson model to insurance claims data: dispersion modelling. *ASTIN Bulletin: The Journal of the IAA*, 32(1):143–157, 2002. 117

[128] CF Jeff Wu. On the convergence properties of the em algorithm. *The Annals of statistics*, pages 95–103, 1983. 119

[129] Bo Wang and D Michael Titterington. Convergence and asymptotic normality of variational bayesian approximations for exponential family models with missing values. In *UAI'04*, 2004. 120

[130] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013. 123

[131] Cyrus R Mehta and Nitin R Patel. Exact logistic regression: theory and examples. *Statistics in medicine*, 14(19):2143–2160, 1995. 123