5-2018

Bayesian Nonparametrics to Model Content, User, and Latent Structure in Hawkes Processes

Xi Tan
Purdue University

BAYESIAN NONPARAMETRICS TO MODEL CONTENT, USER, AND LATENT STRUCTURE IN HAWKES PROCESSES

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Xi Tan

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2018

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Jennifer Neville, Chair

Department of Computer Science and Department of Statistics

Dr. Yuan (Alan) Qi

Department of Computer Science

Dr. Vinayak Rao

Department of Statistics

Dr. Ninghui Li

Department of Computer Science

Approved by:

Dr. Voicu Popescu by Dr. William J. Gorman

Head of the Graduate Program, Department of Computer Science

To my family.

邯郸冬至夜思家

邯郸驿里逢冬至　抱膝灯前影伴身。
想得家中夜深坐　还应说着远行人。

— 唐 · 白居易

ACKNOWLEDGMENTS

Many years later, as my grandchildren ask for Chrismas gifts, I am to remember that distant afternoon when my mom brought back home a used computer with a 50 MHz i486DX CPU and a 6 MB RAM. Ever since then, the curiosity and interest in computers never left me, only growing more profound and passionate with every new discovery I made.

Exploring the deep theories and philosophy of machine learning is an intellectually challenging and, for me, intrinsically fascinating activity. It is like working on a gigantic puzzle - one so large that it could occupy many lifetimes. Looking back on my academic career at Purdue, I have been improving myself everyday over the past years – full of passion; from reading research papers to applying them to my research projects, whether learning, working, or researching; it is my attentiveness and creativity all added to an overall dissertation that represent my intellectual ability.

However, this dissertation would not be possible without the encouragement and support from my advisor Prof. Jennifer Neville. She is the very type of advisor that I could ever have dreamt of – accessible, knowlegebale, and helpful in every aspect of one's Ph.D. life: from offering funding support to discussing research ideas, from polishing up our research papers to facilitating my graduation and dissertation writing. Words are more feeble to express how grateful and privileged I am.

Next, I thank my former advisor Prof. Yuan (Alan) Qi, who introduced to me the Bayesian world and the idea of Hawkes processes behind this dissertation. During the tough times along the course of my Ph.D. study, Alan provided with me unconditional support and always told me to persevere, overcome, and have faith. It is he who makes me even stronger and appreciate that true joy is a serious thing.

I would also like to thank Prof. Vinayak Rao, with whom I collaborated three research papers. Prof. Rao is knowledgeble, diligent, and easy to work with. I was fortunate to have worked with him in my final years as a Ph.D. student.

During my several memorable years at Purdue, I have met many people – some of them passing by, some of them are still in the vicinity of my life. It seems to me, somehow, life is like an Ornstein-Uhlenbeck process: mean-reverting, stochastic, risk of going default, and much more unpredictable in the future. I shall take this opportunity to thank the following people (far from exhaustive) for their accompany along the journey: Yunmei Bai, Run Chen, Bobby Il Yong Chun, Bo Dai, Liangzhi Dai, Wei Deng, Nan Ding, Youhan Fang, Sandra Freeman, Dr. Gorman, Pei He, Yayuan Hu, Amy Ingram, Ishita Khan, Yixuan Li, Fangjia Lu, Qianwen Luo, Renate Mallus-Medot, Hao Peng, Bin Shen, Danqing Shen, Yimeng Shi, Chunyan Sun, Ming Tang, Peter Waddell, Guanwen Wang, Mengjia Wang, Shipeng Wang, Yahui Wang, Wanyu Wu, Yiqiang Xie, Feng Yan, Jiasen Yang, Yifan Yang, Mo You, Peng Yu, Yongyang Yu, Xun Zha, Dan Zhang, Dongfang Zhang, Hanqing Zhang, Jiamin Zhang, Rongrong Zhang, Xiao Zhang, Zixu Zhang, Yiqi Zhao, Shandian Zhe, Yao Zhu.

Last but not least, this dissertation is dedicated to my family. You are the only reason I am, and you are all my reasons.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

SYMBOLS

$\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \boldsymbol{\Theta}$	Bold upper case letters represent matrices.
$\mathbf{x}, \mathbf{y}, \mathbf{z}, \boldsymbol{\theta}$	Bold lower case letters represent random variables.
$\mathcal{X}, \mathcal{Y}, \mathcal{Z}, \Theta$	Curly upper case letters represent support of random variables.
J	number of model parameters.
K	number of latent variables.
N	number of observations.
V	number of entities.
$\mathbf{y}_i$	the i^{th} observation, $i = 1, \ldots, N$.
$\mathbf{x}_i$	the i^{th} observed independent variable, $i = 1, \ldots, N$.
$\boldsymbol{\theta}_j$	the j^{th} model parameter, $j = 1, \ldots, J$.
$\mathbf{z}_k$	the k^{th} latent variable, $k = 1, \ldots, K$.

ABBREVIATIONS

DP	Dirichlet Process
GP	Gaussian Process
HP	Hawkes Process
CRP	Chinese Restaurant Process
GHP	Gaussian-Hawkes Process
IBP	Indian Buffet Process
MAP	Maximum A Priori
nCRP	Nested Chinese Restaurant Process
nCRP-GHP	nCRP-Gaussian-Hawkes Process
IBHP	Indian-Buffet-Hawkes Process

ABSTRACT

Tan, Xi PhD, Purdue University, May 2018. Bayesian Nonparametrics to Model Content, User, and Latent Structure in Hawkes Processes. Major Professor: Jennifer Neville.

Communication in social networks tends to exhibit complex dynamics both in terms of the users involved and the contents exchanged. For example, email exchanges or activities on social media may exhibit reinforcing dynamics, where earlier events trigger follow-up activity through multiple structured latent factors. Such dynamics have been previously represented using models of reinforcement and reciprocation, a canonical example being the Hawkes process (HP). However, previous HP models do not fully capture the rich dynamics of real-world activity. For example, reciprocation may be impacted by the significance and receptivity of the content being communicated, and modeling the content accurately at the individual level may require identification and exploitation of the latent hierarchical structure present among users. Additionally, real-world activity may be driven by multiple latent triggering factors shared by past and future events, with the latent features themselves exhibiting temporal dependency structures. These important characteristics have been largely ignored in previous work. In this dissertation, we address these limitations via three novel Bayesian nonparametric Hawkes process models, where the synergy between Bayesian nonparametric models and Hawkes processes captures the structural and the temporal dynamics of communication in a unified framework. Empirical results demonstrate that our models outperform competing state-of-the-art methods, by more accurately capturing the rich dynamics of the interactions and influences among users and events, and by improving predictions about future event times, user clusters, and latent features in various types of communication activities.

1. INTRODUCTION

1.1 Motivation

Communication in social networks tends to exhibit complex dynamics both in terms of the users involved and the contents exchanged, and quantifying this phenomenon has been a subject of long interest in the social science and machine learning communities [1–7]. For example, email exchanges or activities on social media may exhibit reinforcing dynamics, where earlier events trigger follow-up activity through multiple structured latent factors. Conversations or publications within specific communities are also featured by distinctive group patterns over time.

Such dynamics have been previously represented using models of reinforcement and reciprocation, a canonical example being the Hawkes process (HP) [8,9]. However, previous HP models do not fully capture the rich dynamics of real-world activity. For example, reciprocation may be impacted by the significance and receptivity of the content being communicated, and modeling the content accurately at the individual level may require identification and exploitation of the latent hierarchical structure present among users.

Additionally, real-world activity may be driven by multiple latent triggering factors shared by past and future events, with the latent features themselves exhibiting temporal dependency structures. For example, the ideas in a research paper may be derived from multiple existing works in the literature, each of which contributes one or more factors, with only their combination serving to trigger the event. Similarly, a conversation among individuals may heat up or cool down due to the topics being discussed (e.g., politics vs. weather). Moreover, individual check-in data on platforms like Foursquare or Yelp may depend on combinations of characteristics and activities from previous visited locations. Finally in biological data, pathways are often only

activated when a set of genes is expressed together. These important characteristics have been largely ignored in previous work.

In this dissertation, we address these limitations via three novel Bayesian nonparametric HP models, where the synergy between Bayesian nonparametric models and HP captures the structural and the temporal dynamics of the communication data in a unified framework.

1.2 Problem Statement

Formally, we consider an underlying network where nodes denote interacting entities and links represent communication events taking place in the network. The observed data $\mathcal{D}$ consists of a sequence of N communication observations

$$\mathcal{D} = (\mathbf{y}_1, \cdots, \mathbf{y}_N). \tag{1.1}$$

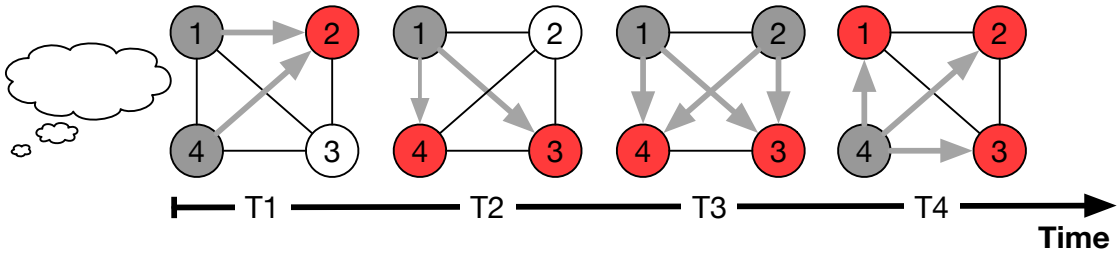
Each $\mathbf{y}_i$ contains the information of the time, senders, receivers, and contents of the i^{th} communication, and is denoted as a quadruplet

$$\mathbf{y}_i = (t_i, S_i, R_i, \mathcal{T}_i), \quad i = 1, \ldots, N, \tag{1.2}$$

where t_i is the time-stamp, S_i the set of senders, R_i the set of receivers, and $\mathcal{T}_i$ the contents of the communication.

Figure 1.1 shows an illustration of the abstract data format. For example, at time T2, it could be the case when individual 1 is sending an email to individuals 3 and 4, where the email text is the content of this communication; at time T3, it could be the case when a publication written by individuals 1 and 2 cites another publication written by individuals 3 and 4, where the publication text written by individuals 1 and 2 is the content of this communication. Data from many other scenarios can be easily adapted to this abstract data format.

Different models may use different subsets of this data format for various purposes, although ignoring any important aspects of the data may result in different levels of modeling ineffectiveness. For example, one may want to use $\mathbf{y}_i = \{t_i, S_i, R_i\}$ to model



Time	T1	T2	T3	T4
Senders	1, 4	1	1, 2	4
Receivers	2	3, 4	3, 4	1, 2, 3
Content	M1	M2	M3	M4

Fig. 1.1. Abstract data format. The datum for each event contains the time, senders, receivers, and contents of the communication.

count data when content information is not available; or use $\mathbf{y}_i = \{t_i, S_i, \mathcal{T}_i\}$ to model activity data where the recipient of the activity is not of interest.

Based on the communication history, we are interested in the task of predicting future events in the following three aspects in a unified framework:

1. Capture temporal intensities of events to infer the pattern of their dynamics.
2. Identify senders and receivers to learn their individual characteristics.
3. Model latent content features to discover interesting underlying representation of contents being communicated.

1.3 Approach and Outcomes

To this end, we propose three novel Bayesian nonparametric HP models. Namely, we propose the Gaussian-Hawkes Process (GHP) [10] to better capture the temporal intensities of events based on the contents that have been communicated, the nCRP-Gaussian-Hawkes Process (nCRP-GHP) [11] to identify senders and receivers based on the underlying hierarchical structure of individuals inferred from the data, and the

Indian-Buffet-Hawkes Process (IBHP) [12] to capture the dependency between the temporal and textual dynamics of the communication in order to discover interesting latent content features. Figure 1.2 illustrates the unified framework and the three model components.

1. **The Gaussian-Hawkes Process (GHP)**. In the first component of the dissertation, we extend [13] by introducing Gaussian processes (GPs) into the Hawkes IRM model, where the GPs are used to model the message significance as well as receptivity, allowing us to more accurately capture the interactions among entities. The application of GPs also enables us to flexibly model the rates of reciprocal activities between two entities, allowing asymmetry in reciprocity to be captured more accurately. This leads to better cluster detection capability.

2. **The nCRP-Gaussian-Hawkes Process (nCRP-GHP)**. In the second component of the dissertation, we propose a novel nonparametric Bayesian model that incorporates senders and receivers of messages into a hierarchical structure that governs the content and reciprocity of communications. We bring the nested Chinese restaurant process (nCRP) from nonparametric Bayesian statistics to HP models of point pattern data. By modeling senders and receivers in such a hierarchical framework, we are better able to make inferences about, more than cluster membership but, the individual authorship and audience of communications, as well as personal behavior such as favorite collaborators and top-pick words.

3. **The Indian-Buffet-Hawkes Process (IBHP)**. In the third component of the dissertation, we propose a novel Bayesian nonparametric stochastic point process model, the Indian Buffet Hawkes Processes (IBHP), that synergizes ideas between the Indian buffet process (IBP) and the HP. The use of the IBP to add multiple triggering factors to the HP helps better model dynamics and improves interpretation, and embedding the temporal information from the HP into the IBP expands its capability to model factor evolution over time.

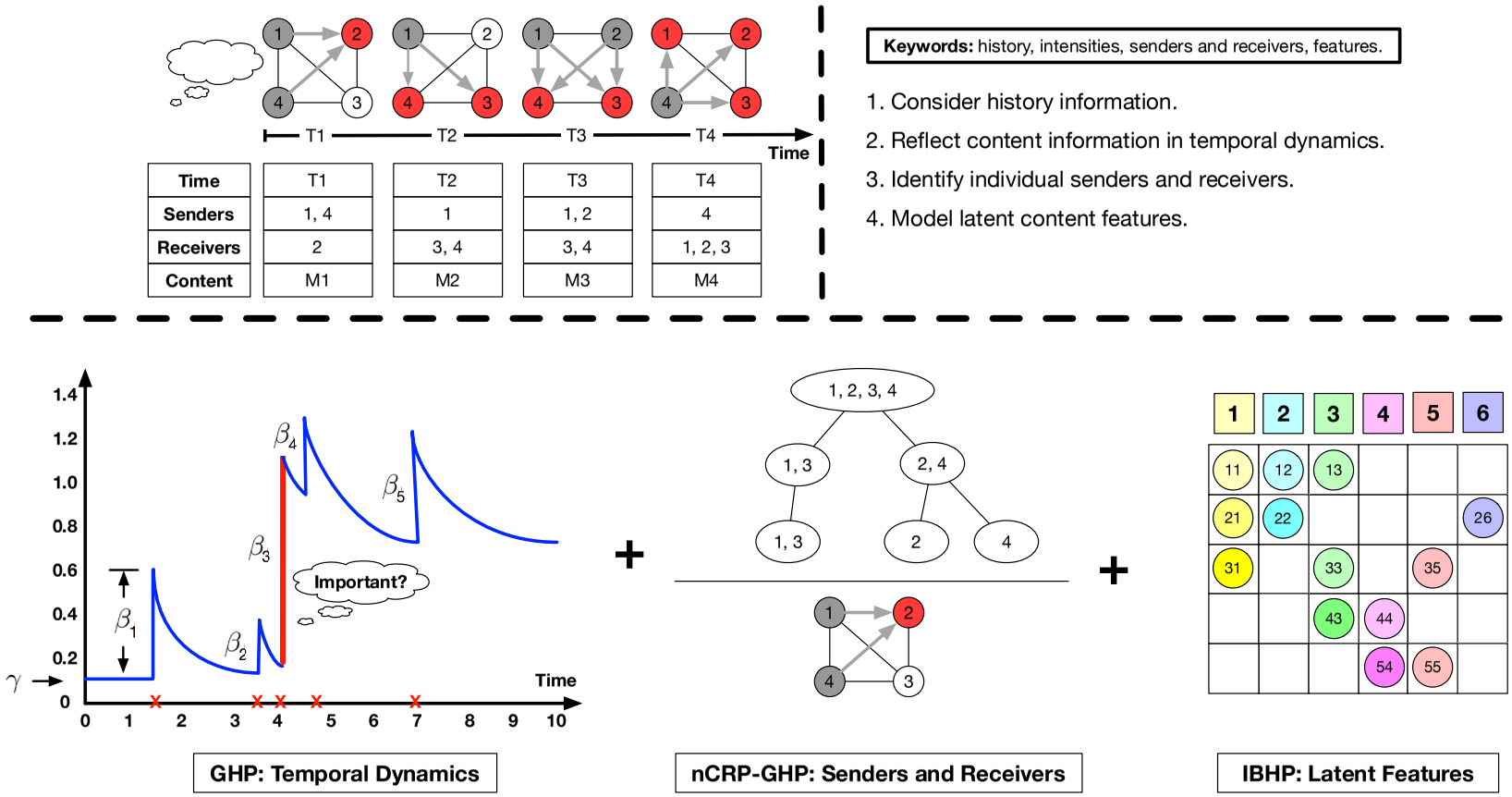


Fig. 1.2. Unified framework with three model components. The GHP better captures the temporal intensities of events based on the contents that have been communicated, the nCRP-GHP identifies senders and receivers based on the underlying hierarchical structure of individuals inferred from the data, and the IBHP captures the dependency between the temporal and textual dynamics of the communication in order to discover interesting latent content features.

1.4 Comparison to the Previous Work

From non-Hawkes to Hawkes Modeling of Relational Data

The interest of modeling relational data dates back to at least the work of [1], who introduced the Bayesian formulation of the stochastic block-model. Early approaches [3, 14, 15] have focused on declared relationships between individuals to infer latent group structures. For example, if three people declare they like each other but dislike others, it is reasonable to put them into one group.

However, these declared relationships are not easily accessible, sometimes incorrect and usually highly subjective. Instead, interaction data have been used to learn latent structure in an unsupervised manner. This approach is motivated by the fact that entities organize themselves into groups having frequent interactions between each other. Unlike previous approaches that focused on subjectively declared relationships, the idea is to exploit the actual evidence at hand to reach conclusions about group formations, resulting in more objective data-driven inferences.

Another limitation of previous models is their incapability to capture reciprocity in social interactions. Reciprocity is a common characteristic in group dynamics. It expresses the fact that social entities reciprocate in their interaction between each other. For example, if Alice has sent a message to Bob, it increases the likelihood of Bob replying back to Alice. Reciprocity is expected to be more prominent between entities within a group, and hence it can be used to infer social groups.

In recent years, HPs [8,9,16–18] have become a popular modeling choice to capture such temporal dynamics [10–13, 19, 20, 20–60, 60–75]. The benefits of using HPs are two-fold: first they capture the self- and mutually-exciting temporal dynamics of communication activities, and second, their probabilistic nature enables the introduction of rich structure into the modeling.

Of particular relevance is the work of [13] that proposed a nonparametric Bayesian model combining HP and the Infinite Relational Model (IRM) [3,14,15] to infer social

structures from continuous time interaction data. Pairs of mutually-exciting HP are able to exploit reciprocity to infer social groups.

Including textual information in HP models

A major factor that encourages the use of HP is the capability to model reciprocity in the interaction between social entities. However, reciprocation can be dynamically conditioned on many factors, among which, two key factors are of particular interest: the significance of each message sent by the sender, and the receptivity to each message received by the receiver.

The model proposed by [13] does not take these factors into consideration, instead it assumes that entities reciprocate simply because they receive a message, and gives no consideration to the content of the message and its effects on the interaction. For social media data, content is clearly an important factor determining how one event affects future activity. In real communication, conveying an important message develops interest in the receiver. Then, if the receiver finds the message relevant, reciprocation takes place. Accordingly, reciprocal communication emerges from the interplay of these two factors.

In the first component of the dissertation, we focus on including message content information into the HP with the Gaussian Hawkes Processes (GHP). The impact of message contents can be represented in the instantaneous rate change of communications, hence it is well justified to introduce GPs to model the intensity shock of HPs, i.e., use message contents as the input for the GPs, and the outputs of the GPs model the intensity shocks of the HPs. This is the main idea behind our GHP model.

In the literature, [76, 77] allow mutually exciting events to be modeled, but they do not use content information to model dependencies between events. Our work is also closely related to [25], which combines mutually exciting HP with random graph models by defining the excitation function, between a pair of nodes, as a product of latent binary indicator variables, indicating the presence or absence of edge, and

weight variable that determines the strength of interaction between the two nodes. But again, unlike our model, their method does not use side information, such as information content, and simply relies on time interaction data to infer latent network structures. Other works [76–78] are based on temporal Poisson-processes, where the rate of events on each edge is independent of every other edge.

Our model uses HPs which are capable of dealing with interaction and reciprocal events, and also uses message content information to capture the dynamics more accurately. By introducing GPs to HPs, we are able to model nonhomogeneous excitation functions. In addition to that, since we use GPs to model the flexible rates of reciprocal activities between two entities, our model can capture the asymmetry in reciprocity more accurately. This, as a by-product, leads to a better cluster detection capability.

From individual HP to sender-receiver HP models

With the power of GPs in hand, we are able to infer more complex structures behind the communication activities among entities. The IRM typically assumes that there is a fixed graph independent of the data, describing the relationship between individuals and their roles of actions, e.g., individuals may be assumeed equally likely to be the senders and receivers of a message. This idea is used in many proposed works [3,14]. We aim to not oversimplify realities with this assumption, but instead to learn the senders and receivers non-parametrically based on their interaction histories.

In the second component of the dissertation, we model senders and receirvers in HPs with the nested Chinese restaurant process (nCRP) and call it nCRP-Gaussian-Hawkes process (nCRP-GHP). The motivation of this model comes from the observation that modelling message contents accurately at the individual level and identifying senders and receivers of messages involve exploiting latent hierarchical structure present among users, and nCRP from the nonparametric Bayesian methods is expected to improve the relatively impoverished structure present in earlier works.

The closest existing works to our nCRP-GHP model are [10, 37, 79], though none of these explore hierarchical clusterings of senders and receivers with HPs. The nested Chinese restaurant franchise process model of [79] combines ideas from the hierarchical Dirichlet process (HDP) [80] and the nested Chinese Restaurant Process (nCRP) [81] to allow each object to be represented as a mixture of paths over a tree, and to decouple the task of modeling hierarchical structure from that of modeling observations. The work of [37] connects Dirichlet processes and HPs to allow the number of clusters to grow while at the same time learning the changing latent dynamics governing the continuous arrival patterns. These works are extended by our model, which has a hierarchical structure embedded with temporal point processes.

Modeling the interplay between temporal and textual dynamics

So far, we have seen how content information can drive the rate dynamics of HPs, but in turn, it is also reasonable to argue that contents can be influenced by the temporal dynamics of communications as well. In the third component of the dissertation, we introduce the interplay between HPs and Bayesian nonparametric latent feature models in the Indian Buffet Hawkes Process (IBHP).

Latent feature models (both parametric and nonparametric) have found wide application in settings where exchangeability holds. A canonical model from Bayesian nonparametric methods is the Indian buffet process (IBP). While there has been some work towards relaxing exchangeability assumptions to allow for temporal dynamics, modeling the full richness of interactions remains an open challenge. Our main contribution in the IBHP is a framework that facilitates the modeling of temporal dynamics through a combination of ideas from the IBP with those of the HP.

The idea of considering nonparametric Bayesian models with temporal point processes in a unified framework has been popular in recent years. For example, [13] proposed a Bayesian nonparametric model that utilizes the Chinese restaurant process (CRP) as a prior for the clusters among individuals, whose rates of communications

are modeled by HPs. HP models with various generalizations of the CRP, such as the distance dependent CRP (ddCRP) [56], the nested CRP (nCRP) [11], and the Chinese restaurant franchise processes (CRFP) [45], have also been explored in the machine learning community.

Perhaps the closest works to our IBHP model are [37] and [63]. In [37], the authors proposed a Dirichlet HP (DHP) model that combines the CRP with HP in a unified framework, where the cluster assignment in CRP is driven by the intensities of HP. [63] further developed this in their Hierarchical Dirichlet HP (HDHP) model by replacing the CRP with a CRFP that is capable of modeling steaming data for multiple users.

However, theses models cannot capture complex dependencies in triggering rules, temporal dynamics, and etc., which lead to two major distinctions compared to our IBHP: 1) In both the DHP and HDHP models, events are triggered by single factors, while in our IBHP, multiple latent triggering factors are introduced; 2) the form of the triggering kernels do not depend on history events, and in contrast, our IBHP model is more flexible to be able to adopt non-additive triggering rules to learn different perspectives of the observed data.

Other attempts have been made by borrowing the ideas from Deep Learning. For example, [55] proposed a model to view the intensity function of a temporal point process as a nonlinear function of the history, and use recurrent neural networks to automatically learn a representation of the influences from the event history. [60] modeled streams of events by constructing a neurally self-modulating multivariate point process where the intensities of multiple event types evolve based on a continuous-time LSTM. Lastly, [72] considered the use of latent factors in HP models to represent dependencies among instances that influence reciprocity over time. But the work focused on modeling static factors of homophily and reciprocity in social networks and not the evolution of factors over time.

1.5 Overview of the Dissertation

The remainder of the dissertation is organized as follows. In Chapter 2, we begin with an introduction to the key concepts that are the building blocks of this dissertation. In Chapters 3, 4, and 5, the main part of this dissertation, we present our three novel Bayesian nonparametric HP models and empirical evaluations on synthetic and real data: 1) the Gaussian Hawkes process (GHP) [10] to incorporate content information using GPs; 2) the nested Chinese restaurant Gaussian Hawkes process (nCRP-GHP) [11] to model senders and receivers with nCRP; and 3) the Indian buffet Hawkes process (IBHP) [12] to model the interplay between temporal and textual dynamics with the generalized IBP. Finally, in chapter 6, we summarize the main contributions of the dissertation and discuss future work.

2. BACKGROUND

2.1 Hawkes Processes

We first review some basic concepts of point processes, where Hawkes process (HP) is a special case.

2.1.1 A Brief Introduction to Point Processes

A stochastic process is a collection of random variables. Temporal point processes are a specific class of stochastic processes defined in the time domain. Formally, we have [82, 83]:

Definition 2.1.1 Let $\mathcal{X}$ be an arbitrary complete separable metric space (c.s.m.s) and $\mathcal{B}_{\mathcal{X}} = \mathcal{B}(\mathcal{X})$ the σ-field of its Borel sets.

1. A Borel measure μ on the c.s.m.s. $\mathcal{X}$ is boundedly finite if $\mu(A) < \infty$ for every bounded Borel set A.

2. $\mathcal{M}_{\mathcal{X}}^{\#}$ is the space of all boundedly finite measures on $\mathcal{B}_{\mathcal{X}}$.

3. $\mathcal{N}_{\mathcal{X}}^{\#}$ is the space of all boundedly finite integer-valued measures $N \in \mathcal{M}_{\mathcal{X}}^{\#}$, called counting measures for short.

4. $\mathcal{N}_{\mathcal{X}}^{\#*}$ is the family of all simple counting measures, consisting of all those elements of $\mathcal{N}_{\mathcal{X}}^{\#}$ for which $N\{x\} \equiv N(\{x\}) = 0$ or $1 \quad (\forall x \in \mathcal{X})$.

5. A point process N on state space $\mathcal{X}$ is a measurable mapping from a probability space $(\Omega, \mathcal{E}, \mathcal{P})$ into $(\mathcal{N}_{\mathcal{X}}^{\#}, \mathcal{B}(\mathcal{N}_{\mathcal{X}}^{\#}))$.

6. A point process N is simple when

$$\mathcal{P}\{N \in \mathcal{N}_{\mathcal{X}}^{\#*}\} = 1. \tag{2.1}$$

One of the most powerful and popular temporal point process models is the inhomogeneous Poisson process, parametrized by a rate function $\lambda(t)$, which is independent from its history events. It has two properties:

1. The number of events in an interval $(a, b]$ is Poisson distributed with mean $\Lambda_{(a,b]} = \int_a^b \lambda(s)\,\mathrm{d}\,s$, and,
2. the number of events in disjoint intervals are independent random variables.

The special case where $\lambda(t)$ equals a constant λ gives the homogeneous Poisson process, and if $\lambda(t)$ is a random variable, a doubly stochastic process, or a Cox process. However, Cox processes can be thought of as conditionally inhomogeneous Poisson processes, i.e., given the intensity $\lambda(t)$, events are still independent. In real-world social network communications however, messages directly and causally affect each other. Poisson processes cannot capture such self- or mutual-excitation, and instead, there has been much interest in using Hawkes processes to model such data.

2.1.2 Hawkes Processes and Their Branching Representation

Hawkes processes (HP) [8, 9, 16–18] are point processes [83] where earlier events have a time-decaying influence on future events. A self-exciting Hawkes process has a rate-function that is dependent on its own history (i.e., $\lambda(t)$ is dependent on the event history for $s \leq t$). Similarly, a pair of mutually-exciting HP have mutually-dependent rate functions that depend on each others' histories.

Let $N(\cdot)$ and $N'(\cdot)$ be a pair of mutually-exciting HP. Recall from Equation 1.2 that each datum is denoted as a quadruplet, where the time-stamp t_i is one of the components. If we denote the event time history of N' as $\mathcal{H}_{N'} = \{t'_1, \cdots, t'_n\}$, then the conditional rate function $\lambda(t)$ of $N(\cdot)$, given the event time history $\mathcal{H}_{N'}$ of N', has the form

$$\lambda(t) = \gamma + \int_0^t \kappa(t - s)\,\mathrm{d}\,N'(s) \tag{2.2}$$

where the constant γ is the base rate, $N'(s)$ the number of arrivals within $[0, s)$, and the non-negative function $\kappa(\cdot)$ is called the *triggering kernel* (and many other names, e.g., *excitation function*, *impact function*, *link function*, *transfer function*, *density function*, and etc.)

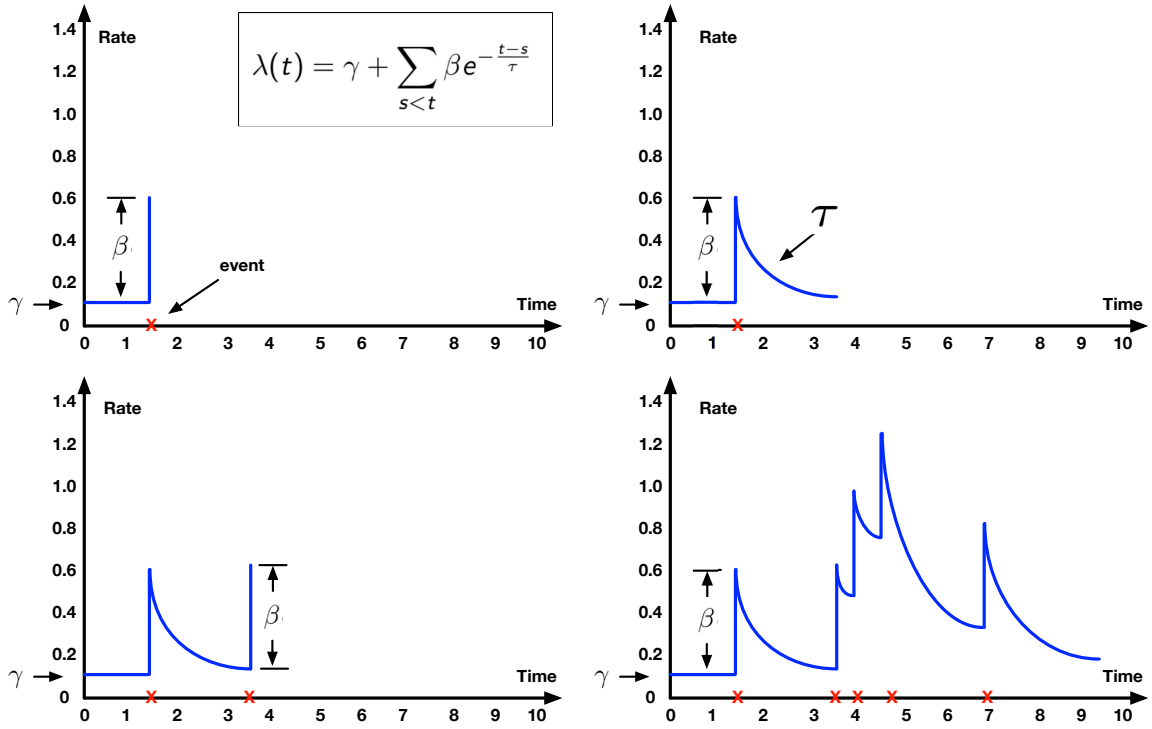


Fig. 2.1. Illustration of a simple self-exciting Hawkes process. The base rate $\gamma = 0.1$, the "jump size" $\beta = 0.4$, and the inverse decay speed $\tau = 1$.

If $\kappa(\cdot) = 0$ then the process becomes a Poisson process with rate γ. If the counting measure $N'(\cdot)$ is $N(\cdot)$ itself, the process is self- exciting: its current rate only depends on its own past events. If the two counting measures are different, the rates are affected by the past events of each other. Figure 2.1 shows an illustration of a simple self-exciting HP. We see that each event creates a jump in the rate function of HP.

If one ignores the time/location of the events, a Hawkes process is simply a branching process. The cluster representation of HP was first discussed in [16]. A branching

process, e.g., *the Galton-Watson model*, is a mathematical description of the growth of a population for which the individual produces offsprings according to stochastic laws. The generative process of HP in its branching representation can be described as follows (see Figure 2.2 for an illustration).

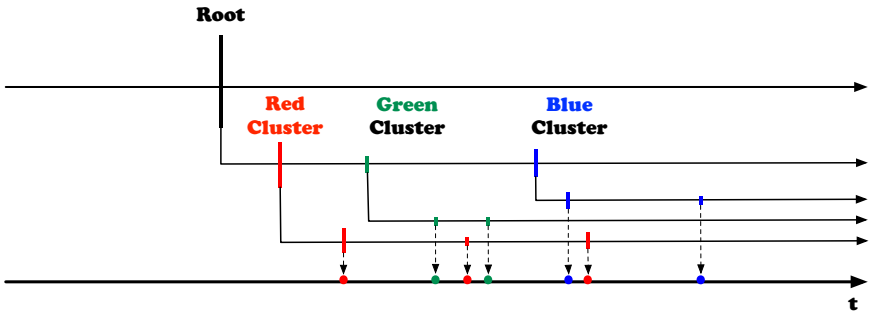


Fig. 2.2. In its branching representation, the illustrated Hawkes process has three immigrants (red, green, and blue), and each of the immigrants has several offspring (red immigrant has 3, green has 2, and blue has 3).

Immigrants. In the branching representation, γ is the base rate, or the exogenous rate of immigrants, i.e., the immigrants' arrivals t_i follow a homogeneous Poisson process with rate

$$\lambda_0 = \gamma \tag{2.3}$$

and the number of immigrants N_0 follows a Poisson distribution

$$N_0 \sim Poisson(\gamma T) \tag{2.4}$$

Hence t_i are i.i.d. $U[0, T)$ random variables conditioned on N_0.

Offspring. The offspring of each immigrant t_i form an inhomogeneous Poisson process, with rate

$$\lambda_i(s) = \kappa(s - t_i), \quad s \geq t_i \tag{2.5}$$

The branching ratio

$$\rho := \rho_i = \int_{t_i}^{\infty} \kappa(s - t_i)\, \mathrm{d}\, s = \int_{0}^{\infty} \kappa(\delta)\, \mathrm{d}\, \delta, \quad \forall i = 1, \ldots, N_0 \tag{2.6}$$

indicates the endogenous rate of offspring, or the expected number of offspring. If $\rho < 1$, which is called subcritical, the process is stationary; otherwise, it is non-stationary and may explode in finite time ($\rho = 1$ is critical and $\rho > 1$ is supercritical). The numbers of offspring for different immigrants are i.i.d. random variables, following a Poisson distribution:

$$N_i \sim Poisson(\rho) \tag{2.7}$$

Conditioned on N_i, the next inter-arrival times are i.i.d. random variables with pdf

$$f(\delta) = \frac{\kappa(\delta)}{\rho} = \frac{\kappa(\delta)}{\int_0^\infty \kappa(\delta)\,\mathrm{d}\,\delta} \tag{2.8}$$

In the case of an exponential kernel, the above has a simple form of an exponential distribution.

2.1.3 Hawkes Processes and Their Statistical Properties

The Hawkes Rate Function and the Cumulative Rate Function

The conditional rate function of temporal point processes at time s is a random variable, whose primitive definition is

$$\lambda(s) := \frac{P(\text{next event time in}[s, s + \mathrm{d}\,s])}{\mathrm{d}\,s} = \frac{\mathbb{E}[\,\mathrm{d}\,N(s) \mid \mathcal{H}_{[0,s)}]}{\mathrm{d}\,s} \tag{2.9}$$

where $\mathcal{H}_{[0,s)} = \{t_{i=1,\dots,n} \mid t_i \in [0, s)\}$ is the collection of all history event times up to time s. It implies there is no event observed in (t_n, s). Since HPs are discrete, we can rewrite Equation 2.2 as:

$$\lambda(t) = \gamma + \sum_{0 \le s_1,\cdots,s_n < t} \kappa(t - s_i)[N(s_{i+1}) - N(s_i)] = \gamma + \sum_{0 \le t_i < t} \kappa(t - t_i) \tag{2.10}$$

where we have used the fact that, for any two given time points, there can be at most one event. The cumulative rate function $\Lambda(0, T)$ of a Hawkes process is defined as:

$$\Lambda(0, T) := \int_0^T \lambda(t)dt \tag{2.11}$$

which is also called the compensator of the Hawkes process.

The Hawkes Next-event-time Distribution

Given the Hawkes rate function $\lambda(s)$, we are able to derive the next-event-time distribution. From the primitive definition of the conditional rate function $\lambda(s)$ in Equation 2.9, we have:

$$\begin{aligned}
\lambda(s)\,\mathrm{d}\,s &= \mathbb{E}(\,\mathrm{d}\,N(s)|\mathcal{H}_{[0,s)}) \\
&= \mathbb{E}(N(s+\,\mathrm{d}\,s) - N(s)|\mathcal{H}_{[0,s)}) \\
&= P(\text{an event in } [s, s+\,\mathrm{d}\,s) \mid \text{next event not in } (t_n, s)) \\
&= \frac{P(\text{an event time in } [s, s+\,\mathrm{d}\,s) \text{ and next event not in } (t_n, s))}{P(\text{next event not in } (t_n, s))} \\
&= \frac{P(\text{next event time in } [s, s+\,\mathrm{d}\,s))}{P(\text{next event not in } (t_n, s))} \\
&= \frac{f(s)\,\mathrm{d}\,s}{1-F(s)}
\end{aligned} \tag{2.12}$$

where we have defined

$$f(s) := P(\text{next event time in } [s, s+\,\mathrm{d}\,s)) \tag{2.13}$$

$$F(s) := P(\text{next event in } (t_n, s)) \tag{2.14}$$

Cancel the $\,\mathrm{d}\,s$ part, we obtain

$$\lambda(s) = \frac{f(s)}{1-F(s)} = \frac{\frac{\mathrm{d}}{\mathrm{d}\,s}F(s)}{1-F(s)} = -\frac{\mathrm{d}}{\mathrm{d}\,s}\log(1-F(s)) \tag{2.15}$$

Integrating both sides from t_n to t, we have

$$\Lambda(t_n, t) := \int_{t_n}^{t} \lambda(s)\,\mathrm{d}\,s = -\log(1-F(t)) \tag{2.16}$$

hence for $t \in (t_n, \infty)$,

$$F(t) = 1 - \exp\{-\Lambda(t_n, t)\} \tag{2.17}$$

$$f(t) = F(t)' = \lambda(t)\exp\{-\Lambda(t_n, t)\} \tag{2.18}$$

where $f(t)$ is the Hawkes next-event-time density function, conditioned on the last event happened at time t_n.

The Hawkes Likelihood Function

Proposition 2.1.1 The loglikelihood function of a Hawkes process is

$$\log \mathcal{L}(\lambda(t)) = -\Lambda(0,T) + \sum_{i=1}^{n} \log \lambda(t_i) \tag{2.19}$$

Proof The joint data likelihood during time interval $[0,T)$, of events $t_1,\ldots,t_n$ and no event in (t_n,T), is nothing but:

$$\begin{aligned}
\mathcal{L}(\lambda(t)) &= (1-F(T)) \prod_{i=1}^{n} f(t_i) \\
&= e^{-\int_{t_n}^{T} \lambda(t)dt} \prod_{i=1}^{n} \lambda(t_i) e^{-\int_{t_{i-1}}^{t_i} \lambda(s)ds} \\
&= \left\{ e^{-\int_{t_n}^{T} \lambda(t)dt} \prod_{i=1}^{n} e^{-\int_{t_{i-1}}^{t_i} \lambda(s)ds} \right\} \prod_{i=1}^{n} \lambda(t_i) \\
&= e^{-\int_{0}^{T} \lambda(t)dt} \prod_{i=1}^{n} \lambda(t_i) \\
&= \exp\{-\Lambda(0,T)\} \prod_{i=1}^{n} \lambda(t_i)
\end{aligned} \tag{2.20}$$

or in its log likelihood form:

$$\log \mathcal{L}(\lambda(t)) = -\Lambda(0,T) + \sum_{i=1}^{n} \log \lambda(t_i) \tag{2.21}$$

2.1.4 Hawkes Processes with Exponential Kernels

Cumulative Rate Function

Suppose the triggering function $\kappa(\delta)$ of a Hawkes process takes the form

$$\kappa(\delta) = \beta \exp\left(-\frac{\delta}{\tau}\right) \tag{2.22}$$

then the cumulative rate function, for the duration $[0, T)$, can be written as:

$$\begin{aligned}
\Lambda(0,T) := \int_0^T \lambda(t)\,\mathrm{d}\,t &= \int_0^T \left[\gamma + \int_0^t \beta e^{-\frac{t-s}{\tau}}\,\mathrm{d}\,N(s)\right]\,\mathrm{d}\,t \\
&= \gamma T + \int_0^T \int_0^t \beta e^{-\frac{t-s}{\tau}}\,\mathrm{d}\,N(s)\,\mathrm{d}\,t \\
&= \gamma T + \int_0^T \int_s^T \beta e^{-\frac{t-s}{\tau}}\,\mathrm{d}\,t\,\mathrm{d}\,N(s) \\
&= \gamma T - \beta\tau \int_0^T \left[e^{-\frac{T-s}{\tau}} - 1\right] \mathrm{d}\,N(s) \\
&= \gamma T - \beta\tau \sum_{i=1}^{n} \left[e^{-\frac{T-t_i}{\tau}} - 1\right]
\end{aligned} \tag{2.23}$$

where n is the total number of events happened during $[0, T)$. Specifically, if $T = t_n$, then

$$\Lambda(0, t_n) := \gamma t_n - \beta\tau \sum_{i=1}^{n} \left[e^{-\frac{t_n - t_i}{\tau}} - 1\right] \tag{2.24}$$

Likelihood Function

Based on Equation 2.21, the loglikelihood function of a Hawkes process with an exponential kernel is:

$$\begin{aligned}
\log \mathcal{L}(\lambda(t)) &= -\Lambda(0,T) + \sum_{i=1}^{n} \log \lambda(t_i) \\
&= -\gamma T + \beta\tau \sum_{i=1}^{n} \left[e^{-\frac{T-t_i}{\tau}} - 1\right] + \sum_{i=1}^{n} \log\left(\gamma + \beta \sum_{j=1}^{i-1} e^{-\frac{t_i - t_j}{\tau}}\right)
\end{aligned} \tag{2.25}$$

Recursive Definition of the Rate Function and the Likelihood Functions

To compute the likelihood of HPs with exponential kernels, there is a useful recursive definition of $\lambda(t)$ for $t > t_n$:

$$\begin{aligned}
\lambda(t) &= \gamma + \sum_{0 \le t_i < t} g(t - t_i) \\
&= \gamma + \sum_{j=1}^{n} \beta \exp\left(-\frac{t - t_j}{\tau}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \gamma + \sum_{j=1}^{n} \beta \exp\left(-\frac{t - t_n + t_n - t_j}{\tau}\right) \\
&= \gamma + \exp\left(-\frac{t - t_n}{\tau}\right) \sum_{j=1}^{n} \beta \exp\left(-\frac{t_n - t_j}{\tau}\right) \\
&= \gamma + \exp\left(-\frac{t - t_n}{\tau}\right) \left[\sum_{j=1}^{n-1} \beta \exp(-\frac{t_n - t_j}{\tau}) + \beta \exp(-\frac{t_n - t_n}{\tau})\right] \\
&= \gamma + \beta \exp\left(-\frac{t - t_n}{\tau}\right) \left[\frac{\lambda(t_n) - \gamma}{\beta} + 1\right] \qquad (2.26)
\end{aligned}
$$

where t_n is the last event time before time t. Moreover, rearrange the terms we obtain

$$
\frac{\lambda(t) - \gamma}{\beta} = \exp\left(-\frac{t - t_n}{\tau}\right) \left[\frac{\lambda(t_n) - \gamma}{\beta} + 1\right] \qquad (2.27)
$$

This suggests us to define

$$
\begin{aligned}
A(i) := \sum_{j=1}^{i-1} e^{-\frac{t_i - t_j}{\tau}} &= \frac{\lambda(t_i) - \gamma}{\beta} \\
&= \exp\left(-\frac{t_i - t_{i-1}}{\tau}\right) \left[\frac{\lambda(t_{i-1}) - \gamma}{\beta} + 1\right] \\
&= \exp\left(-\frac{t_i - t_{i-1}}{\tau}\right) [A(i-1) + 1] \qquad (2.28)
\end{aligned}
$$

and hence the likelihood function can be computed in $\mathcal{O}(n)$ with the above definition. To summarize, we write

$$
\log \mathcal{L}(\lambda(t)) = -\gamma T + \beta\tau A(n) + \sum_{i=1}^{n} \log\left(\gamma + \beta A(i)\right) \qquad (2.29)
$$

where

$$
A(1) = 0, \qquad (2.30)
$$

$$
A(i) = \exp\left(-\frac{t_i - t_{i-1}}{\tau}\right) [A(i-1) + 1], \quad i = 2, \ldots, n \qquad (2.31)
$$

$$
A(T) = \exp\left(-\frac{T - t_n}{\tau}\right) [A(n) + 1]. \qquad (2.32)
$$

Derivatives of the Hawkes Rate Function

For optimization tasks w.r.t HPs with exponential kernels, a useful expression is the derivative of the rate function:

$$\begin{aligned}\lambda'(t) &= \lim_{\Delta t \to 0} \frac{\sum_{0 \le t_j < t+\Delta t} \beta e^{-\frac{t+\Delta t-t_j}{\tau}} - \sum_{0 \le t_j < t} \beta e^{-\frac{t-t_j}{\tau}}}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{e^{-\frac{\Delta t}{\tau}} \sum_{0 \le t_j < t} \beta e^{-\frac{t-t_j}{\tau}} - \sum_{0 \le t_j < t} \beta e^{-\frac{t-t_j}{\tau}}}{\Delta t} \\ &= (\lambda(t) - \gamma) \lim_{\Delta t \to 0} \frac{e^{-\frac{\Delta t}{\tau}} - 1}{\Delta t} = -\frac{1}{\tau}[\lambda(t) - \gamma] \end{aligned} \tag{2.33}$$

2.1.5 Simulation Algorithms for Hawkes Processes

Popular simulation algorithms for HP include the branch clustering method ([16], [84], [85]), Ogatas modified thinning method ([86]), and the fast thinning method for HP with exponential triggering kernels ([87]).

2.1.6 Inference Algorithms for Hawkes Processes

Inference algorithms for HP fall mainly into three categories [59]: 1) methods related to Maximum Likelihood Estimation (MLE) [88], which are usually quite restrictive and incompatible with rich latent structure; 2) variational approximations [33], which often suffer from poor convergence issues and are best applicable when the inference problem exhibits a convenient simplifying approximation; and 3) sampling methods. In this dissertation, we focus on the sampling algorithms.

2.2 Bayesian Nonparametric Models

Next to the Hawkes process, Bayesian nonparametric models are another important component in our framework. We review some of the related concepts here.

2.2.1 The Gaussian Process (GP)

Definition 2.2.1 A Gaussian process (GP) [89] is a collection of random variables, any finite number of which have a joint Gaussian distribution.

A GP is completely specified by its mean function and covariance function. We define mean function $m(\mathbf{x})$ and the covariance function $k(\mathbf{x}, \mathbf{x}')$ of a real process $f(\mathbf{x})$ as

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \tag{2.34}$$

and will write the Gaussian process as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{2.35}$$

Usually, for notational simplicity we will take the mean function to be zero.

The covariance function specifies the covariance between pairs of random variables. A common choice is the squared exponential covariance function:

$$cov(f(\mathbf{x}_p), f(\mathbf{x}_q)) = k(\mathbf{x}_p, \mathbf{x}_q) = \tau^2 \exp\left(-\frac{|\mathbf{x}_p - \mathbf{x}_q|^2}{l^2}\right)\left(\tag{2.36}$$

where τ is called the *amplitude parameter*, and l the *length scale parameter*. Here τ controls the magnitude and l the smoothness of the functions drawn from a GP.

2.2.2 The Chinese Restaurant Process (CRP)

The Chinese restaurant process (CRP) [90] is an infinitely exchangeable probability distribution over partitions that can be described using the following metaphor involving customers entering a restaurant of infinity number of (possible) tables: The first customer sits at table 1; the following customers pick a new table with probability proportional to some constant, and pick an existing table with probability proportional to the number of people already assigned to that table:

$$p(\pi_i|\pi_{-i}) = \begin{cases} \left(\frac{\alpha}{N-1+\alpha} \right. & \text{if } \pi_i \text{ a new table} \\ \left(\right. & \\ \left(\right. & \\ \left. \frac{|B_j|}{N-1+\alpha} \right. & \text{if } \pi_i \text{ an existing table } j \end{cases} \tag{2.37}$$

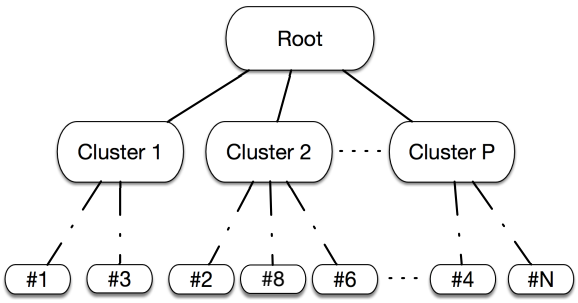


Fig. 2.3. A clustering tree sampled from an nCRP.

where π_{-i} is the assignment vector π without the i^{th} entry, and $|B_j|$ is the number of customers seated at table j The joint probability is $p(\pi|\alpha) = \alpha^{|B|}\frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}\prod_{j=1}^{|B|}\Gamma(|B_j|)$, where $|B|$ is the total number of tables, and $(|B_j|-1)!$ is the factorial of $|B_j|-1$, the number of individuals in the j^{th} table minus one.

The nested Chinese restaurant process (nCRP) is similar to a CRP, but with a hierarchical tree structure (see Figure 2.3). For an nCRP with L levels, rather than being assigned to a single table, a user is assigned to a sequence of L tables. After a customer comes into the first restaurant and picks a table, the table no longer has seats but instead directs the customer is directed to a level-2 restaurant, again picking tables according to the paths of previous users. This process repeats $L-1$ times until the customer finds a seat at a level-L restaurant. The consequence now is that a customer selects not just one table, but a sequence of tables; in our application, this will allow a message to belong not just to a user or group, but a nested set of groups. For more details on the nCRP, see [79, 81].

Likelihood of the Chinese Restaurant Processes (CRP)

The conditional distribution of the CRP can be written as

$$p(\pi_i|\pi_{-i},\alpha) = \begin{cases} \frac{\alpha}{N-1+\alpha} & \text{if } \pi_i = \text{new table} \\ \frac{|B_j|}{N-1+\alpha} & \text{if } \pi_i = \text{one of the existing tables } B_j \end{cases} \tag{2.38}$$

where π_{-i} is the assignment vector π without the i^{th} entry. The joint probability is

$$p(\pi|\alpha) = \frac{\alpha^{|B|}\prod_{j=1}^{|B|}(|B_j|-1)!}{\prod_{j=1}^{N}(j-1+\alpha)} = \alpha^{|B|}\frac{\Gamma(\alpha)}{\Gamma(N+\alpha)}\prod_{j=1}^{|B|}\Gamma(|B_j|) \tag{2.39}$$

where $|B|$ is the total number of tables, and $(|B_j|-1)!$ is the factorial of $|B_j|-1$, the number of individuals in the j^{th} table minus one.

2.2.3 The Indian Buffet Process (IBP)

The Indian Buffet Process (IBP) [91] is a Bayesian nonparametric prior over an infinite dimensional binary matrix whose columns represent exchangeable factors underlying observations. Suppose there are N customers (observations) arriving sequentially in a restaurant with infinite number of dishes (factors). Each customer is assigned dishes as follows:

1. The first customer comes in and helps herself to Poisson(α) dishes.
2. When the n^{th} customer arrives, they independently choose each existing dish with probability m_k/n, where m_k is the number of customers that have already sampled dish k (the popularity of the dish).
3. In addition, they sample Poisson(α/n) new dishes.

Additionally, the IBP has several distinctive features:

1. Each observation can have multiple factors.
2. The number of factors grows non-parametrically depending on the size of the dataset.
3. The probability of adding new factors deceases – since the number of new factors follows Poisson(α/n) which decreases as n increases.
4. The row sums are distributed Poisson(α).

Likelihood of the Nested Chinese Restaurant Processes (nCRP)

We write the binary feature matrix as $\mathbf{Z}$. The conditional probability that element $z_{ik} = 1$ is given by

$$P(z_{ik} = 1|\mathbf{z}_{-i,k}) = \frac{m_{-ik}}{N} \tag{2.40}$$

where $\mathbf{z}_{-ik}$ is the k^{th} column without considering the i^{th} observation, and m_{-ik} is the sum of $\mathbf{z}_{-ik}$. We need only condition on $\mathbf{z}_{-ik}$ rather than including other columns because the columns of the matrix are generated independently under this prior. In a Bayesian framework, the posterior can be written as:

$$P(z_{ik} = 1|\mathbf{Z}_{-ik}, \mathbf{X}) \propto P(\mathbf{X}|\mathbf{Z})P(z_{ik} = 1|\mathbf{z}_{-ik}) \tag{2.41}$$

where $P(X|Z)$ is the data likelihood.

3. INCORPORATING CONTENT INFORMATION WITH GAUSSIAN PROCESSES

In this component of the dissertation [10], we explore how to incorporate textual content into HP models with the Gaussian process (GP).

3.1 Motivation

A major factor that encourages the use of HP is the capability to model reciprocity in the interaction between social entities. However, reciprocation can be dynamically conditioned on many factors, among which, two key factors are of particular interest: the significance of each message sent by the sender, and the receptivity to each message received by the receiver.

The model proposed by [13] does not take these factors into consideration, instead it assumes that entities reciprocate simply because they receive a message, and gives no consideration to the content of the message and its effects on the interaction. For social media data, content is clearly an important factor determining how one event affects future activity. In real communication, conveying an important message develops interest in the receiver. Then, if the receiver finds the message relevant, reciprocation takes place. Accordingly, reciprocal communication emerges from the interplay of these two factors.

In this chapter, we focus on including message content information into the HP with the Gaussian Hawkes Processes (GHP). The impact of message contents can be represented in the instantaneous rate change of communications, hence it is well justified to introduce GPs to model the intensity shock of HPs, i.e., use message contents as the input for the GPs, and the outputs of the GPs model the intensity shocks of the HPs. This is the main idea behind our GHP model.

3.2 Background

Amongst the models that use declared relationships to infer group information, the Infinite Relational Model (IRM) [14] is especially flexible and popular. [13] has combined the IRM idea with the concept of HPs to model reciprocity in the interaction between entity groups.

Let V denote the vertices of the graph, corresponding to individuals. The generative model of this HP+IRM is defined as follows:

$$\pi|\alpha \sim CRP(\alpha) \tag{3.1}$$

$$\lambda_{pq}(t)|\gamma_{pq}, \beta_{pq}, \tau_{pq} = \gamma_{pq} n_p n_q + \int_{-\infty}^{t} g_{pq}(t-s)\,\mathrm{d}\,N_{qp}(s), \quad \forall p, q \in range(\pi) \tag{3.2}$$

$$N_{pq}(\cdot)|\lambda_{pq} \sim HawkesProcess(\lambda_{pq}) \tag{3.3}$$

$$N_{uv}(\cdot)|N_{\pi(u)\pi(v)}, \pi \sim Thin(N_{\pi(u)\pi(v)}), \quad \forall u, v \in V \tag{3.4}$$

Here π is a partition of the vertices V , distributed according to the Chinese restaurant process (CRP) with concentration parameter α. We use p and q to index clusters of π. We denote the cluster that vertex u belongs to as $\pi(u)$. The operator $Thin$ refers to thinning; this means distributing the atoms of $N_{pq}(\cdot)$ among each $N_{uv}(\cdot)$, such that $N_{pq} = \sum_{u,v} N_{u,v}(\cdot)$. Any thinning scheme may be used, such as a uniform thinning, which uniformly picks to elements of a cluster. The type of reciprocation (parameterized by g_{pq} and g_{qp}, respectively) differs with events from cluster p to cluster q and events from cluster q to cluster p. This difference in reciprocity is what the model exploits to learn about social groups.

In this model, the Hawkes process conditional rate function can be written as:

$$\lambda_{uv}(t) = \gamma_{pq} + \int_{0}^{t} \beta_{uv} e^{-\frac{t-s}{\tau_{uv}}} dN_{vu}(s) \tag{3.5}$$

where $p = \pi^{-1}(u), q = \pi^{-1}(v)$ are the clusters individuals u and v belong to; and the triggering function $g_{uv}(\cdot)$ is defined as:

$$g_{uv}(\delta) = \beta_{uv} e^{-\frac{\delta}{\tau_{uv}}} \tag{3.6}$$

Geometrically, the excitation function β_{pq} is essentially the "jump size" of the rate function $\lambda_{uv}(t)$ whenever a new message is received. However, in the above definition, β_{uv} is not affected by the content of the message; its value does not change based on the significance and receptivity of the messages.

3.3 Model

We would like to define β_{uv} in a way such that it measures the significance and receptivity of individual messages communicated between individuals u and v. The content measure x_{vu} can be suitably defined according to the application, for example, it can be a distribution of words, the length of the message, or the text entropy of the message, etc. The individual level excitation function $\beta_{uv}(x_{vu}(s)) = 0$ if no message was sent from v to u at time s, but can be otherwise any non-negative function.

We propose to use two sets of Gaussian Process (GP) priors to address sources of inhomogeneity of the excitation functions $\beta_{uv}(\cdot)$, one for the significance of the message and one for the receptivity of the message:

$$\beta_{uv}(s) = e^{r_u(x_{vu}(s))+s_v(x_{vu}(s))} \tag{3.7}$$

where

$$r_u(\cdot) \sim \mathcal{GP}(0, k_r) \tag{3.8}$$

$$s_v(\cdot) \sim \mathcal{GP}(0, k_s) \tag{3.9}$$

k_r and k_s are radial basis function (RBF) kernels of the GPs. The exponential transformation is used to make sure that $\beta_{uv}(\cdot)$ is non-negative.

Larger values of r_u and s_v indicate that an important message has been sent by the sender, and receiver is receptive to the message, these result in larger values for β_{uv}. If either r_u or s_v is small, or both of them have smaller values, it leads to smaller values of β_{uv}. Application of GP functions also allows us to flexibly model the rates of reciprocal activities between two entities, allowing the asymmetry in reciprocity to be

captured more accurately. This, as a by-product, leads to a better cluster detection capability.

The receptivity and significance functions r_u and s_v may have different behaviors and hence are designed to come from two different GPs. One subtle point is that although r_u and s_v seem exchangeable in the definition of β_{uv} and both use message content x_{vu} as input, they are evaluated from different perspectives: r_u evaluates x_{vu} from the receiver u's perspective, while s_v from the sender v's perspective. One alternative way is to model a single pair of GPs $s(\cdot)$ and $r(\cdot)$ for all users, instead of this per-user GP $s_u(\cdot)$ and $r_v(\cdot)$ framework. Experiments have shown that both the modeling schemes have good performances, however, we believe that the per-user GP setting can reveal more interesting user-specific details, and hence in the later sections, our results are based on the per-user GP framework.

The generative process of our model can be summarized as (see Figure 3.1):

$$\pi|\alpha \sim CRP(\alpha) \tag{3.10}$$

$$\lambda_{uv}(t)|\gamma_{pq}, \beta_{uv}(\cdot), \tau_{uv} = \gamma_{pq} + \int_{-\infty}^{t} \beta_{uv}(\mathcal{X}_{vu}) e^{-\frac{t-s}{\tau_{uv}}} dN_{vu}(s) \tag{3.11}$$

$$N_{uv}(\cdot)|\lambda_{uv} \sim HawkesProcess(\lambda_{uv}) \tag{3.12}$$

where $\mathcal{X}_{vu} = \{x_{vu}(s)\}$ is the set of all messages sent from v to u, and the cluster level excitation function β_{pq} can be seen as an additive effect of β_{uv}:

$$\beta_{pq}(\mathcal{X}_{qp}) = \sum_{\pi(u)=p, \pi(v)=q} \beta_{uv}(x_{vu}(s)) \tag{3.13}$$

This model is a GP extension of the Hawkes IRM, and we call it Gaussian Hawkes process (GHP) for short.

In this new model, the excitation function β_{pq} is no longer a constant, as in [13], but a function of the message content in the past events of the reciprocal process N_{qp}, taking into account both the significance and the receptivity of the messages. Our model is a generalization of the model described in [13], and if β_{uv} in equation 3.7 are constants, our model reduces to the model described in [13]. Therefore, all the

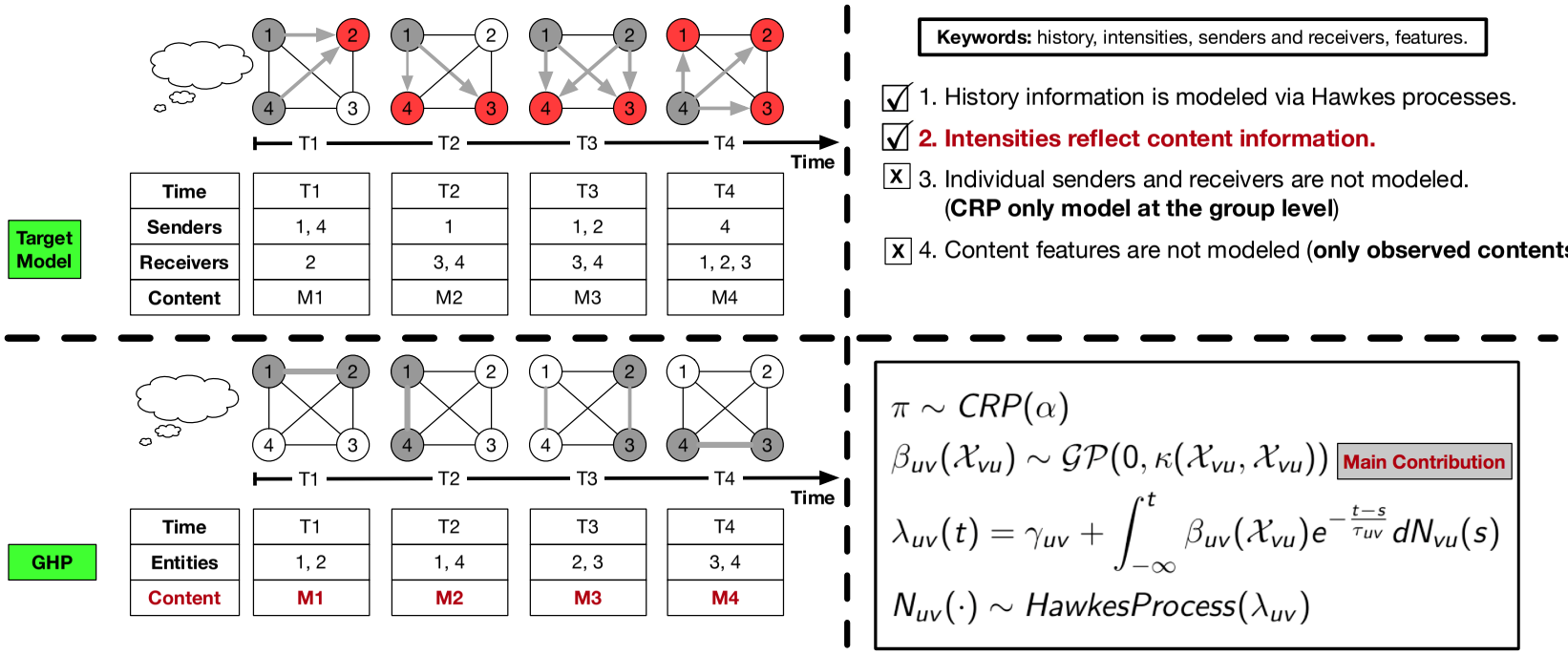


Fig. 3.1. An illustration of the Gaussian Hawkes process (GHP) model, where the content information is taken into account via GPs to model the "jump sizes" in the HP rate functions.

basic features of the original model are inherited by our model. Also, in our modeling framework, the individual rate function λ_{uv} is affected by the group initial rate γ_{pq}, which, on the one hand, tends to put similarly behaving individuals into the same cluster; and on the other hand, if one member of a group is heavily influenced by a particular message, it is highly likely that other individuals in the same group will also be affected.

Stability Conditions

For Hawkes processes with constant excitation functions β_{pq}, the sufficient condition of stationarity is $\beta_{pq}\tau_{pq} < 1$, derived from the condition $\int_0^\infty \beta(s)ds < 1$. By contrast, since our β_{pq} is a function of message contents, the expectation of $\lambda(t)$ cannot be time invariant. Therefore, the stationarity condition no longer holds. However, since β_{pq} is evaluated at finite locations (in the domain of message content x), we can define β_{pq}^{MAX} to be the maximum value of β_{pq} across all locations. For our model, we can still require that $\beta_{pq}^{MAX}\int_0^\infty e^{-\frac{u}{\tau_{pq}}}du < 1$. Since $\beta_{pq}^{MAX}\int_0^\infty e^{-\frac{u}{\tau_{pq}}}du = \beta_{pq}^{MAX}\tau_{pq}$, we just need to make sure that $\beta_{pq}^{MAX}\tau_{pq} < 1$.

3.4 Algorithm

We perform posterior inference using Markov chain Monte Carlo method. In our model there is no conjugacy between prior and the likelihood, hence we can not marginalize out parameters and must sample all of them separately. To infer the partition of individuals π, the concentration parameter α, the parameters of each Hawkes process $\theta_{pq} = \{\gamma_{pq}, \tau_{pq}\}$, the training and test point projections of functions r_u and s_v, we use Algorithm 5 in [92] to draw samples from the posterior. We use elliptical slice sampling [93] for r_u and s_v, and standard slice sampling [94] for γ_{pq}, τ_{pq} and α. In case of τ_{pq} we set the upper bound of the slice sampler to $\frac{1}{\beta_{pq}^{MAX}}$, to ensure that $\beta_{pq}^{MAX}\tau_{pq} < 1$.

3.5 Experiments

We compared our model (GHP) to four methods: 1) Poisson process model (Poisson), 2) Hawkes process model (HP), 3) Poisson processes with IRM (Poisson + IRM), and 4) Hawkes processes with IRM (HP + IRM).

3.5.1 Synthetic Dataset Experiments

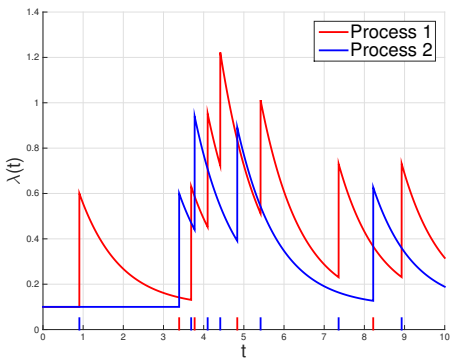
We tested several synthetic data sets under various conditions to compare different model fittings to the rate functions, as well as their clustering behaviors.

A Simple Case Consists of Two Individuals. To generate synthetic data set, we need to set parameter values γ_{uv}, and τ_{uv}, as well as the functional form of $\beta_{uv}(\cdot)$ and message content measure x_{vu}. In figure 3.2, two mutually-exciting Hawkes processes are simulated during time interval $(0, 10]$, where $\gamma_{12} = \gamma_{21} = 0.1$, $\tau_{12} = \tau_{21} = 1$.

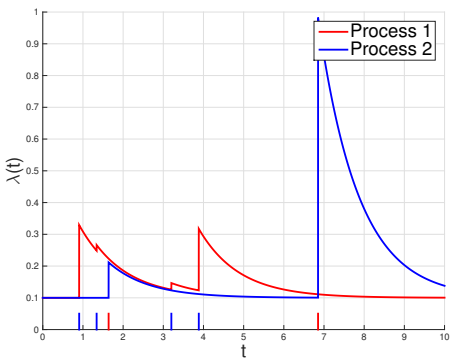
In part (a), case 1 used a constant message content $x_{12}(t_i) = x_{21}(t'_i) = 1$ for all event times t_i and t'_i, and a constant excitation function $\beta_{12}(x) = \beta_{21}(x) = x = 1$ for all messages. Since this synthetic data set has constant β values, it is essentially generated from a HP+IRM; we see that HP+IRM and our model, a generalization to HP+IRM, both perform well, and are better than other models, in terms of log-likelihood shown in table 3.1.

In part (b), case 2 used the same settings as part (a), except for the introduction of variable message content, where both $x_{12}(t_i)$ and $x_{21}(t'_i)$ follow an exponential distribution exp(0.5), which can be thought of as different message entropy values at different event times t_i and t'_i. We see that the jump sizes of both processes are no longer constant. This cannot be modeled by a constant β model, but can only be handled by models like ours, which allow variable β. The effectiveness of our model in this case can be seen from the comparison of the log-likelihoods in table 3.1.

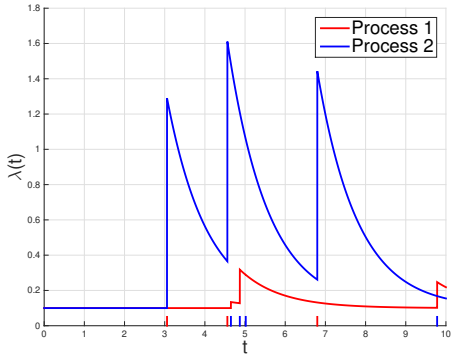
In part (c), case 3 further introduced non-constant $\beta_{uv}(\cdot)$, with all other settings being the same as in case 2, but $\beta_{12}(t_i) = e^{2sin(x_{21}(t_i))+1.5log(x_{21}(t_i))}$ and $\beta_{21}(t'_i) =$



(a) Case 1.



(b) Case 2.



(c) Case 3.

Fig. 3.2. Simulated rate functions of two individuals. In case 1, x is constant, β a simple function $\beta = x$ – the "jump sizes" are constant. In case 2, x is random, β a simple function $\beta = x$ – the "jump sizes" are not constant. In case 3, x is random, β a non-trivial function – the "jump sizes" are not constant.

$e^{0.1cos(x_{12}(t_i'))+0.2\sqrt{x_{12}(t_i')}}$, where $r_1(x_{21}(t_i)) = 2sin(x_{21}(t_i))$, $r_2(x_{12}(t_i')) = 0.1cos(x_{12}(t_i'))$, $s_1(x_{12}(t_i')) = 0.2\sqrt{x_{12}(t_i')}$, and $s_2(x_{21}(t_i)) = 1.5log(x_{21}(t_i))$. Again, the jump sizes for both processes are not constant, and also note that $\beta_{21}(x) > \beta_{12}(x), \forall x \in (0, 10)$. This suggests that process 2 is excited to respond to any messages received from process 1, while process 1 is reluctant to respond to messages sent from process 2. In this case, the difference in log-likelihoods of different models is pronounced even more.

Table 3.1.
Log likelihood comparison for the three-case synthetic dataset.

	case 1	case 2	case 3
Our Model	-21.88	-13.41	-10.86
HP+IRM	-22.97	-35.53	-82.78
Poisson + IRM	-72.31	-89.73	-126.33
HP	-129.37	-238.94	-192.78
Poisson	-127.83	-182.76	-187.23

Next, we will discuss our modeling preferences based on the three-case example used in figure 3.2.

GP Against Simple Parametric Functions. In order to demonstrate the effectiveness of using GP in our model, we compared its performances with simple parametric functions. In table 3.2, we summarize the log likelihood for the three-case synthetic data set mentioned earlier in figure 3.2, using GP and simple polynomials (up to order 3). The results clearly show the superior performance of GP over polynomial functions. The coefficients of polynomials are estimated by sampling from the posterior.

Table 3.2.
Log likelihood comparison between GP and simple parametric functions

	GP	Cubic	Quad	Linear
Case 1	-21.88	-38.67	-38.88	-39.18
Case 2	-13.41	-61.27	-78.17	-89.28
Case 3	-10.86	-71.26	-72.13	-76.73

Estimate Kernel Width From Data. In our experiment, we used the RBF (radial basis function) kernel, which has the form:

$$k(\delta) = \exp\left(-\frac{\delta^2}{2\sigma^2}\right)\left(\tag{3.14}$$

where δ is the distance between two data points, and σ the kernel width. The estimation of the kernel width is crucial in our modeling framework as it controls the complexity of the underlying receptivity and significance functions. We applied 3 different approaches to estimate σ: Bayesian, heuristic, and fixed. The Bayesian approach introduces a prior on σ and obtains an estimate using MCMC; the heuristic way, bearing in mind that sigma largely depends on the maximum distance among the training data, estimates σ directly from sample data distances; and the fixed approach manually assigns a fixed value to the kernel width. It is evident from table 3.3 that the Bayesian approach is the best choice for our model in terms of log likelihood.

Table 3.3.
Log likelihood comparison for kernel estimation using different methods.

	Bayesian	Heuristic	Fixed
Case 1	-21.88	-25.12	-39.78
Case 2	-13.41	-17.16	-18.72
Case 3	-10.86	-22.13	-24.67

Comparison Between Different Information Metrics. We compared four strategies to evaluate the information content of a message: KL divergence of word distribution, message length, TF-IDF, and message Shannon entropy. Using length as the measure of information may not be sufficient in practice; the importance of a message is simply determined by its longevity, without giving any consideration to the content. In case of Shannon entropy, however, the significance and receptivity of the message are better captured. TF-IDF has similar behavior and characteristics as those of message entropy. The best performance in our experiments were given by using KL divergence of word distribution and Shannon entropy, and we preferred KL divergence of word distribution over the other measures because it is more interpretable, and seemed to give consistent good performances in terms of log-likelihoods as shown in table

3.4. However, encoding content information efficiently is still an open question, and certainly a direction for future work.

Table 3.4.
Log likelihood comparison for four different information metrics.

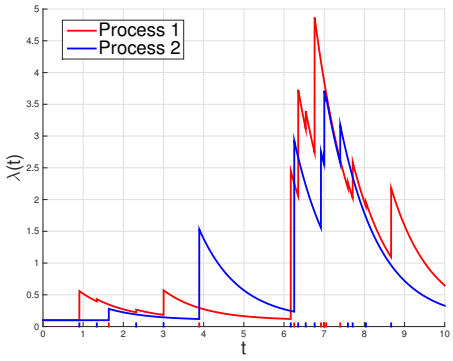
	Word KL	Entropy	TF-IDF	Length
Case 1	-21.88	-21.98	-39.38	-128.76
Case 2	-13.41	-12.78	-28.61	-87.21
Case 3	-10.86	-12.63	-23.78	-72.13

Next, we will discuss a more detailed example consisting of three individuals.

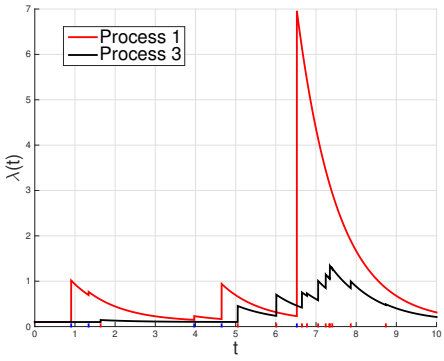
A Full Example Consists of Three Individuals. In this example, we put processes 1 and 2 in one cluster whereas process 3 is in another cluster, and we also intentionally made them behave differently to each other.

The settings we used were $m_{ij} \sim multinomial(p = [0.25, 0.25, 0.25, 0.25], n = 4), \forall i, j \in \{1, 2, 3\}$, which could represent a dialog consisting of only four words, and each m_{ij} can be thought of as the distribution of these four words in a message sent from j to i. We define the message content measure as $x_{ij} = KL(m_{ij}||\bar{m}_i)$, where $\bar{m}_i$ is the four-word distribution assigned to individual i ($\bar{m}_i = (1, 1, 1, 1), \forall i$ in our experiment). For the excitation functions we have: $\beta_{12} = \beta_{21} = 5\exp(1/x)$, $\beta_{23} = \beta_{31} = 0.1\exp(1/x)$, and $\beta_{13} = \beta_{32} = 10\exp(1/x)$. Note that $\beta_{12} = \beta_{21}$, $\beta_{31} < \beta_{13}$, and $\beta_{32} > \beta_{23}$.

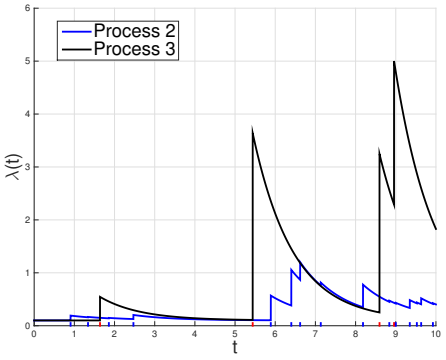
Figure 3.3 (a) shows that processes 1 and 2 are frequently interacting in a similar way, while in part (b), process 3 is not excited to respond to messages from process 1 but tends to, suggested in part (c), reply to process 2's messages more actively. In figure 3.3 (g, h, and i), we see that only our model was able to correctly cluster processes 1 and 2 in the same cluster and put process 3 in a separate one. On the other hand, the other models generated redundant clusters. We have also shown in



(a) processes 1 and 2.



(b) processes 1 and 3.



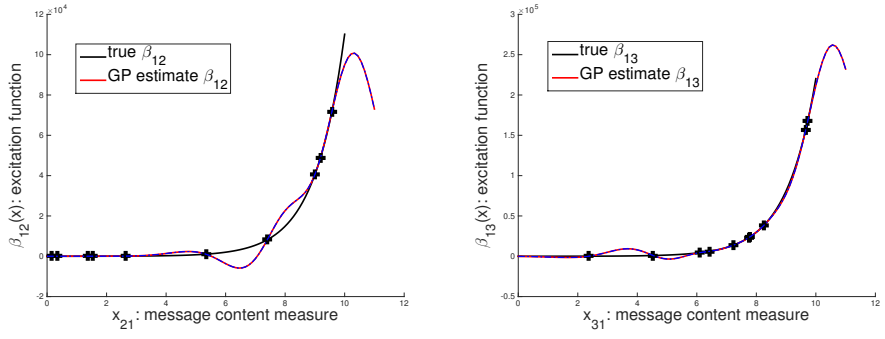
(c) processes 2 and 3.

Fig. 3.3. Simulated rate functions of three individuals and their cluster configurations, where $\beta_{12} = 5\exp(1/x), \beta_{21} = 5\exp(1/x); \beta_{13} = 10\exp(1/x), \beta_{31} = 0.1\exp(1/x); \beta_{23} = 0.1\exp(1/x), \beta_{32} = 10\exp(1/x)$.

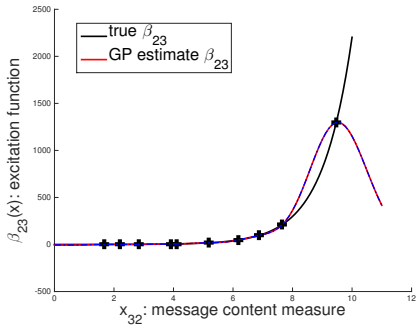
figure 3.3 (d, e, and f) that our model successfully recovered the underlying excitation functions.

3.5.2 Real Dataset Experiments

We tested our model on various turn-taking data sets, which include public meetings, private conversations, email communications, and publication citations. Each data set has several lines of event records, indicated by a quadruplet $(t_i, S_i, R_i, \mathcal{T}_i)$,

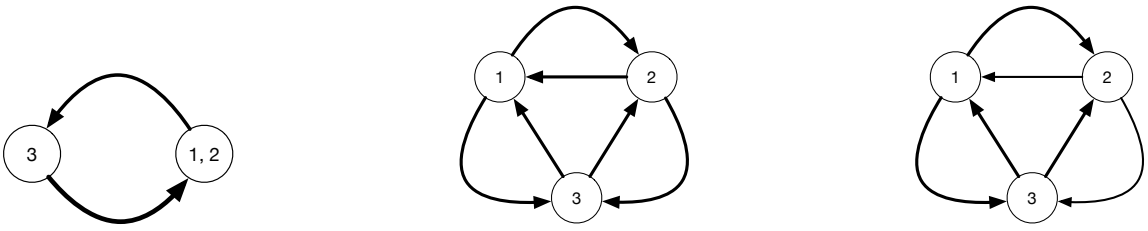


(a) GP plot of β_{12} (b) GP plot of β_{13}



(c) GP plot of β_{23}

Fig. 3.4. GP estimation plots for the synthetic dataset.



(a) HPGP+IRM (b) HP+IRM (c) Poisson+IRM

Fig. 3.5. Underlying clusters inferred by GHP from the synthetic dataset.

where t_i is the time when the event took place, S_i the index of the sender, R_i the index of the recipient, and $\mathcal{T}_i$ the message contents.

We divided the data set into two parts: the first part consists of the first 90% of the data lines, used as the training data set; and the second part contains the remaining 10% of the data lines, used as the testing data set. To compute the average log probability, we ran our code 10 times with different prior settings and computed the mean and standard deviation of the 10 values.

Enron email threads The Enron data set (ENRON) contains about half a million email messages sent or received by about 150 senior managers of the Enron corporation [95]. We restricted ourselves to "true" conversation emails (e.g., auto-messages were ignored) sent and received only from the domain "@enron.com", and identified the threads by time, sender, receiver, and the subject line. The longest email communication was selected.

Santa Barbara Conversation Corpus The Santa Barbara Corpus [96] data set (SB) contains text and video recordings for various conversations. The data set used (#33) is a lively family argument/discussion recorded at a vacation home in Falmouth, Massachusetts. There are eight participants, all relatives or close friends. Discussion centers around a disagreement Jennifer (#2) is having with her mother Lisbeth (#5).

High-energy Physics Theory Citation Network The Arxiv HEP-TH (high energy physics theory) citation data set (CITATION) covers all 352,807 citations of 27,770 papers published during the time period January 1993 to April 2003 (124 months). We converted paper citation events to author citation events. For example, if a paper by authors A and B cited another paper by authors C, D, and E, we would record six events: A cited C, D, and E; and B cited C, D, and E. Only the most cited 17 authors and 97 citation events in the year 2003 were used from this data set.

Results Table 3.5 and 3.6 show, for training and test data sets respectively, the predictive probability results as well as the most probable predictive number of clusters for competing methods. We used 10-fold cross-validation to prevent our

model from being over-fitted to training data sets, and the performances on real data sets suggested a good generalization ability of our model.

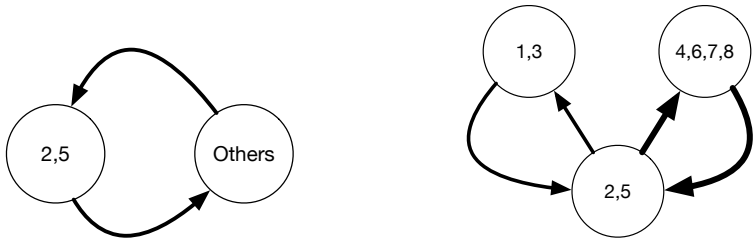
Table 3.5.
Average log likelihood for each model with standard error (training datasets). N is number of individuals, T is number of events, and C the predicted number of clusters.

	Enron	SB #33	Citation
(N, T, C)	(2, 896, 2)	(8, 499, 8)	(17, 97, 17)
Our Model	5612.67 ± 0.13	672.03 ± 0.11	1265.31 ± 0.14
HP + IRM	5513.25 ± 0.12	475.13 ± 0.50	987.34 ± 0.23
Poisson + IRM	2360.37 ± 0.06	572.35 ± 0.11	918.56 ± 0.17

Table 3.6.
Average log predictive likelihood for each model with standard error (test datasets).

	Enron	SB #33	Citation
C	2	2	11
Our Model	327.13 ± 0.02	126.87 ± 0.05	217.51 ± 0.43
HP + IRM	270.36 ± 0.01	89.05 ± 0.04	127.81 ± 0.32
Poisson + IRM	46.21 ± 0.01	13.08 ± 0.00	97.00 ± 0.41

We also compared our model with HP+IRM in terms of cluster detection capability. Figure 3.6 shows the cluster configurations generated by our model and HP+IRM. This dataset is a record of a lively family argument/discussion. There were eight participants, all relatives or close friends, but the main communication was between Jennifer (#2) and her mother Lisbeth (#5). For our model, Jennifer and Lisbeth were put in one cluster, and rest in the other. This is more consistent with data evidence: Jennifer and Lisbeth reciprocate each other more frequently, and



(a) Our Model (b) HP+IRM

Fig. 3.6. Diagram for data set *SB #33*. The thickness of the arrows are proportional to the expectation of the rate function.

respond occasionally to others, despite receiving a lot of messages from them. Individuals other than #2 and #5 may be further decomposed into subgroups, but at this level, the best clustering would probably be the one given by our model. The contrast in the thicknesses of the arrows between the two clusters correctly reveals this aspect. On the other hand, the cluster configuration generated by HP+IRM model indicates a high level of reciprocity, indicated by comparable thicknesses of the two arrows, between clusters {2,5} and {4,6,7,8} which is inconsistent with data evidence. Additionally, the model creates an extra cluster,{1,3}, which is inconsistent with data evidence.

3.6 Related Work

The interest of modeling relational data dates back to at least the work of [1], who introduced the Bayesian formulation of the stochastic block-model. This model was then extended by [14] to the Infinite Relational Model (IRM).

The IRM typically assumes that there is a fixed graph, describing the relationship between individuals, which is observed. This idea is used in many proposed works [3, 14]. Our model does not make this assumption, but learns the relationship among participants' interactions.

There have also been research works modeling relational events via latent classes [97]. They assume each event's sender, receiver, and action type are conditionally independent given the latent class for that event. This strong assumption greatly simplifies the model, but may not reflect real situations. Our model is not limited to any fixed number of action types.

Other works [76–78] are based on temporal Poisson-processes, where the rate of events on each edge is independent of every other edge. Although [76, 77] allow mutually exciting events to be modeled, they do not use content information to model dependencies between events. Our model uses Hawkes processes which are capable of dealing with interaction and reciprocal events, and also use message content information to capture the interactions more accurately.

Our work is also closely related to [25]. They combine mutually exciting Hawkes process with random graph models by defining the excitation function, between a pair of nodes, as a product of a latent binary indicator variable, indicating the presence or absence of edge, and weight variable that determines the strength of interaction between the two nodes. However, unlike our model, their method does not use side information, such as information content, and simply relies on time interaction data to infer latent network structures. Lastly, our work extends the work of [13]. In their paper, the excitation function is not affected by the information content of the message. By introducing GPs, we are able to model non homogeneous excitation functions. In addition to that, since we use Gaussian processes to model the flexible rates of reciprocal activities between two entities, our model can capture the asymmetry in reciprocity more accurately. This, as a by-product, leads to a better cluster detection capability. The model in [34] does not have this leverage.

3.7 Summary

In this chapter, we present a non-parametric Bayesian model that uses Hawkes processes to model reciprocal relationships. Unlike previous approaches, our model

utilizes the content of the messages to model reciprocity. Based on the content, our model captures the significance of the message sent by the sender, and receptivity to the message received by the receiver. This gives us the leverage to model reciprocity in a more realistic manner and more accurately. Empirical results suggest that our novel model formulation can yield improved predictive probability results, and can reveal clusters more accurately than alternative methods.

4. THE MODELLING OF LATENT USER HIERARCHICAL STRUCTURE

In this component of the dissertation [11], we explore how to add user structure to HP models with the nested Chinese restaurant process (nCRP).

4.1 Motivation

In the previous chapter, we see that with the power of GPs in hand, we are able to infer more complex structures behind the communication activities among entities. We also see that the IRM-based models typically assumes that there is a fixed graph independent of the data, describing the relationship between individuals and their roles of actions, e.g., individuals may be assumeed equally likely to be the senders and receivers of a message. This idea is used in many proposed works [3,14]. We aim to not oversimplify realities with this assumption, but instead to learn the senders and receivers non-parametrically based on their interaction histories.

In the second component of the dissertation, we model senders and receirvers in HPs with the nested Chinese restaurant process (nCRP) and call it nCRP-Gaussian-Hawkes process (nCRP-GHP). The motivation of this model comes from the observation that modelling message contents accurately at the individual level and identifying senders and receivers of messages involve exploiting latent hierarchical structure present among users, and nCRP from the nonparametric Bayesian methods is expected to improve the relatively impoverished structure present in earlier works.

4.2 Background

Recall from Equation 1.2 in the introduction section, the observed data $\mathcal{D}$ consists of a sequence of n messages $\mathcal{D} = (\mathbf{y}_1, \cdots, \mathbf{y}_N)$, sorted by their time stamps. Each message $\mathbf{y}_i$ is a quadruplet $\mathbf{y}_i = \{t_i, S_i, R_i, \mathcal{T}_i\}$, where t_i are the time-stamps, S_i the sets of senders, R_i the sets of receivers, and $\mathcal{T}_i$ the content of the messages. Note that we allow multiple senders (e.g., in modeling citation networks) and multiple receivers (e.g., in modeling email data). We are interested in the following tasks:

1. At the node level, we would like to learn a hierarchical clustering $\mathcal{C}$ for all the entities in the network, such that entities in the same cluster share some common features of communication including rates, content, collaborators, audiences, etc.

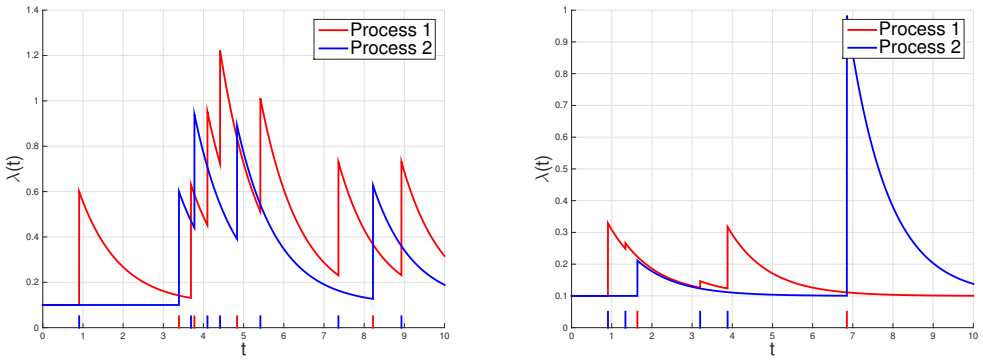
2. At the link level, given previous activity $\mathcal{D}$, we would like to predict the message quadruplet $\mathbf{y}_{N+1}|\mathcal{D} = (t_{N+1}, S_{N+1}, R_{N+1}, \mathcal{T}_{N+1})|\mathcal{D}$, both at the cluster level and at the individual level. Realistic modeling of $\mathcal{T}_{N+1}$ requires sophisticated language models, which is not our focus. Instead, we are interested in demonstrating how incorporating hierarchical structure at the node level significantly improves predictions of message time and content. Accordingly, we limit ourselves to predicting keywords in user messages, rather than detailed message content.

4.3 Model

Since every piece of information in our data is indexed by time, modeling t_i is of central importance. Recall that if we only have one individual, the form of a Hawkes process with an exponential-decay excitation function g is given by:

$$\lambda(t) = \gamma + \int_{-\infty}^{t} \beta e^{-\frac{t-s}{\tau}} \, \mathrm{d}\, N(s) \tag{4.1}$$

The parameter β can be seen as a "jump size" of the rate function whenever a new message is received (see Figure 4.1), and τ indicates the *inverse* rate of decaying.



(a) Constant $\beta's$ (b) Variable $\beta's$

Fig. 4.1. Hawkes process rate functions with constant and variable $\beta's$.

To incorporate text information, we first allow the jump sizes to depend on the message content via a function $\beta : \Re \to \Re$. The function β takes some feature of the message $\mathcal{T}_i$ as input (e.g. the entropy of the message), and determines the size of the Hawkes excitation. We model β with a Gaussian Process (GP) [89]:

$$\lambda(t) = \gamma + \int\!\!\int_{-\infty}^{t} \beta(f(\mathcal{T}_s)) e^{-\frac{t-s}{\tau}} \, \mathrm{d}\, N(s) \tag{4.2}$$

$$\beta(f(\mathcal{T}_i)) \sim \exp(\mathcal{GP}(0, \kappa)) \tag{4.3}$$

where $\mathcal{T}_i$ is the text communicated at t_i, $f(\cdot)$ some transformation that converts text content into numerical measurement, κ the squared exponential kernel of the GP, and the exponential transformation is used to make sure that $\beta(\cdot)$ is non-negative.

While there are many ways to implement the transformation $f(\mathcal{T}_i)$, we propose the following:

1. Calculate TF-IDF scores for each word in the message $\mathcal{T}_i$, so that the sentence is represented by a vector.

2. From their vector representations, calculate distances between pairs of sentences in the message.

3. Use the TextRank [98] algorithm to pick the top sentences and summarize a top-word distribution.

4. Compute the KL-divergence between this top-word distribution and the personalized word distribution of the individual. Effectively, this allows us to quantify how 'relevant' each message is to the receiver.

4.3.1 Modeling Senders and Receivers

Now suppose we have multiple individuals, and a *flat* (one level) clustering $\mathcal{C}$. We define the rate function between two individuals u and v as

$$\lambda_{uv}(t) = \frac{1}{n_p n_q}\gamma_{pq} + \int_{-\infty}^{t} \beta_{uv} e^{-\frac{t-s}{\tau_{uv}}} \, \mathrm{d}\, N_{vu}(s) \tag{4.4}$$

where u and v belong to clusters p and q respectively, and n_p, n_q the number of individuals in clusters p and q. The subscript ordering of N_{vu} (instead of N_{uv}) indicates these Hawkes processes are mutually exciting. Unlike work in [13], which models rates at the cluster level, we model rate functions at the individual level. The benefits of this are three-fold: first, individuals in the same cluster share common behavior through cluster level parameters γ_{pq}; second, unlike cluster-level models (which uniformly pick individuals from a cluster), we explicitly model activity at the individual level; and finally, we need not separately define cluster level rate functions. Instead, the latter can be computed as sums of individual rate functions:

$$\lambda_{pq}(t) = \sum_{p=\pi(u), q=\pi(v)} \lambda_{uv}(t) \tag{4.5}$$

where $\pi(u)$ is the cluster assignment of individual u. To select senders and receivers from clusters, define the *unconditional* cumulative rate of a sender u, and the *conditional* cumulative rate of a receiver v of a message from a set of senders S as

$$\bar{\lambda}_{u\cdot}(t) = \sum_{v} \lambda_{uv}(t), \quad \bar{\lambda}_{\cdot v|S}(t) = \sum_{u \in S} \lambda_{uv}(t). \tag{4.6}$$

Then the probabilities of u and v respectively being selected as one of the receivers and senders are proportional to their cumulative rate ratios:

$$Z_{u \in S} \sim Ber\left(\frac{\bar{\lambda}_{u\cdot}(t)}{\sum_u \bar{\lambda}_{u\cdot}(t)}\right) \tag{4.7}$$

$$Z_{v \in R|S} \sim Ber\left(\frac{\bar{\lambda}_{\cdot v|S}(t)}{\sum_u \bar{\lambda}_{\cdot v|S}(t)}\right) \tag{4.8}$$

where $Z_{u \in S}$ and $Z_{v \in R|S}$ are indicator variables that u and v being selected. The receivers are conditionally picked *after* the selection of senders.

4.3.2 The Overall Model

Recall that at the node level, we would like to learn, not a flat, but a hierarchical tree-like clustering for all the individuals in a network. We model this as a sample from a nested Chinese restaurant process. Conditioned on this tree, it is straightforward to compute all the rates in a *bottom-up* fashion, by summing up the rates, level by level, all the way from the leaf nodes (individuals), using equation 4.5. Based on these rates, senders and receivers can be selected recursively in a *top-down* fashion, using equations 4.7 and 4.8. The generative process of our model works as follows:

1. Sample a clustering tree from the nCRP prior.
2. Based on historical data $\mathcal{D} = (\mathbf{y}_1, \cdots, \mathbf{y}_N)$, compute the rate at the root by summing up over all relevant lower level rates (at the beginning, we only have the base rates γ_{pq}).
3. Simulate a new event time t_{N+1} based on the root rate.
4. Select senders S_{N+1} and receivers R_{N+1} of *each level* of the clustering tree for this new message (the real senders and receivers will be the ones at the leaf level); 4) generate the message text $\mathcal{T}_{N+1}$ from a multinomial distribution based on senders S_{N+1} and receivers R_{N+1} at the leaf level.
5. Finally, update the rate functions of all the receivers. Thus we have generated $\mathbf{y}_{N+1} = (t_{N+1}, S_{N+1}, R_{N+1}, \mathcal{T}_{N+1})|\mathcal{D}$.
6. Repeat steps 2 through 5 with $\mathcal{D} = (\mathbf{y}_1, \cdots, \mathbf{y}_N, \mathbf{y}_{N+1})$.

This can be summarized as (see Figure 4.2):

$$\pi|\alpha \sim nCRP(\alpha) \tag{4.9}$$

$$\lambda_{uv}(t) = \frac{1}{n_p n_q}\gamma_{pq} + \int_{-\infty}^{t} \beta_{uv}(f(\mathcal{T}_s))e^{-\frac{t-s}{\tau_{uv}}} dN_{vu}(s) \tag{4.10}$$

$$\lambda_{pq}(t) = \sum_{p=\pi(u),q=\pi(v)} \lambda_{uv}(t) \tag{4.11}$$

$$M_{new} = \begin{cases} t_{new} \sim HawkesProcess(\lambda_{root}(\cdot)) \\ Z_{u\in S_{new}} \sim Ber\left(\frac{\bar{\lambda}_{u\cdot}(t_{new})}{\sum_u \bar{\lambda}_{u\cdot}(t_{new})}\right) \\ Z_{v\in R_{new}|S_{new}} \sim Ber\left(\frac{\bar{\lambda}_{\cdot v|S}(t_{new})}{\sum_v \bar{\lambda}_{\cdot v|S}(t_{new})}\right) \\ \mathcal{T}_{new} \sim Multinomial(\theta_{S_{new},R_{new}}) \end{cases} \tag{4.12}$$

where nCRP is the nested Chinese Restaurant Process, and $\beta_{uv}(f(\mathcal{T}_i)) \sim \exp(\mathcal{GP}(0, \kappa_{uv}))$. The texts are generated from multinomial distributions whose parameters depend on the senders and receivers: We add and normalize the individual word distributions of the senders and receivers and use the aggregated one for the multinomial distribution.

This model is an nCRP extension to the Gaussian Hawkes process (GHP) model we described in the previous chapter, and we call the new model nCRP Gaussian Hawkes process (nCRP-GHP) for short.

4.4 Algorithm

For our model, the inference problem is nonparametric and non-convex, and there is no conjugacy between the priors and the likelihood functions. We therefore adopt and extend the inference framework from [13] and [10], which performs posterior inference using MCMC sampling. The state space of the model is defined over $\{\pi_u, \gamma_{uv}, \tau_{uv}, \beta_{uv}, \theta_u\}$, and the conditional distributions used in the MCMC algorithm can be obtained based on section 4.3.2. The sketch of the algorithm can be described as follows: 1) Initialize the state variables by sampling from their priors. 2) Until convergence, iteratively and sequentially sample each state variable conditioned on

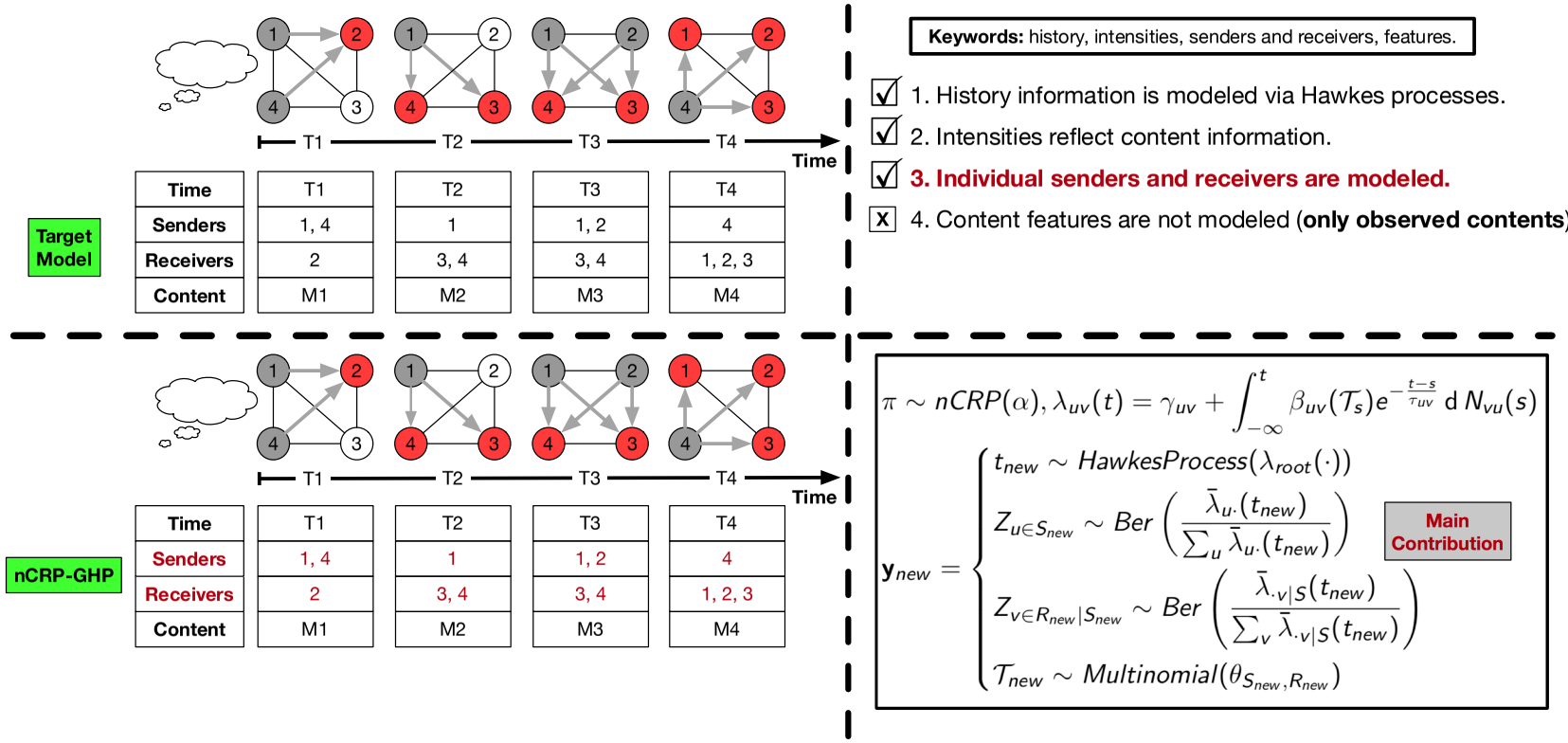


Fig. 4.2. An illustration of the nested Chinese Restaurant Gaussian Hawkes processes (nCRP-GHP) model, where the senders and receivers are explicitly modeled based on a hierarchical tree structure from the nCRP.

the current state of all other variables – sample π_u using the standard Gibbs sampling algorithm [81]; sample $\{\theta_u, \gamma_{uv}, \tau_{uv}\}$ using slice sampling [94]; sample β_{uv} using elliptical slice sampling [93].

For a dataset of V individuals, N messages, and K top words, the number of model parameters is $\mathcal{O}(V^2)$, and the computational cost at each iteration is $\mathcal{O}(NV^2K^3)$. One of the bottlenecks of the algorithm comes from the inference of the GP related parameters β_{uv}, which costs $\mathcal{O}(K^3)$, where K is the number of top words. To ameliorate this situation, we restrict K to be a reasonably small number in our experiments, e.g., $K = 20$. We also want to point out that, at each iteration, not all of the $O(V^2)$ parameters are updated or used to update other parameters. For example, after an update of π_u, only the affected individuals and clusters should be considered – which is usually a small subset of the population in practice.

4.5 Experiments

We compare our model with four existing models (discussed in sections 4.1 and 4.6): nCRP+HP, GHP, IRM+HP, and HP. Recall that IRM stands for the infinite relational model, HP for the Hawkes process and GHP for the Hawkes process with a Gaussian process controling jumps. We first present experimental results based on synthetic data, which focus on quantitative analysis of model performance as well as qualitative discussions of model effectiveness. We then explore some of the findings from real data using our model. The observed data $\mathcal{D}$ used in this section has the same format, consisting of a sequence of messages $\mathcal{D} = (M_1, \cdots, M_n)$, sorted by their time stamps. Each message M_i is a quadruplet $M_i = \{t_i, S_i, R_i, \mathcal{T}_i\}$, where t_i is the time-stamp, S_i the set of senders, R_i the set of receivers, and $\mathcal{T}_i$ the text content of the message. $\mathcal{D}$ is divided into three segments: the first 80% the training set, the next 10% the validation set, and the last 10% the test set. To compute the average log probability, we run each experiment ten times with different prior settings and report the credible interval based on their means and standard deviations.

4.5.1 Synthetic Dataset Experiments

Following the generative process described in section 4.3.2, we simulate 1000 message communications among 7 individuals (shown in Figure 4.3). The clustering tree has two levels, $\{\#1, \#2, \#3\}$ are in cluster 1 (red), $\{\#4, \#5\}$ in cluster 2 (green), and $\{\#6, \#7\}$ in cluster 3 (blue). The initial rate γ at the root is set to 1, and this is distributed among its offspring proportional to their cluster sizes. The inverse decay rates τ_{uv} are set to 0.1 for all pairs of u, v. The "jump size" function is taken to be an exponential $\beta(x) = \exp(x)$. The vocabulary of the synthetic corpus we used consisted of the top 10,000 words from the Neural Information Processing Systems (NIPS) dataset (consisting of 5811 papers published during the years 1987 to 2015). We generate 1000 messages, each containing 20 words. The personalized distributions over the 10,000 words of the seven users are randomly generated through a Dirichlet distribution, the concentration parameters of which are drawn from a Dirichlet prior with uniform concentration parameters.

Predictive log-likelihood. We compare our method with the alternatives, showing results in Table 4.1. We see that our model achieved the best performance in terms of predictive log-likelihood. This is not surprising, given that the data is generated from the model.

Table 4.1.
Our model against other models. Log-likelihoods with standard deviations (10 runs).

	Predictive Log-likelihood
Our Model	312.89 (± 12.37)
nCRP + HP	221.97 (± 10.16)
GHP	207.63 (± 13.28)
IRM + HP	197.23 (± 16.12)
HP	101.01 (± 16.12)

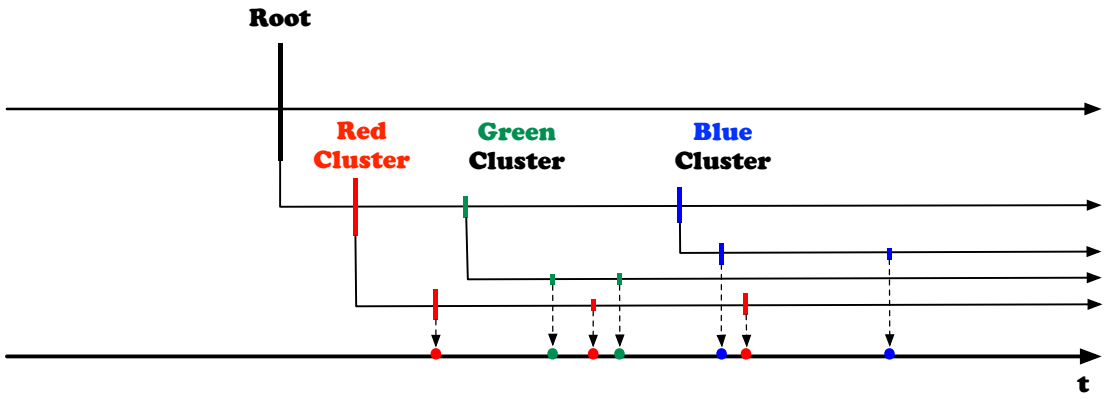


Fig. 4.3. Illustration of the synthetic data. The clustering tree has two levels (root is at level 0): the first level consists of three clusters (red, green, and blue), and at the second level each of the cluster has several individuals (red cluster has 3 individuals, green has 2, and blue has 3). Individuals *receive* messages (represented by color dots) at different times, which bump the rate functions of individuals (represented by color bars) by a certain amount (decided by the GPs). The heights of the bars at the cluster level and at the root illustrate the aggregate effect from lower level rates.

The three main components of our model are: 1) GP to model varying "jump sizes"; 2) nCRP for hierarchical clustering; and 3) senders and receivers to model personalized textual information. We investigate the effectiveness of these model characteristics.

Usefulness and identifiability of the GPs. In Table 4.1, we already see that our model had better log-likelihoods compared to nCRP+HP, suggesting that including the GPs helps our models overall predictive performance. Here, we take a closer look at the actual fit of each GP compared to the ground truth (the exponential). Shown below are the GP plots of the first three of the seven individuals (along with the truth), showing the ability of the GPs to recover the underlying "jump size" function $\beta(x)$.

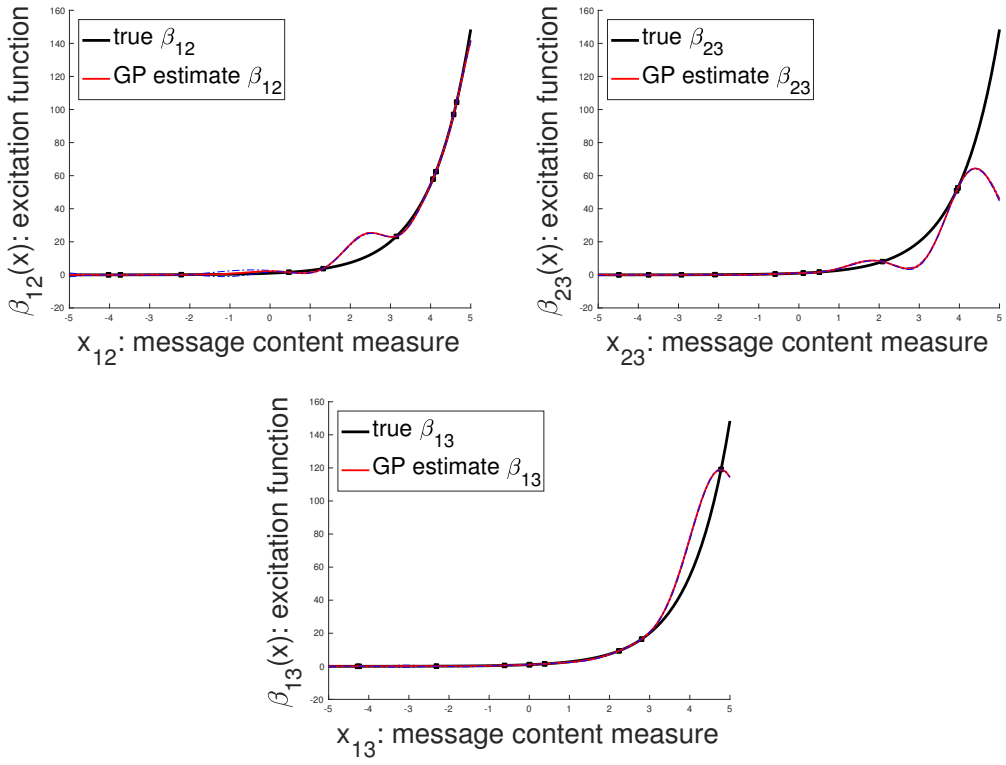


Fig. 4.4. GP plots of β_{12}, β_{23} and β_{13}. The underlying "jump size" function is taken to be an exponential $\beta(x) = \exp(x)$.

Table 4.2.
Sampled trees against manual trees. Log-likelihoods with standard deviations (10 runs).

	Predictive Log-likelihood
(nCRP) sampled tree	312.89 (± 12.37)
(correct) manual tree	321.92 (± 7.86)
(wrong) manual tree	126.27 (± 21.63)
no tree	179.61 (± 9.17)

Effect of nCRP for modeling hierarchical clustering structure. We compare our model with two manually designed trees: one being the true underlying tree; the other being an incorrect tree that puts all 7 individuals in one single cluster. Our model which samples trees from nCRP prior recovers the tree structure, and from Table 4.2 we see that it obtained very similar predictive log-likelihood as that of a correct manual tree, compared to the much worse performance from a wrong manual tree. The correct manual tree achieves smaller standard deviation over 10 experiment runs, which is what we expected since the fixed tree reduces randomness of the model. It is also clear that ignoring the tree results in poor predictive log-likelihood.

Benefits of including senders and receivers. One of the advantages of introducing senders and receivers is the ability to generalize the thinning procedure in Hawkes processes. In the existing literature e.g., [13], uniform thinning is a popular choice. That is, a new message is assigned to an individual with equal probability. Our model on the other hand can assign a message to its senders and receivers based on 1) its event history via the HPs; 2) text information via the GPs; and 3) collaborator and audience via the nCRPs.

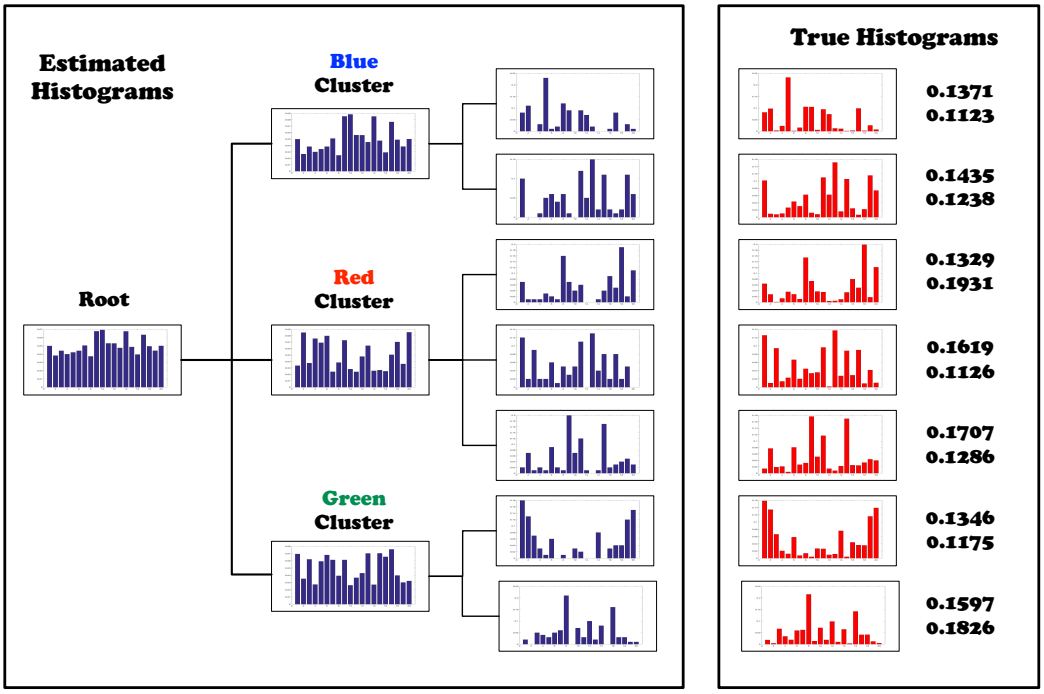


Fig. 4.5. Posterior keyword distributions of synthetic dataset. The first numbers are the *estimated* word distributions at each node on the nCRP tree; and the second numbers are the *true* word distributions, together with their L_1 distances (against top 20 words and 20,000 full vocabulary).

To demonstrate these benefits, the final experiment on synthetic data focuses on learning the posterior keyword distributions of individuals, which may be used to suggest personalized favorite words, and in turn decide the authorship and audiences of the new messages.

The leaf nodes in Figure 4.5 shows the posterior keyword distributions of the seven individuals. The cluster level keyword distribution is aggregated from its members' distributions (top words of the union of top words), and the root keyword distribution is aggregated from the cluster ones. Thus, the top words in each histogram may not be the same. We also notice that at the root, the words are almost uniformly distributed, which suggests that the most important words across all individuals are almost of the same importance. We may use these top words to identify clusters.

4.5.2 Real Dataset Experiments

We apply our method to three different real datasets:

1. **NIPS Dataset [99].** This contains the counts of 11,463 words appearing in the 5,811 papers published in the conference Neural Information Processing Systems (NIPS) during the years 1987 to 2015. Authors and citations are obtained through the paper IDs. We treated authors as message "senders", and cited authors as "receivers".

2. **Facebook Dataset.** This data contains Facebook message communications among 20,603 individuals. We pick the top 10 individuals based on their number of friends, and add in their 1st and 2nd connection friends (376 in total).

3. **Santa Barbara Corpus Dataset [96].** The Santa Barbara Corpus [96] dataset (SB) contains text recordings for various conversations. The data we use (#33) is a lively family discussion recorded at a vacation home in Falmouth, Massachusetts. There are eight participants, all relatives or close friends. Discussion centers around a disagreement that Jennifer (#2) is having with her mother Lisbeth (#5).

Predictive log-likelihood. We evaluate our model performance in terms of predictive log-likelihood, and present our findings about keywords and clusters. For all of these three datasets, the predictive log-likelihoods of our model constantly outperform existing alternative methods.

Next, we show the effectiveness and consistency of our model, i.e., what our model can do with different types of datasets and whether or not it gives us consistent performance under different scenarios.

Exploratory analysis. *1) Identifying clusters and learning interesting community features.* Figure 4.6 shows the posterior word distribution at the root node for the **Facebook dataset**. The size of each word is proportional to its "importance",

Table 4.3.
Model comparison on the real datasets. The numbers reported in each cell are the log-likelihoods for training, validation, and test set (in bold), respectively.

	NIPS Dataset
Our Model	9708.23, 1297.83, **1127.21**
nCRP+HP	9026.78, 1028.36, **997.82**
GHP	8934.67, 1186.22, **1128.76**
IRM+HP	4896.17, 567.18, **682.70**
HP	3490.78, 518.70, **683.18**
	Facebook Dataset
Our Model	1208.37, 199.12, **218.93**
nCRP+HP	992.70, 181.11, **178.86**
GHP	1118.61, 175.81, **182.49**
IRM+HP	928.14, 128.76, **129.83**
HP	312.78, 59.08, **61.93**
	Santa Barbara Dataset
Our Model	491.37, 118.12, **109.82**
nCRP+HP	391.87, 96.24, **99.68**
GHP	438.71, 101.83, **97.20**
IRM+HP	412.98, 81.87, **52.73**
HP	303.82, 59.83, **70.23**

based on the TF-IDF scores. We see that: firstly, the sizes are quite uniform, agreeing with our findings from synthetic data analysis; and secondly, the words with highest "importance" are "happy" and "birthday", confirming the 'viral" nature of mutually-exciting Hawkes processes. We also summarize the sizes of the first two clusters, as well as top 3 words of each cluster. Cluster 1 has 128 individuals, with top 3 keywords {*workout, class, homework*}; Cluster 2 has 95 individuals, with top 3 keywords {*time, work, break*}. We suggest that cluster 1 is more about study and school life, cluster 2 is more about work and related activities.

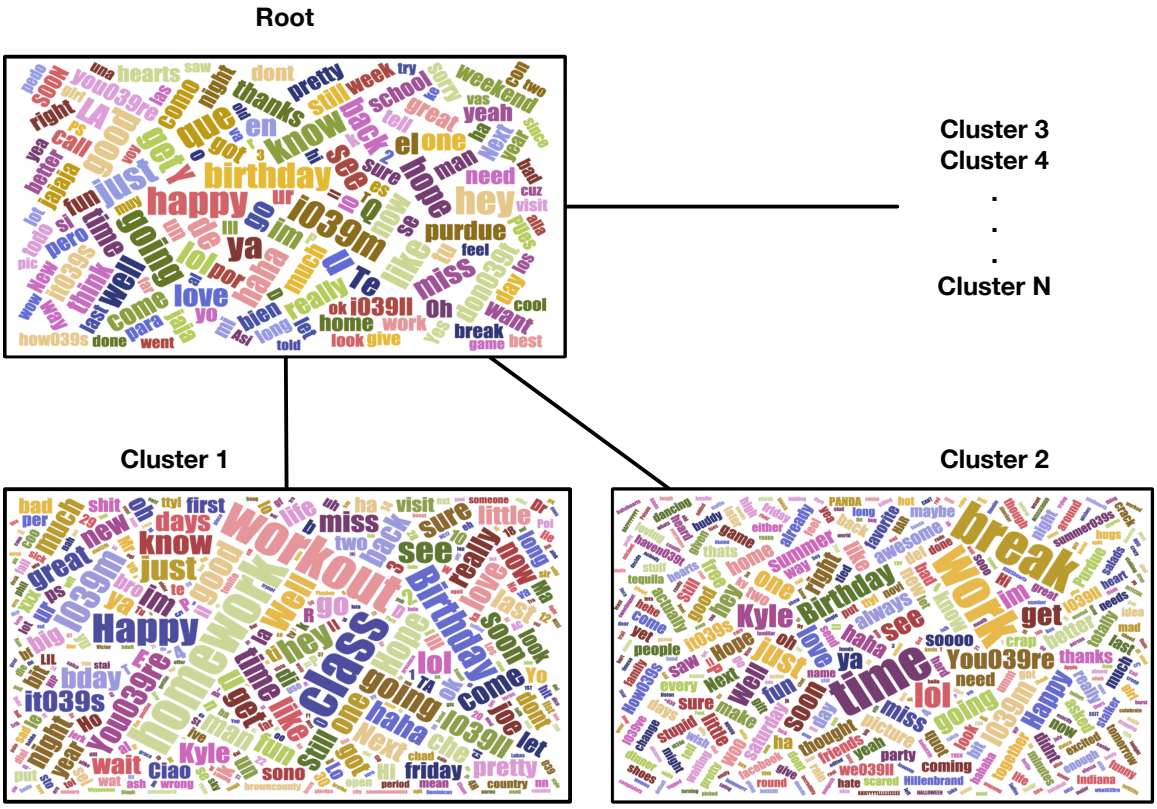


Fig. 4.6. Facebook data WordCloud.

2) Predicting preferences of senders/receivers within each cluster. Shown below are the predicted collaborators and keywords of three selected top authors (in terms of number of papers and citations) from the **NIPS dataset**.

- *Y. Bengio (+ G. Hinton, Y. LeCun):* deep learning, neural network, data, machine learning, features, gradient.
- *Z. Ghahramani (+ M. Jordan, D. Blei):* neural network, kernel, variational, probabilistic, Gaussian processes, regression.
- *Y. LeCun (+G. Hinton, Y. Bengio):* generative, embedding space, auto-encoder, supervised.

This clearly aligns with what we know about the authors' research interests. These predicted preferences of individuals play an important role in deciding the authorship and patterns of future communications.

3) Interpret individual behavior via quantifiable evidence. Figure 4.7 shows the rate function plots of two clusters from the **Santa Barbara dataset**: Jennifer and her mother Lisbeth, and the rest of the people. We see that there is a trend that whenever topic 1 (between Jennifer and her mother Lisbeth) is active, topic 2 tends to become silent. This phenomenon is clearly observed during (normalized) time frame 70 to 90. The actual transcript of this conversation shows that this was one of the occasions when Jennifer and Lisbeth were arguing with each other. It is even clearer when we look closer at the rate functions at the individual level. Figure 4.7 implies that Jennifer and Lisbeth's individual rate functions are complement to each other.

Learning parameters with an incorrect tree. To evaluate the importance of jointly learning the tree structure from the data, we shuffle the tree and re-learn the parameters and compare the log-likelihoods as follows: 1) Learn a tree $\mathcal{T}$ from the model; 2) shuffle nodes to obtain a new tree $\mathcal{T}'$; and then 3) use $\mathcal{T}'$ and re-learn the parameters. Repeat the process ten times and report mean and standard deviation.

In table 4.4, our model outperform the ones without a tree and shuffled-trees, and the more we destroy the structure of the tree, the worse the model performance.

Table 4.4.
Log-likelihood comparison after shuffling the tree from the model, under different depth. The numbers reported in each cell are the log-likelihoods for training, validation, and test (in bold) datasets, with their standard deviations, respectively.

	NIPS Dataset
model	9708.23, 1297.83, **1127.21**
without a tree	8934.67, 1186.22, **1128.76**
bottom 1 level	3790±130.1, 489±79.8 **414±27.3**
bottom 2 levels	1279±189.7, 316±88.6, **316±28.7**
bottom 3 levels	997±212.8, 283±107.7, **278±30.6**
	Facebook Dataset
model	1208.37, 199.12, **218.93**
without a tree	1118.61, 175.81, **182.49**
bottom 1 level	216±29.78, 37±7.63, **67±9.82**
bottom 2 levels	186±31.78, 21±9.27, **51±10.67**
bottom 3 levels	121±36.15, 21±10.62, **45±12.19**
	Santa Barbara Dataset
model	491.37, 118.12, **109.82**
without a tree	438.71, 101.83, **97.20**
bottom 1 level	278±12.96, 79±9.71, **87±7.12**
bottom 2 levels	212±9.18, 71±12.38, **72±10.37**
bottom 3 levels	217±18.92, 68±17.92, **67±16.84**

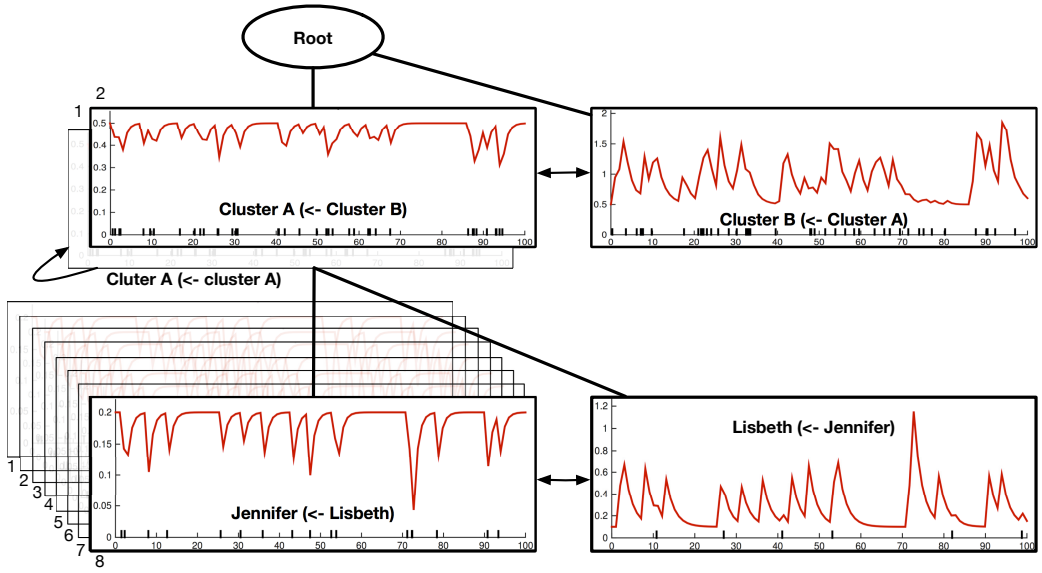


Fig. 4.7. Rate function plots of the SB data at the cluster level: {A: Jennifer and Lisbeth} and {B: Others}; and individual level. At the individual level, there are eight rate functions associated with each person (only shown Jennifer in the plot), including the one with him/herself. Cluster rates are aggregations of individual rates, as defined in equation 4.5.

This confirms that our model's superior performance is not because of the additional parameters from the tree: it is the tree structure itself that is important.

Model comparisons. For each real dataset, we divide the dataset into 10 equal-length pieces $D_1, D_2, \cdots, D_{10}$, and then perform an increasing-size training strategy: use D_1 to train the model and test on D_{10}; use D_1 and D_2 for training and test on D_{10}; and so on, until finally, train model using $D_1, \cdots, D_9$ and test on D_{10}. The results in figure 4.8 suggest that our model consistently outperform other models being compared, especially the ability to learn better at the early stage with relatively small amount of data. For large amounts of data, the model without the tree structure performs comparably, explaining some of the results in Table 4.

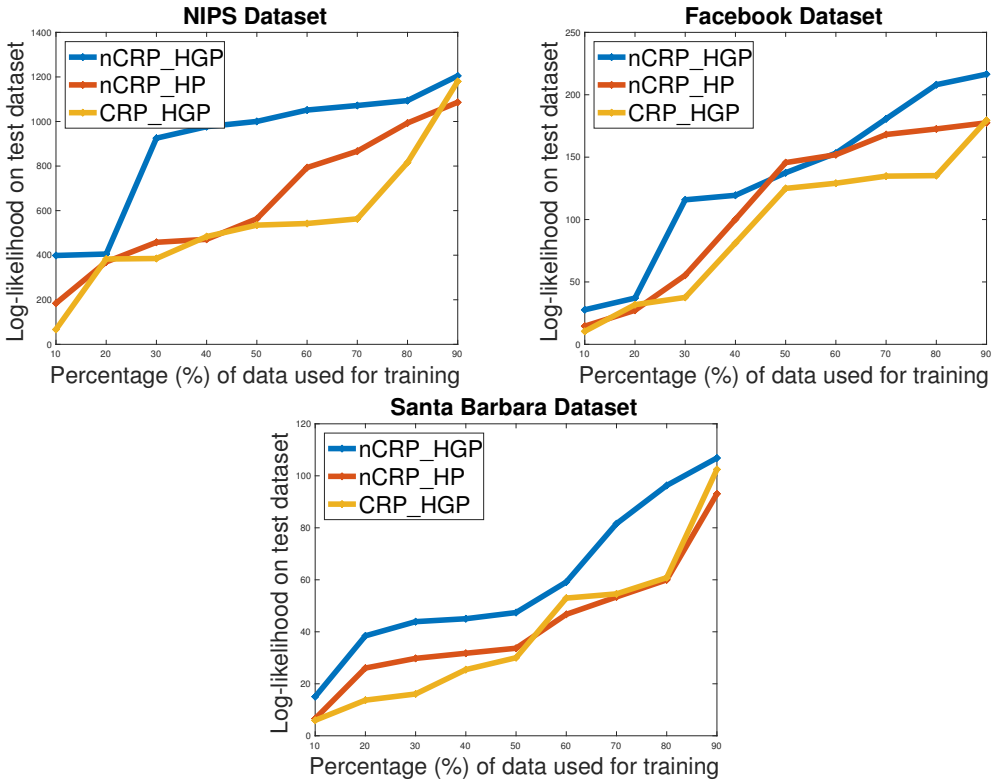


Fig. 4.8. Log-likelihood comparison on test datasets with increasing-size training data.

4.6 Related Work

The closest existing work to our model are [10, 37, 79], though none of these explore hierarchical clusterings of senders and receivers with Hawkes processes. The nested Chinese restaurant franchise process model of [79] combines ideas from the hierarchical Dirichlet process (HDP) [80] and the nested Chinese Restaurant Process (nCRP) [81] to allow each object to be represented as a mixture of paths over a tree, and to decouple the task of modeling hierarchical structure from that of modeling observations. The work of [37] connects Dirichlet processes and Hawkes processes to allow the number of clusters to grow while at the same time learning the changing latent dynamics governing the continuous arrival patterns. Our model extends these works, which has a hierarchical structure embedded with temporal point processes.

Recently, [25, 31, 33, 41, 57, 58, 63] proposed different models to address similar problems. However, while we define each observed message $\mathbf{y}_i$ as a quadruplet $\mathbf{y}_i = \{t_i, S_i, R_i, \mathcal{T}_i\}$, these previous works, in our opinion, all missed some important aspects of the information. The loss of these may result in ineffectiveness of modeling personal level details. For example, [31] modeled $\mathbf{y} = \{t, S, R\}$, [33, 57, 63] modeled $\mathbf{y} = \{t, S, T\}$, and [41] modeled $M = \{t, S\}$ and the cluster $\mathcal{C}$. Our work explicitly treats senders $\{S_i\}$ and receivers $\{R_i\}$ as important components of the model, which greatly extends the existing methods in the literature and enables inference about authorship and audience of communications, as well as their favorite collaborators and top-pick words.

Moreover, we focus on different modeling perspectives, specifically, (1) modeling mutually-exciting transactions between users (e.g., email communications) rather than individual self-exciting actions of users (e.g., purchases/clicks), and (2) modeling personalized textual content between pairs of users (with a continuous metric), rather than modeling individual topics/tasks (with a discrete metric). While topics/tasks can be viewed as discrete labels of the "content" of activities (and it is meaningful to use this concept in cases such as web activities), in the context of com-

munications/transactions, the content being communicated is highly personalized, a continuous metric affords more flexibility to make better use of it.

4.7 Summary

In this chapter, we have established a novel and unified framework combining the advantages of Bayesian nonparametrics and temporal point processes to model not only the temporal (t_i) and textual ($\mathcal{T}_i$) information of the messages being communicated in a network, but also the senders (S_i) and receivers (R_i) who are involved in the communications. Empirical results suggest that our novel model formulation can provide with improved predictions about event times, clusters, etc. In addition, our method offers inference about authorship and audience of communications, as well as their personal behavior such as their favorite collaborators and top-pick words, which greatly extends the existing methods in the literature.

5. THE INTERPLAY BETWEEN TEMPORAL DYNAMICS AND CONTENT FEATURES

In this component of the dissertation, we explore the interplay between the temporal and textual dyanmics in the HP models with the Indian buffet process (IBP).

5.1 Motivation

So far, we have seen how content information can drive the rate dynamics of HPs, but in turn, it is also reasonable to argue that contents can be influenced by the temporal dynamics of communications as well.

Latent feature models (both parametric and nonparametric) have found wide application in settings where exchangeability holds. A canonical model from Bayesian nonparametric methods is the IBP. While there has been some work towards relaxing exchangeability assumptions to allow for temporal dynamics, modeling the full richness of interactions remains an open challenge.

Our main contribution in the model (IBHP) of this chapter is a framework that facilitates the modeling of temporal abd textural dynamics through a combination of ideas from the IBP with those in the HP.

5.2 Background

The standard Hawkes process has a number of limitations pertinent to the problem we are considering:

1. Each event is triggered by a single observation instead of possibly multiple ones (see Figure 5.1), which can be better seen in its branching representation.

2. The way each event is triggered does not depend on its history events, which implies that the excitation kernels are independent of the past observations.

3. Rich temporal dependency structures are not captured for the latent features of events.

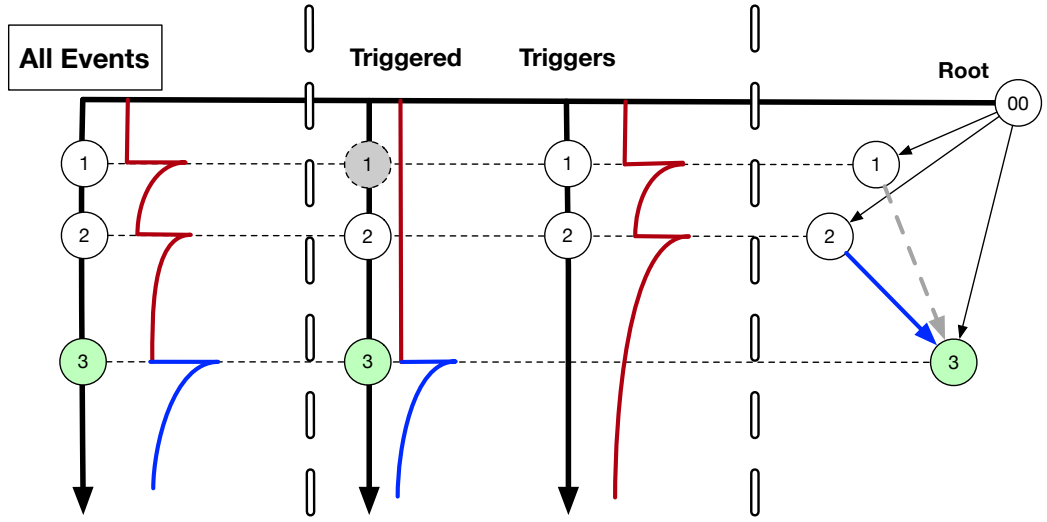
As a result, previous HP models do not capture the rich dynamics of real-world activity—which can be driven by multiple latent triggering factors shared by past and future events, with the latent features themselves exhibiting temporal dependency structures.

For instance, rather than view a new document just as a response to other documents in the recent past, it is important to account for the factor-structure underlying all previous documents. This structure itself is not fixed, with the influence of earlier documents decaying with time.

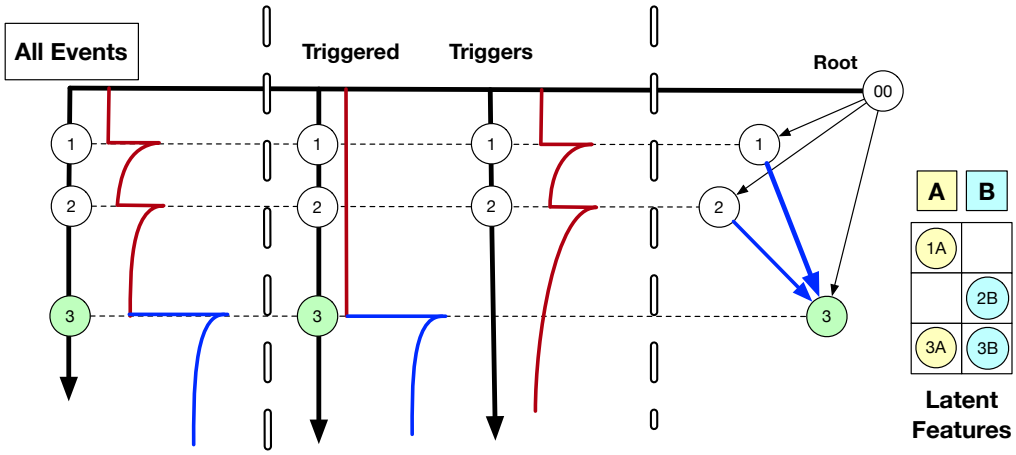
To this end, we propose a novel Bayesian nonparametric stochastic point process model, the Indian Buffet Hawkes Processes (IBHP) [12], to learn multiple latent triggering factors underlying streaming document/message data (see Figure 5.2 for the framework). The IBP facilitates the inclusion of multiple triggering factors in the HP, and the HP allows for modeling latent factor evolution in the IBP. We develop an efficient and scalable learning algorithm for the IBHP based on Sequential Monte Carlo and demonstrate the effectiveness of the model, both quantitatively and qualitatively, in experiments on synthetic and real data.

5.3 Model

We propose the IBHP, which can be viewed as a nonparametric latent state space model, where past events $\mathbf{y}_i = \{t_i, \mathcal{T}_i\}$ influence future observations through latent state variables $\mathbf{z}_i = \{\mathbf{K}_i, \mathbf{V}_i\}$ (described below). The $\mathbf{z}_i$'s summarize information about the past, and themselves evolve following dynamics based on the IBP. Algorithmically, the generative process can be described in the following three steps (see Algorithm 1 for details).



(a) A Hawkes process with single triggers.



(b) A Hawkes process with multiple triggers.

Fig. 5.1. HP with single and multiple triggers. In (a), #3 is triggered by a single event #1, while in (b) it is triggered by #1 and #2. The triggering kernels can be quite different depending on how the triggering has happened. HP with single triggers would fail to model influences from both #1 and #2 at the same time, as shown in (b).

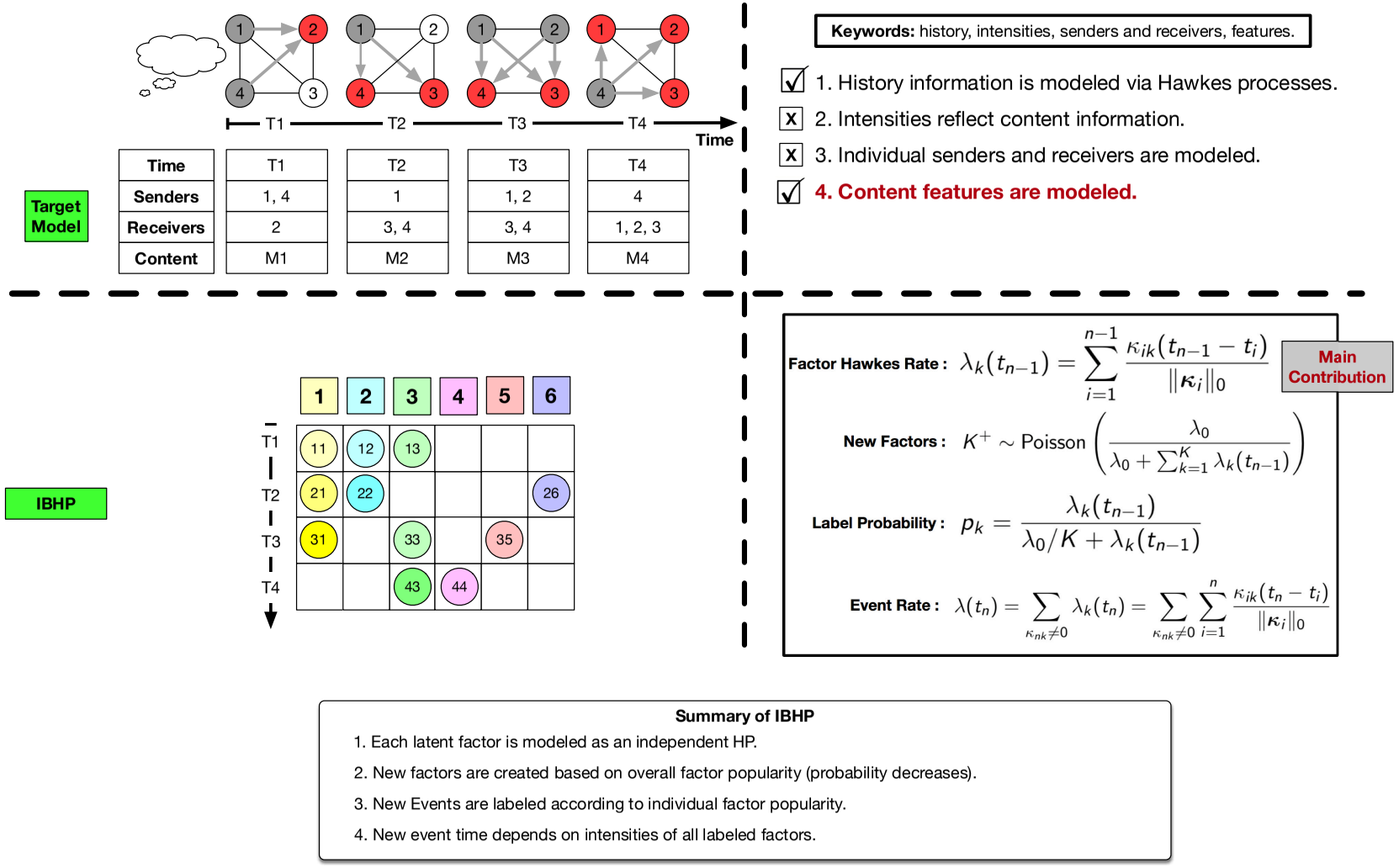


Fig. 5.2. An illustration of the Indian Buffet Hawkes Process (IBHP), which models the interplay between the textual and temporal dynamics in order to learn the latent feature represention.

5.3.1 Initilization

To setup the model, we first specify a triplet $\mathcal{M} = \{\mathcal{S}, D, L\}$: the dictionary $\mathcal{S}$, representing the vocabulary of all possible words in the observations, the text length D of each document, and the number of basis kernels L. We also require a pair of hyper parameters $\mathbf{\Pi} = \{\mathbf{w}_0, \mathbf{v}_0\}$ for the priors of the kernel and word distribution weights.

Each latent factor influences the content of future events through a set of *dictionary weights*, which are used to generate text, i.e., $\mathbf{v}_k$ is a vector of weights (of length $|S|$, which sums to one) for the k^{th} factor. The weights $\mathbf{v}_k$ are sampled from a Dirichlet prior (with hyper parameter $\mathbf{v}_0$) whenever a new factor is created (see later).

Each latent factor also influences the timing of future events through a *triggering kernel*, and we assume each kernel is a linear combination of a set of L bases. Throughout, we assume L *exponential* basis kernels:

$$\gamma_l(\delta) = \beta_l e^{-\frac{\delta}{\tau_l}}, \quad l = 1, \ldots, L. \tag{5.1}$$

This requires a set of parameters $\{(\beta_l, \tau_l)\}$, each of which captures a distinct type of excitation pattern. A binary matrix $\mathbf{C}$ indicates which factors are associated with each observation. The k^{th} *factor kernel* for the i^{th} observation κ_{ik} is a weighted sum of the L basis kernels:

$$\kappa_{ik}(\delta|\mathbf{w}_k, c_{ik}) = \begin{cases} \sum_{l=1}^{L} w_{kl} \cdot \gamma_l(\delta), & \text{if } c_{ik} = 1 \\ 0, & \text{if } c_{ik} = 0 \end{cases} \tag{5.2}$$

where the weights w_{kl} are sampled from a Dirichlet prior (with hyper parameter $\mathbf{w}_0$) whenever a new factor is created (see later). Thus, immediately after an event (when $\delta = 0$), there is a jump in the event rate with amplitude equal to $\kappa_{ik} = \mathbf{w}_k^{\intercal}\boldsymbol{\beta}$. Observations with the same factor share the factor kernel. We write the model parameters as $\mathbf{\Theta} = \{\lambda_0, \{\beta_l\}, \{\tau_l\}\}$, where λ_0 is a base-rate at which events happen spontaneously.

5.3.2 The First Event

To generate the observation $\mathbf{y}_1 = \{t_1, \mathcal{T}_1\}$, we first sample the auxiliary variables $\mathbf{c}_1$ and $\mathbf{w}_{1:K}$.

The *factor label variable* $\mathbf{c}_1$ is a binary vector of length K, where $K \sim \text{Poisson}(\lambda_0)$ is the number of existing factors. $c_{nk} = 1$ implies that the n^{th} observation has a label of factor k. Set $c_{1k} = 1$ for $k = 1, \ldots, K$.

The *kernel weights* $\mathbf{w}_k$ is a vector of weights for the k^{th} factor to load the basis kernels. Each $\mathbf{w}_k$ is of length L (the number of basis kernels), and sum to one. Sample $\mathbf{w}_k \sim \text{Dir}(\mathbf{w}|\mathbf{w}_0)$ for $k = 1, \ldots, K$.

Given the values of $\mathbf{c}_1$ and $\mathbf{w}_{1:K}$, we can sample the associated latent variables $\mathbf{z}_1 = \{\mathbf{K}_1, \mathbf{V}_1\}$. Define the $1 \times K$ *IBHP matrix* $\mathbf{K}_1$, whose rows are $\boldsymbol{\kappa}_1$, with values $\mathbf{w}_k^\intercal \boldsymbol{\beta}$ (see Equation 5.2) – since $\delta = 0$. For $n = 1$, sample $\mathbf{v}_k \sim \text{Dir}(\mathbf{v}|\mathbf{v}_0)$ for $k = 1, \ldots, K$, and define the $|\mathcal{S}| \times K$ matrix $\mathbf{V}_1$, whose columns are $\mathbf{v}_k$.

Conditioned on these state variables $\mathbf{z}_1$, we sample the first observation $\mathbf{y}_1 = \{t_1, \mathcal{T}_1\}$: The *time stamp* t_1 is sampled from a Poisson process with rate λ_0; and the *text* $\mathcal{T}_1$ is sampled from $\text{Multi}(D, \sum_{k=1}^{K} \mathbf{v}_k / K)$, where the weight parameter is the averaged factor weight of the first observation.

5.3.3 Follow-up Events

Conditioning on $\mathbf{z}_{n-1}$, suppose there are K existing factors, each of which can be represented by an independent Hawkes process. At time t_{n-1}, the *factor rate* is:

$$\lambda_k(t_{n-1}) = \sum_{i=1}^{n-1} \frac{\kappa_{ik}(t_{n-1} - t_i)}{\|\boldsymbol{\kappa}_i\|_0} \tag{5.3}$$

As with the generation of the initial event, follow-up events ($n > 1$) are also generated by two steps. First, we sample the auxiliary variables $\mathbf{c}_i$ and set $\mathbf{w}$ and $\mathbf{v}$ for any newly generated factors.

The first K components of the *factor label variable* $\mathbf{c}_i$ is sampled independently from a Bernoulli distribution with probability parameter

$$p_k = \frac{\lambda_k(t_{n-1})}{\lambda_0/K + \lambda_k(t_{n-1})} \tag{5.4}$$

Meanwhile, K^+ new factors are created by setting $c_{nk'} = 1$, for $k' = K + \{1, \ldots, K^+\}$, where K^+ is a Poisson random variable:

$$K^+ \sim \text{Poisson} \quad \frac{\lambda_0}{\lambda_0 + \sum_{k=1}^{K} \lambda_k(t_{n-1})}\Bigg)\Bigg(\tag{5.5}$$

If κ are binary, which is the case in the standard IBP setting, and $\lambda_0 = 1$, then the mean of K^+ becomes $1/n$ and $p_k = (n-1)/n$, which reduces to the case of IBP with parameter 1:

$$\sum_{k=1}^{K}\sum_{i=1}^{n-1}\Bigg(\frac{\kappa_{ik}(t_{n-1} - t_i)}{\|\boldsymbol{\kappa}_i\|_0} = \sum_{i=1}^{n-1}\frac{\|\boldsymbol{\kappa}_i\|_0}{\|\boldsymbol{\kappa}_i\|_0} = n - 1 \tag{5.6}$$

For each new factor k', we draw from the corresponding priors for $\mathbf{w}_{k'} \sim \text{Dir}(\mathbf{w}|\mathbf{w}_0)$ and $\mathbf{v}_{k'} \sim \text{Dir}(\mathbf{v}|\mathbf{v}_0)$.

Next, we decide the hidden state variables $\mathbf{z}_i = \{\mathbf{K}_i, \mathbf{V}_i\}$. $\mathbf{V}_i$ is constructed by simply adding columns for the $\mathbf{v}_{k'}$ for newly sampled factors to $\mathbf{V}_{n-1}$. $\mathbf{K}_i$ is constructed by first updating $\mathbf{K}_{n-1}$ with respect to the new lag time $\delta = t_i - t_i$. This step is done *symbolically*, since we do not know t_i yet. Then we add the rows $\kappa_{ik'}$ for the newly sampled event based on Equation 5.2 with $\delta = 0$. We emphasize that $K_n(t_i) : \mathbf{R}^+ \to \mathbf{R}^{n \times (K+K')}$ at this moment is a symbolic function of t_n.

Conditioned on these state variables $\mathbf{z}_i$, we sample the n^{th} observation $\mathbf{y}_i = \{t_i, \mathcal{T}_i\}$: The *time stamp* t_i, depending on its related factors, is sampled from a Poisson process with rate

$$\lambda(t_i) = \sum_{\kappa_{nk} \neq 0}\Bigg(\lambda_k(t_i) = \sum_{\kappa_{nk} \neq 0}\sum_{i=1}^{n}\frac{\kappa_{ik}(t_i - t_i)}{\|\boldsymbol{\kappa}_i\|_0} \tag{5.7}$$

The overall rate of IBHP, however, includes the base rate and other factors too:

$$\bar{\lambda}(t_i) = \lambda_0 + \lambda(t_i) + \sum_{\kappa_{nk}=0} \lambda_k(t_i) \tag{5.8}$$

Now, at this point, since t_i is known, we can proceed to compute the actual values of $\mathbf{K}_i$. Finally, we sample the *dictionary text* $\mathcal{T}_i$ from Multi$(D, \sum_{\kappa_{nk}\neq 0} \mathbf{v}_k / \|\boldsymbol{\kappa}_i\|_0)$, where the weight parameter is the averaged of all $\|\boldsymbol{\kappa}_i\|_0$ factor weights associated with the n^{th} observation.

---

1. Initialization:

- Model specifications: $\mathcal{M} = \{L, D, \mathcal{S}\}$;
- Model hyper parameters: $\mathbf{\Pi} = \{\mathbf{w}_0, \mathbf{v}_0\}$;
- Model parameters: $\mathbf{\Theta} = \{\lambda_0, \{\beta_l, \tau_l\}\}$;

2. Generate the First Event:

- Set $c_{1,1:K} = 1$, where $K \sim Poisson(\alpha_0)$;
- Sample $\mathbf{w}_k \sim \text{Dir}(\mathbf{w}|\mathbf{w}_0)$ and set $\boldsymbol{\kappa}_1$;
- Sample $\mathbf{v}_k \sim \text{Dir}(\mathbf{v}|\mathbf{v}_0)$;
- Sample $t_1 \sim \mathcal{PP}(\lambda_0)$;
- Sample $\mathcal{T}_1 \sim \text{Multi}\left(D, \sum_{\kappa_{1k}\neq 0} \mathbf{v}_k / \|\boldsymbol{\kappa}_1\|_0\right)$

3. Generate Follow-up Events:

for $n = 2, \ldots, N$ **do**

- Sample $\mathbf{c}_i$ according to Equations 5.4 and 5.5.
- Sample $\mathbf{w}_{k'} \sim \text{Dir}(\mathbf{w}|\mathbf{w}_0)$ and set $\boldsymbol{\kappa}_i$;
- Sample $\mathbf{v}_{k'} \sim \text{Dir}(\mathbf{v}|\mathbf{v}_0)$;
- Sample $t_i \sim \mathcal{PP}(\lambda(t_i))$ by Equation 5.7.
- Sample $\mathcal{T}_i \sim \text{Multi}\left(D, \sum_{\kappa_{nk}\neq 0} \mathbf{v}_k / \|\boldsymbol{\kappa}_i\|_0\right)$.

end

Algorithm 1: Generative process of IBHP.

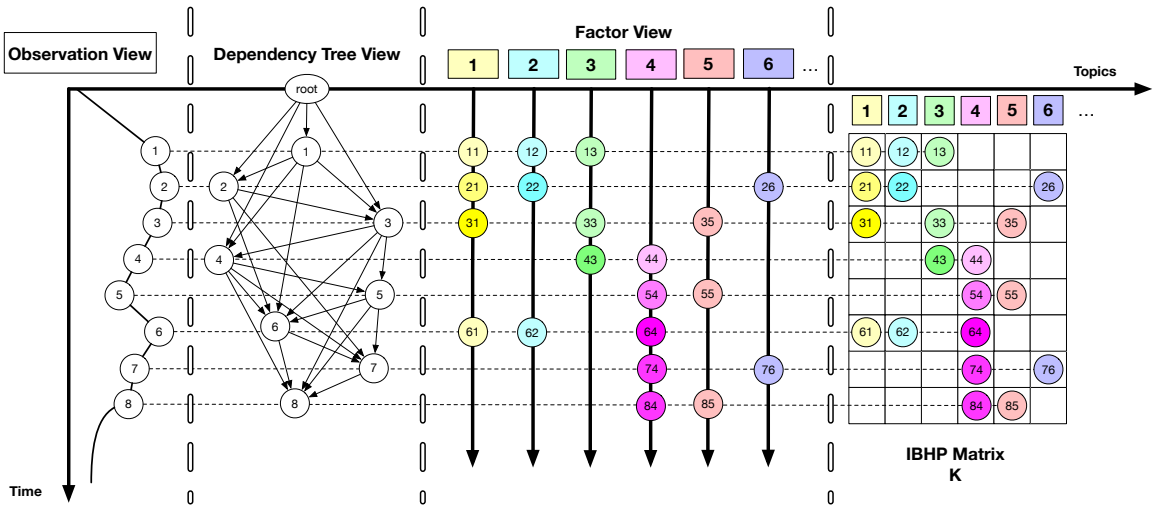


Fig. 5.3. An example of IBHP. In this IBHP realization, the first 8 observations created 6 factors. Each factor has a distinctive color, and color intensities represent instantaneous factor popularities. An observation may be labeled with multiple factors, and are colored in its decomposed factor view accordingly. The dependency tree describes the related events for each observation, where the directed arrows indicate dependency relations. The rate for any *observation* is the aggregation of all its *related factor* rates (see Equation 5.7), whereas the overall rate at any *time* is the sum of *all factor* rates – so the overall rate can be excited by one observation multiple times through different factors. The overall rate is represented by its height relative to the reference time line. See Section 5.5.1 for more details.

5.4 Algorithm

Sequential Monte Carlo [100] (SMC) methods are powerful and flexible tools for sequential models, where the observation process $\mathbf{y}_n$ is driven by the latent state process $\mathbf{z}_n$, which is represented by a set of F particles at any time (i.e., from 1 to N). Here, we adapt particle filtering methods to our set up, allowing us to scale our model to large-data regimes. We build on ideas from [101], extending them to our more structured setting.

The idea at a high level is to propagate each particle forward by one time step according to the prior, and then reweight each particle by how 'compatible' it is with the observation at that time. If a few of the particles have weights that dominate the rest (resulting in a small effictive number of particles), then the algorithm resamples F particles with replacement proportional to the particle weights. Our algorithm for IBHP can be described as follows (see Algorithm 2 for pseudocode):

A. Initialize Particle Weights. The particle weights are initialized uniformly: $u_1^f = \frac{1}{F}$, for $f = 1, \ldots, F$.

Then for each time step $i = [1..N]$, we do the following:

B. Sample Particles. According to [101], our particles $\tilde{\mathbf{z}}_i^f = \{\tilde{\mathbf{K}}_i^f, \tilde{\mathbf{V}}_i^f\}$ are sampled based on the conditional distributions $p(\mathbf{z}_i|\mathbf{z}_{i-1})$ described in Section 5.3.3.

C. Sample Model Parameters. Since the posterior of the model parameter $\mathbf{\Theta} = \{\lambda_0, \{\beta_l\}, \{\tau_l\}\}$ is proportional to the product of its priors and the data likelihood described in Equation 2.19 and Section 5.3, we can first sample from its priors, and then use the product of the priors and the HP data likelihood as weights of the samples to approximate the posterior [37]. We update the triggering kernels using the new parameters.

D. Update Particle Weights. The importance weight is the ratio between the true posterior and the proposal distribution. If we use the prior as the proposal, we update the particle weights by $u_i^f = u_{i-1}^f p(\mathbf{y}_i|\tilde{\mathbf{z}}_i^f, \mathbf{\Theta})$ and then normalize them to $u_j^f = u_j^f / (\sum_{f=1}^F u_j^f)$.

E. Resample Particles. If the effective number of particles is too small, we resample with replacement F particles from the existing ones with the normalized weights.

The algorithm described here for the IBHP is easy to implement, flexible, and scalable. Due to the sequential updating strategy, from the pseudocode in Algorithm 2, we see that for large N, the time complexity of this algorithm is $\mathcal{O}(NF)$, where N is the number of observations and F is the number of particles. We will demonstrate and discuss the effectiveness of the algorithm in the experiment section in more detail.

Initialize the F uniform particle weights.

for *each observation* $\mathbf{y}_i = \{t_i, \mathcal{T}_i\}$, $i = 1, \ldots, n$ **do**

  for *each particle* $\mathbf{z}_i^f = \{\mathbf{K}_i, \mathbf{V}_i\}$ *of observation* $\mathbf{y}_i$, $f \in \{1, \ldots, F\}$ **do**

    - Sample the auxiliary variables $\mathbf{w}_i, \mathbf{c}_i$ and latent factor particles $\mathbf{z}_i^f = \{\mathbf{K}_i, \mathbf{V}_i\}$.
    - Sample the model parameters $\mathbf{\Theta} = \{\lambda_0, \{\beta_l\}, \{\tau_l\}\}$.
    - Update the triggering kernels.
    - Update the particle weights $\mathbf{u}_i^f$.

  end

  Normalize the particle weights.

  if $\|\mathbf{u}_i\|_2^{-2} <$ *threshold, i.e., the effective number of particles is too low* **then**
    Resample particles with replacement based on the particle weights.

  end

end

Algorithm 2: SMC inference algorithm for the IBHP.

5.5 Experiments

We compare our model with three methods from the previous section: the vanilla Hawkes process (HP), the Dirichlet Hawkes (DHP; [37]), and the Hierarchical Dirich-

let Hawkes (HDHP; [63]). We evaluate the models on both synthetic and real-world data.

5.5.1 Synthetic Dataset Experiments

The purpose of our synthetic-data experiments is twofold: 1) to understand the identifiability of our model and the accuracy of our SMC algorithm when the true data generation process satisfies the model assumptions, and 2) to understand the effects of misspecification.

Our setup is as follows. The Hawkes process base rate is $\lambda_0 = 2$. For the basis kernels, we use: $\gamma_1(\delta) = e^{-\delta/0.3}, \gamma_2(\delta) = 2e^{-\delta/0.2}, \gamma_3(\delta) = 3e^{-\delta/0.1}$. γ_1 has the smallest jump but also the largest time-scale; at the other extreme, γ_3 has the largest jump with a fast decay-parameter. γ_2 might be used to model 'regular' events, while γ_1 and γ_3 are for non-urgent and urgent ones respectively. We construct the dictionary $\mathcal{S}$ from the top 1000 words from the NIPS dataset [99], and the document lengths are set to $D = 20$. The hyperparameters, which are not to be estimated, are set as $\mathbf{w}_0 = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}), \mathbf{v}_0 = (\frac{1}{1000}, \ldots, \frac{1}{1000})$. We generate $N = 1000$ observations with this setup, and use the first 80% of the dataset for training, and the last 20% for testing. For each SMC iteration, we use 10 particles, and report averages and errorbars based on 10 runs with different random seeds.

A. Parameter learning and prediction. Experiments A1 and A2 shown in Table 5.1 are the parameter estimates and the log-likelihoods over training and test datasets. Our model outperforms other models both in terms of predictive log-likelihood. This demonstrates two points. Firstly, our SMC algorithm is able to accurately recover the underlying model parameters. Furthermore, estimating parameters for the misspecified models on this dataset is fair, since they have the same interpretation. Thus for instance, our results tell us that fitting a Hawkes model that does not include multiple triggering factors results in a significant overestimation of the base rate λ_0: a result that one might have expected.

Table 5.1.
Model comparison over the synthetic datasets.

A1. Parameter Estimation			
Parameter	λ_0	$\{\beta_l\}$	$\{\tau_l\}$
Values	2	1,2,3	0.3,0.2,0.1
Our Model	**1.8**	**0.92, 1.63, 2.71**	**0.33, 0.18, 0.09**
HDHP	3.3	0.77, 4.56 6.11	3.75, 3.20, 2.94
DHP	2.9	0.83, 5.72, 5.83	1.21, 1.58, 1.28
HP	5.4	2.25, 4.38, 3.01	0.73, 2.54, 3.56
A2. Log-likelihoods			
	Training		Test
Our Model	**318.52**		**47.68**
HDHP	192.74		12.23
DHP	201.96		11.78
HP	81.68		6.18
B. Learn Latent State Variables ($K = 5, 10, 20$)			
	Jaccard($\mathbf{K}$)		1 - Hellinger($\mathbf{V}$)
Our Model	**0.83, 0.81, 0.77**		**0.79, 0.73, 0.68**
HDHP	0.56, 0.40, 0.35		0.51, 0.44, 0.29
DHP	0.61, 0.42, 0.38		0.64, 0.41, 0.36

B. Learn latent state variables. Table 5.1 part B focuses on learning the latent state variables. Now, rather that generating data from our nonparametric model, we fix $K = 5, 10, 20$ in the data-generating process, and then compare these with our nonparametric esimates using two metrics: the Jaccard Index to compare the *binary* matrices $\mathbf{C}$ and the Hellinger distance for $\mathbf{V}$. A first complication is that these matrices need not have the same number of columns, and so for each comparison, we pad the smaller matrix with zero-columns to facilitate comparison. A bigger challenge

is a 'label-switching' issue that arises since column permutations do not effect the quality of the estimates. To overcome this, after matching dimensions, we greedily match columns, and then compute scores. We point out that padding with zeros favors the alternative methods, since their solutions have many zeros; nevertheless, our model still gives the best Jaccard scores as well as Hellinger distances (we actually report *complementary* Hellinger distances (viz. one minus the actual distance), so that large numbers imply better performance for both statistics. As before, our results demonstrate the insufficiency of the alternate models and justifies the need for multiple factors.

C. The effects of base rate and basis kernels. The base rate λ_0, together with the evolving kernels, control the dynamics of latent factors. In Table 5.2 part C, we vary the value of λ_0, and see that increasing this increases the average number of factors per observation increases—more strongly violating the *single* factor assumption of competing methods. We also see that this is accompanied by a widening of the performance gap between our model and the alternatives.

Table 5.2.
Effects of model specifications.

C. Effects of Base Rate					
	λ_0	Topics	Jaccard	Hellinger	Test
Our Model	4	9.01	0.79	0.75	50.21
	8	12.28	0.72	0.69	68.37
	16	28.33	0.64	0.61	72.07
HDHP	4	1	0.32	0.40	43.78
	8		0.28	0.38	51.06
	16		0.31	0.26	50.79
DHP	4	1	0.29	0.37	41.67
	8		0.33	0.31	49.18
	16		0.27	0.28	52.33

D. The effects of triggering rule. In Equation 5.7, the event rate depends on the rates of the underlying factors in an additive manner. We can allow more flexible triggering rules by allowing richer interactions among factor dynamics. For example, we define a "double-sharing" triggering rule as follows: *trigger a jump in the rate function only when two or more factors are shared with a previous observation.* Thus Equation 5.7 becomes:

$$\lambda(t_n) = \sum_{\kappa_{nk}\neq 0} \left(\left[\sum_{i=1}^{n-1} \frac{\kappa_{ik}(t_n, t_i)}{\|\boldsymbol{\kappa}_i\|_0} + \phi \left(\frac{\kappa_{ik}(t_n, t_n)}{\|\boldsymbol{\kappa}_n\|_0} \right) \right] \right(\tag{5.9}$$

where $\phi = 0$ if the rule is not triggered—there is no "jump", otherwise $\phi = \mathbf{w}_k^\intercal \boldsymbol{\beta} / \|\boldsymbol{\kappa}_n\|_0$—there is a "jump". We sketch this out in Figure 5.4.

Incorporating such nonlinearities result in dynamics that are significantly different from the additive setup: this is evidenced in Table 5.3, where the simpler additive version the our model now has a degraded score. There are numerous variations to our simple "double double-sharing" rule that are relevant across a variety of situations.

Table 5.3.
Model comparison with "double-sharing" dataset.

D. Predictive Log-likelihoods on Double-sharing Data		
	Additive Model	Double-sharing Model
Our Model	15.38 ±3.82	20.82 ±3.23
HDHP	8.97 ±4.07	12.36 ±3.18
DHP	8.26 ±3.19	10.17 ±3.20
HP	4.98 ±3.61	5.04 ±3.22

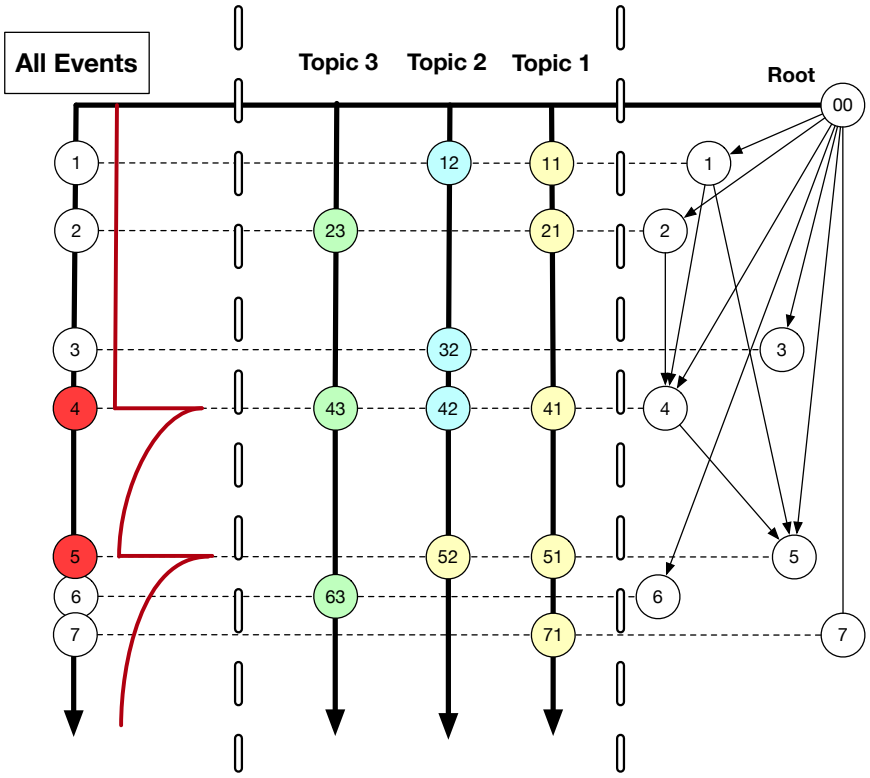


Fig. 5.4. Our model with the "double-sharing" rule. Obs. 2 does not trigger a "jump" because no previous observations share more than two factors with it. However, obs. 4 triggers *two* jumps because it shares two factors with obs. 1 (factor 1 and 2), and two with obs. 2 (factor 1 and 3).

5.5.2 Real Dataset Experiments

The purpose of our real data experiments is threefold: 1) to verify that multiple triggers are indeed relevant to real applications, 2) to demonstrate that our inference algorithm is scalable for real-world datasets, and 3) to use our model to present meaningful findings, both quantitative and qualitative. We consider four different datasets: **Facebook Dataset.** This data contains Facebook message communications among 20,603 individuals. We pick the top 10 most connected individuals (based on the number of friends), and add in their one-hop and two-hop friends. This results in a total of 376 individuals. **NIPS Dataset [99].** The Kaggle NIPS dataset contains the title, authors, abstracts, and extracted text for all 7241 NIPS papers from the

first 1987 conference to the current 2017 conference. This dataset is different in that it contains rich message information; however the number of time-points is just 30. **Santa Barbara Corpus Dataset [96].** This is a standard dataset used for applications involving Hawkes processes. We use conversation #33, a lively family discussion which centers around a disagreement that an individual, Jennifer, is having with her mother, Lisbeth. **Enron Email Dataset [95].** The Enron dataset contains about half a million email messages communicated among about 150 senior managers of the Enron corporation. We pick the longest thread of emails.

For each experiment, we use the first 80% of the dataset as training set, the next 10% as validation set, and the last 10% as test set. We train our model on training sets with different hyperparameters, then pick the best one based on their performances on the validation set, and use this model to report performances on the test set. The reported values are based on ten runs with different random number seeds. The dictionary $\mathcal{S}$ is all the unique words in the dataset; the document length D_n is counted from each observed text $\mathcal{T}_n$; and we use the three ($L = 3$) exponential basis kernels defined in Equation 5.1.

Assessing model fit

A. Predictive log-likelihood. The log-likelihoods in Table 5.4 show that for three of four datasets, our model outperforms the alternatives. The performance gaps exhibit a range of values. On the NIPS dataset, our model shows a massive improvement over the competition, while there is no significant improvement for the Enron dataset. The numbers in parentheses, giving the average number of topics associated with each message, provides a partial explanation. For the Enron dataset, this number is just two, suggesting that there is limited benefit from modeling multiple factors, and that the simpler HDHP model may be more appropriate. For the NIPS dataset, this number is about 10, explaining the gap in performance.

Table 5.4.
Model comparisons over the real datasets.

FB Dataset (average # factors = 4.19)			
	Training	Validation	Test
Our Model	**1822** ±96	**219** ±10	**277** ±11
HDHP	1083 ±88	123 ±10	133 ±10
DHP	1058 ±90	144 ±9	200 ±14
HP	782 ±75	62 ±7	69 ±7
NIPS Dataset (average # factors = 10.21)			
	Training	Validation	Test
Our Model	**8378** ±172	**913** ±23	**1012** ±28
HDHP	3229 ±169	216 ±12	191 ±11
DHP	2018 ±164	203 ±10	202 ±10
HP	390 ±48	49 ±8	40 ±7
SB Dataset (average # factors = 6.52)			
Our Model	**520** ±62	**187** ±12	**137** ±9
HDHP	132 ±9	32 ±6	34 ±6
DHP	169 ±10	51 ±7	78 ±9
HP	96 ±10	15 ±4	23 ±4
Enron Dataset (average # factors = 2.17)			
Our Model	2602 ±101	**313** ±12	381 ±12
HDHP	2322 ±117	203 ±10	**392** ±11
DHP	**2639** ±118	268 ±11	339 ±12
HP	729 ±92	28 ±5	19 ±5

Table 5.5.
Model comparison on the shuffled NIPS dataset.

Test Log-likelihoods on the NIPS Dataset			
	Original	Shuffled	Relative Loss
Our Model	**1012.08**	**914.76**	**-9.62%**
HDHP	191.29	88.19	-53.90%
DHP	201.73	79.05	-60.81%
HP	40.17	18.22	-54.64%

B. Latent structure vs. dynamics. The rich structure of the NIPS dataset is balanced by its simple temporal structure just with 30 time points. This raises the question: how much of our models performance is due to the latent structure incorporated into our modeling framework, and how much is due to temporal dynamics of this structure. To study this more carefully, we shuffle the publication years (documents published in the same year remain together, however), thus destroying temporal information. Table 5.5 shows that this incurs a relatively small loss now, suggesting that most of the performance gains observed in Table 5.4 are due to the latent factors. However, removing temporal information still incurs enough of a hit in performance to justify our methodology.

C. Discovering popular topics and words. One of the immediate benefits of our IBHP is that it returns the factor rate matrix $\mathbf{K}$ and the word-distribution matrix $\mathbf{V}$, providing a rich summary of popular topics and words. Figure 5.5 shows, in the NIPS dataset, the most popular three topics at the end of the training dataset time span. The lists of words suggest that the first topic is related to kernel methods, the second to deep learning, and the third to Bayesian methods. The intensity of the colors indicates popularities. Our model suggests that topic 2, which hypothetically is related to deep learning, has been increasingly more popular in the NIPS community.

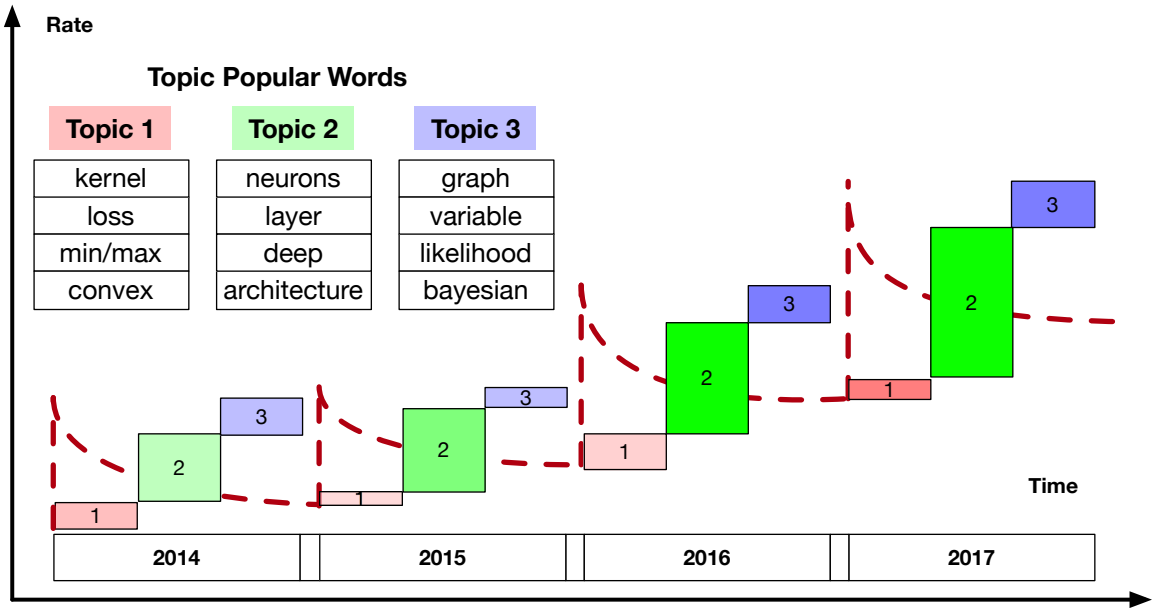


Fig. 5.5. NIPS dataset. Popular topics and words.

D. Learning factor dynamics. Unlike the IBP, the IBHP matrix not only carries binary "present/missing" information, but also real-valued kernel weights κ_{ik}, which reveal the temporal dynamics of the factors. Figure 5.6 shows two most popular factors from the FB dataset. The first relates to school life, and the second to off-class activities. To confirm this, we plot the average of the estimated rate functions across four similar one week periods in Figure 5.6. The patterns of the two factor rates are quite different: The first factor is active after Monday, and peaks in the middle of the week, before cooling down near the weekend. By contrast, the second factor climbs steadily, though at a much lower rate, and becomes more excited than the first factor during the weekend.

E. Infering dependencies and causalities. According to Equation 5.7, the rate after an event depends on earlier events that share factors with it. Figure 5.7 provides a detailed view of the IBHP on the SB dataset under the usual additive rule. We also apply the "double-sharing" rule to the dataset and plot the results in the same format in Figure 5.8. We see several consequences: 1) the rate functions are not triggered until the 6^{th} observation under the double-sharing rule, 2) the IBHP matrices are different, and 3) the inferred factors are different. Further investigation shows the first red circle corresponds to the observation with text *"I am mean to you all the time!"* and the last red circle to *"What time is it?"*—one to heat up the process and one to cool it down. This suggests that adopting different triggering rules may allow us to capture different aspects of the dataset, which in our SB double-sharing case, bookends an active family discussion.

F. Predict future event times. In Table 5.4, we report the log-likelihoods on the test datasets for each a model. To evaluate the predictive ability of our model in more depth, we use it for a different predictive task: predict the time of the next event in windows of increasing sizes, and for each case, report the absolute different from the observed data. Table 5.6 shows that, as the size of predictions increases,

(a) Word cloud of factor 1.

(b) Word cloud of factor 2.

(c) Rate functions of factors 1/2.

Fig. 5.6. Topic dynamics on FB dataset.

Table 5.6.
Predicting future event times on FB dataset.

	Prediction Window Size		
	$pws = 1$	$pws = 5$	$pws = 10$
Our Model	0.61 ±0.11	0.97 ±0.18	1.37 ±0.28
HDHP	0.82 ±0.13	1.24 ±0.20	2.18 ±0.33
DHP	0.87 ±0.10	1.19 ±0.16	2.21 ±0.29
HP	0.92 ±0.17	2.06 ±0.23	3.56 ±0.31

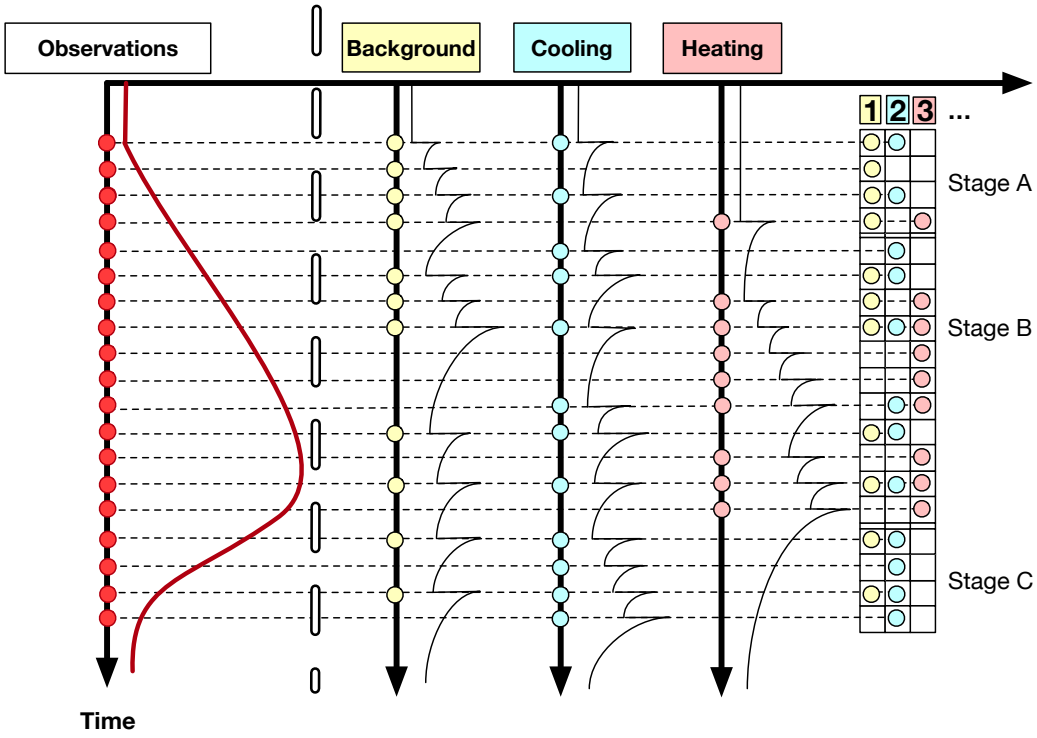


Fig. 5.7. Additive rule on SB dataset. Every observation creates a jump of the rate function. Topics can be interpreted as background, cooling, and heating activities.

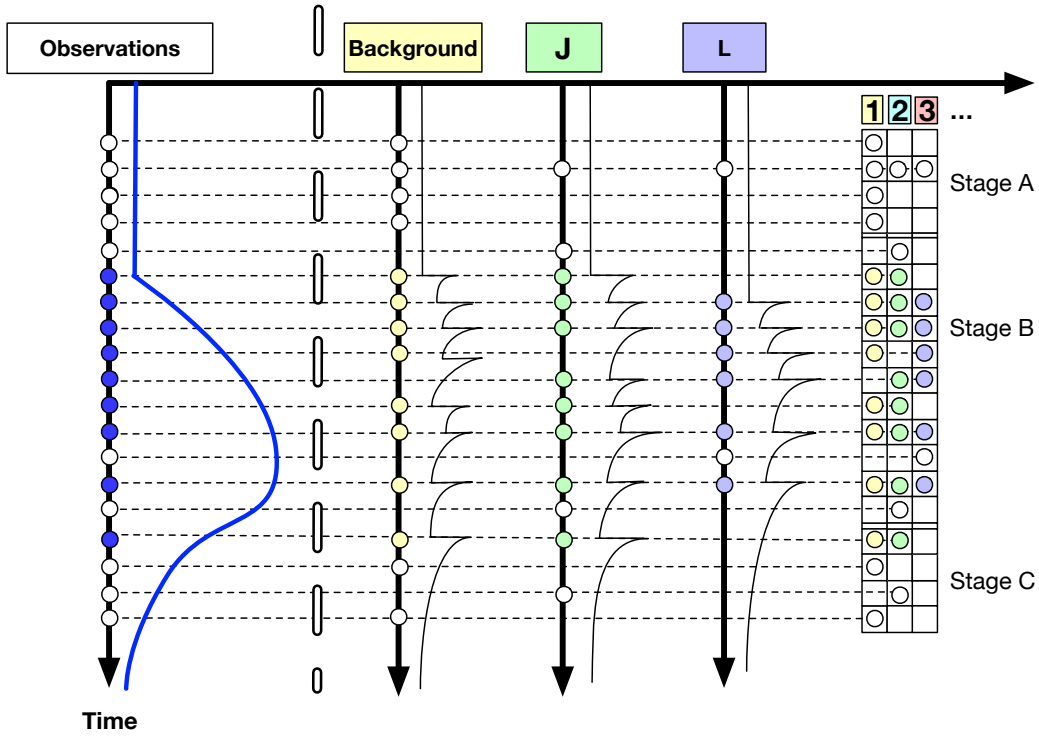


Fig. 5.8. SB Dataset with double-sharing. White circles represent observations that do not trigger the rule. Topics can be interpreted as background activities, and those of Jennifer and Lisbeth.

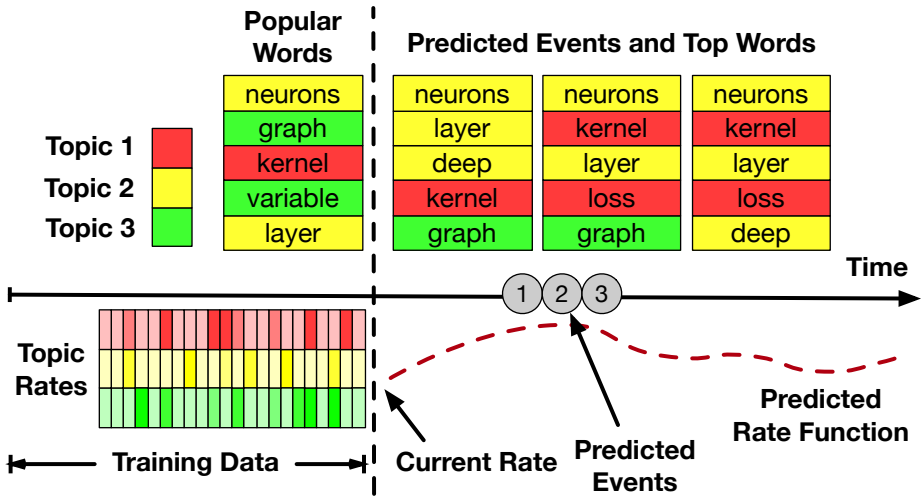


Fig. 5.9. Predicted events on NIPS dataset.

the mean absolute error increases, as well as the standard error: as the predictions becomes harder, the predictions becomes inaccurate and unreliable. Nonetheless, our model outperforms competing models in according to this metric as well.

G. Predicting future topics and words. Our last experiment is concerned with the prediction of the latent state variables. The dotted line in Figure 5.9 represents the end of the training phase, where we have obtained the latent factor rate matrix $\mathbf{K}$ and the latent factor word distribution matrix $\mathbf{V}$. To the right of the dotted line, we show the projected rate function, along with the first three predictions and their predicted top words. Our model suggests that, for the NIPS dataset, topic 2 is taking over topic 3 and may become dominant in the next few events.

5.6 Related Work

The idea of considering nonparametric Bayesian models with temporal point processes in a unified framework has been popular in recent years. For example, [13] proposed a Bayesian nonparametric model that utilizes the Chinese Restaurant Processes (CRP) as a prior for the clusters among individuals, whose rates of communications

are modeled by HP. [10] used a similar idea but further extended the model by modeling the jump sizes of HP using Gaussian Processes (GP). HP models with various generalizations of a CRP, such as the distance dependent CRP (ddCRP) [56], the nested CRP (nCRP) [11], and the Chinese Restaurant Franchise Processes (CRFP) [45], have also been explored.

Other attempts have been made by borrowing the ideas from Deep Learning. For example, [55] proposed a model to view the intensity function of a temporal point process as a nonlinear function of the history, and use recurrent neural networks to automatically learn a representation of the influences from the event history. [60] modeled streams of events by constructing a neurally self-modulating multivariate point process where the intensities of multiple event types evolve based on a continuous-time LSTM. Lastly, [72] considered the use of latent factors in HP models to represent dependencies among instances that influence reciprocity over time. But the work focused on modeling static factors of homophily and reciprocity in social networks and not the evolution of factors over time.

Perhaps the closest works to our model are [37] and [63]. In [37], the authors proposed a Dirichlet Hawkes Processes (DHP) model that combines the CRP with HP in a unified framework, where the cluster assignment in CRP is driven by the intensities of HP. [63] further developed this in their Hierarchical Dirichlet Hawkes Processes (HDHP) model by replacing the CRP with a CRFP that is capable of modeling steaming data for multiple users. However, there are several major distinctions compared to our IBHP: 1) In both the DHP and HDHP models, events are triggered by single factors, while in our IBHP, multiple latent triggering factors are introduced; 2) the form of the triggering kernels do not depend on history events, and in contrast, our IBHP model is more flexible to be able to adopt non-additive triggering rules to learn different perspectives of the observed data. We will compare our model to [37] and [63] next.

5.7 Summary

In this chapter, we proposed the Indian Buffet Hawkes Process (IBHP)—a novel Bayesian nonparametric stochastic point process model for learning multiple latent triggering factors of streaming document/message data. Our approach establishes the synergy between Indian Buffet Processes (IBP) and Hawkes processes (HP): on the one hand, we use the IBP to add multiple triggering factors to the HP, which helps to better model dynamics and improves interpretation, and on the other hand, the temporal information from the HP is embedded into the IBP to drive the latent factor estimation, which expands its capability to model evolution of factors.

6. CONCLUSIONS AND FUTURE WORK

In this dissertation, we propose three novel Bayesian nonparametric HP models, and present their effectiveness through various empirical evidence.

In the first component, we outline the GHP, which extends the work of [13] by introducing GP into the Hawkes IRM model. We use these to account for the content of the messages, capturing the message significance as well as receptivity. This allows us to more accurately capture the interactions among entities. The interaction between a pair of clusters is modeled as the additive effect of the interactions between all pairs of nodes in the two clusters, allowing us to identify the contribution of each pair of nodes, where the actual communication is taking place, to the interaction between a pair of clusters. The introduction of GPs also allows us to flexibly model the rates of reciprocal activities between two entities, hence the asymmetry in reciprocity can be captured more accurately. We show how this leads to a better cluster detection capability. Since our proposed work is a natural extension of Hawkes IRM, it covers both Poisson processes and IRM as special cases.

In the second component, we outline the nCRP-GHP, which introduces senders (S_i) and receivers (R_i) into a novel and unified framework combining the advantages of hierarchical nonparametric Bayesian models and temporal point processes. This enables us to leverage temporal (t_i) and textual ($\mathcal{T}_i$) information present in the communications, allowing improved predictions about event times and clusters. Our method exploits senders' and receivers' properties to characterize message content, enabling inference about authorship and audience of communications, as well as their personal behavior such as favorite collaborators and top-pick words. Empirical results with our nonparametric Bayesian point process model show that our formulation has improved predictions about event times and clusters. In addition, the latent struc-

ture revealed by our model provides a useful qualitative understanding of the data, facilitating interesting exploratory analyses.

In the third component, we outline the IBHP, which is a novel Bayesian nonparametric stochastic point process model for learning multiple latent triggering factors of streaming document/message data. Our approach establishes the synergy between IBP and HP: on the one hand, we use the IBP to add multiple triggering factors to the HP, which helps to better model dynamics and improves interpretation, and on the other hand, the temporal information from the HP is embedded into the IBP to drive the latent factor estimation, which expands its capability to model evolution of factors. In addition, we developed an efficient and scalable learning algorithm based on Sequential Monte Carlo (SMC) and demonstrated the effectiveness of our model and algorithm across various experiments on both synthetic and real datasets.

HPs are powerful tools to model temporal data, and together with Bayesian nonparametric models, the synergy brings us to a new level of detailed modeling. However, the complexity of the framework also implies new challenges. We give two examples of future work here. First, efficient inference algorithms are still open problems for HP. In this dissertation, we have demonstrated the power and flexibility of sampling methods. Although sampling methods are widely applicable and easy to implement, they suffer from issues of high computational complexity and limited scalability. Therefore, designing new efficient inference algorithms for HPs is a promising direction of future work. Second, a unified framework exploring the possibilities of combining the three components in this dissertation is also interesting. Since each of the three modeling components presented in this dissertation is already complex in nature, it is with top priority to frame them into a structured modeling scheme. As deep learning is becoming increasingly popular in recent years and many applications – of particular interest in data with complex dynamics – are proved to be very effective with deep learning techniques, it would be therefore another interesting direction to explore the joint modeling of HPs and deep learning ideas in the future.

REFERENCES

REFERENCES

[1] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," in *Social networks*, 1983.

[2] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2003.

[3] Z. Xu, V. Tresp, K. Yu, and H. P. Kriegel, "Infinite hidden relational models," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2006.

[4] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.

[5] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, and S. Suri, "Feedback effects between similarity and social influence in online communities," in *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2008.

[6] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," in *arXiv preprint arXiv:1105.0697*, 2011.

[7] L. Mitchell and M. E. Cates, "Hawkes process as a model of social interactions: a view on video dynamics," in *Journal of Physics A: Mathematical and Theoretical*, 2009.

[8] A. G. Hawkes, "Point spectra of some mutually exciting point processes," in *Journal of the Royal Statistical Society. Series B (Methodological)*, 1971.

[9] ——, "Spectra of some self-exciting and mutually exciting point processes," in *Biometrika*, 1971.

[10] X. Tan, S. A. Naqvi, A. Y. Qi, K. A. Heller, and V. Rao, "Content-based modeling of reciprocal relationships using Hawkes and Gaussian processes." in *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2016.

[11] X. Tan, V. Rao, and J. Neville, "Nested CRP with Hawkes-Gaussian processes," in *Artificial Intelligence and Statistics (AISTATS)*, 2018.

[12] ——, "The Indian buffet Hawkes process to model evolving latent influences," in *submission to the Conference on Uncertainty in artificial intelligence (UAI)*, 2018.

[13] C. Blundell, J. Beck, and K. A. Heller, "Modelling reciprocating relationships with Hawkes processes," in *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[14] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda, "Learning systems of concepts with an infinite relational model," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2006.

[15] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing, "Mixed membership stochastic blockmodels," in *Journal of Machine Learning Research (JMLR)*, 2008.

[16] A. G. Hawkes and D. Oakes, "A cluster process representation of a self-exciting process," in *Journal of Applied Probability*, 1974.

[17] T. J. Liniger, "Multivariate Hawkes processes," Ph.D. dissertation, ETH Zurich, 2009.

[18] L. Zhu, "Nonlinear Hawkes processes," Ph.D. dissertation, New York University, 2013.

[19] A. Simma and M. I. Jordan, "Modeling events with cascades of Poisson processes," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.

[20] K. Zhou, H. Zha, and L. Song, "Learning triggering kernels for multi-dimensional Hawkes processes," in *International Conference on Machine Learning (ICML)*, 2013.

[21] S. H. Yang and H. Zha, "Mixture of mutually exciting processes for viral diffusion," in *International Conference on Machine Learning (ICML)*, 2013.

[22] T. Iwata, A. Shah, and Z. Ghahramani, "Discovering latent influence in online social activities via shared cascade Poisson processes," in *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2013.

[23] L. Li and H. Zha, "Dyadic event attribution in social networks with mixtures of Hawkes processes," in *International Conference on Information & Knowledge Management (CIKM)*, 2013.

[24] J. F. Olson and K. M. Carley, "Exact and approximate em estimation of mutually exciting Hawkes processes," in *Statistical Inference for Stochastic Processes*, 2013.

[25] S. Linderman and R. Adams, "Discovering latent network structure in point process data," in *International Conference on Machine Learning (ICML)*, 2014.

[26] M. Farajtabar, N. Du, M. G. Rodriguez, I. Valera, H. Zha, and L. Song, "Shaping social activity by incentivizing users," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[27] T. Gunter, C. Lloyd, M. A. Osborne, and S. J. Roberts, "Efficient Bayesian nonparametric modelling of structured point processes," in *arXiv preprint arXiv:1407.6949*, 2014.

[28] L. Li, H. Deng, A. Dong, Y. Chang, and H. Zha, "Identifying and labeling search tasks via query-based Hawkes processes," in *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2014.

[29] L. Li and H. Zha, "Learning parametric models for social infectivity in multi-dimensional Hawkes processes." in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2014.

[30] R. Lemonnier and N. 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*, 2014.

[31] M. Farajtabar, Y. Wang, M. G. Rodriguez, S. Li, H. Zha, and L. Song, "Coevolve: A joint point process model for information diffusion and network co-evolution," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[32] N. Du, Y. Wang, N. He, J. Sun, and L. Song, "Time-sensitive recommendation from recurrent user activities," in *Advances in Neural Information Processing Systems (NIPS)*, 2015.

[33] X. He, T. Rekatsinas, J. Foulds, L. Getoor, and Y. Liu, "Hawkestopic: A joint model for network inference and topic modeling from text-based cascades," in *International Conference on Machine Learning (ICML)*, 2015.

[34] F. Guo, C. Blundell, H. Wallach, and K. Heller, "The Bayesian echo chamber: Modeling social influence via linguistic accommodation," in *Artificial Intelligence and Statistics (AISTATS)*, 2015.

[35] D. Luo, H. Xu, Y. Zhen, X. Ning, H. Zha, X. Yang, and W. Zhang, "Multi-task multi-dimensional Hawkes processes for modeling event sequences." in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[36] Y. L. K. Samo and S. Roberts, "Scalable nonparametric Bayesian inference on point processes with Gaussian processes," in *International Conference on Machine Learning (ICML)*, 2015.

[37] N. Du, M. Farajtabar, A. Ahmed, A. J. Smola, and L. Song, "Dirichlet-Hawkes processes with applications to clustering continuous-time document streams," in *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2015.

[38] C. Lloyd, T. Gunter, M. Osborne, and S. Roberts, "Variational inference for Gaussian process modulated Poisson processes," in *International Conference on Machine Learning (ICML)*, 2015.

[39] S. W. Linderman and R. P. Adams, "Scalable Bayesian inference for excitatory point process networks," in *arXiv preprint arXiv:1507.03228*, 2015.

[40] R. Fierro, V. Leiva, and J. Moller, "The Hawkes process with different exciting functions and its asymptotic behavior," in *Journal of Applied Probability*, 2015.

[41] L. Tran, M. Farajtabar, L. Song, and H. Zha, "Netcodec: Community detection from individual activities," in *International Conference on Data Mining (ICDM)*, 2015.

[42] Y. Wang, N. Du, R. Trivedi, and L. Song, "Coevolutionary latent feature processes for continuous-time user-item interactions," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[43] A. De, I. Valera, N. Ganguly, S. Bhattacharya, and M. G. Rodriguez, "Learning and forecasting opinion dynamics in social networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[44] M. Farajtabar, X. Ye, S. Harati, L. Song, and H. Zha, "Multistage campaigning in social networks," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[45] P. Lin, T. Guo, Y. Wang, and F. Chen, "Infinite hidden semi-markov modulated interaction point process," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[46] H. Xu, M. Farajtabar, and H. Zha, "Learning granger causality for Hawkes processes," in *International Conference on Machine Learning (ICML)*, 2016.

[47] Y. Wang, B. Xie, N. Du, and L. Song, "Isotonic Hawkes processes," in *International Conference on Machine Learning (ICML)*, 2016.

[48] Y. Lee, K. W. Lim, and C. S. Ong, "Hawkes processes with stochastic excitations," in *International Conference on Machine Learning (ICML)*, 2016.

[49] A. Gunawardana and C. Meek, "Universal models of multivariate temporal point processes," in *Artificial Intelligence and Statistics (AISTATS)*, 2016.

[50] B. Cseke, D. Schnoerr, M. Opper, and G. Sanguinetti, "Expectation propagation for continuous time stochastic processes," in *Journal of Physics A: Mathematical and Theoretical*, 2016.

[51] S. Wheatley, V. Filimonov, and D. Sornette, "The Hawkes process with renewal immigration & its estimation with an EM algorithm," in *Computational Statistics & Data Analysis*, 2016.

[52] E. Bacry, S. Gaiffas, I. Mastromatteo, and J.-F. Muzy, "Mean-field inference of Hawkes point processes," in *Journal of Physics A: Mathematical and Theoretical*, 2016.

[53] K. W. Lim, Y. Lee, L. Hanlen, and H. Zhao, "Simulation and calibration of a fully Bayesian marked multidimensional Hawkes process with dissimilar decays," in *Asian Conference on Machine Learning (ACML)*, 2016.

[54] J. Etesami, N. Kiyavash, K. Zhang, and K. Singhal, "Learning network of multivariate Hawkes processes: A time series approach," in *arXiv preprint arXiv:1603.04319*, 2016.

[55] N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song, "Recurrent marked temporal point processes: Embedding event history to vector," in *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2016.

[56] P. Lin, B. Zhang, T. Guo, Y. Wang, and F. Chen, "Interaction point processes via infinite branching model." in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2016.

[57] S. A. Hosseini, A. Khodadadi, A. Arabzadeh, and H. R. Rabiee, "HNP3: A hierarchical nonparametric point process for modeling content diffusion over social media," in *International Conference on Data Mining (ICDM)*, 2016.

[58] H. Xu and H. Zha, "A Dirichlet mixture model of Hawkes processes for event sequence clustering," in *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[59] Y. Yang, J. Etesami, N. He, and N. Kiyavash, "Online learning for multivariate Hawkes processes," in *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[60] H. Mei and J. M. Eisner, "The neural Hawkes process: A neurally self-modulating multivariate point process," in *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[61] Y. Wang, X. Ye, H. Zha, and L. Song, "Predicting user activity level in point processes with mass transport equation," in *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[62] M. Achab, E. Bacry, S. Gaiffas, I. Mastromatteo, and J.-F. Muzy, "Uncovering causality from multivariate Hawkes integrated cumulants," in *International Conference on Machine Learning (ICML)*, 2017.

[63] C. Mavroforakis, I. Valera, and M. Gomez-Rodriguez, "Modeling the dynamics of learning activity on the web," in *International Conference on World Wide Web (WWW)*, 2017.

[64] M. A. Rizoiu, L. Xie, S. Sanner, M. Cebrian, H. Yu, and P. Van Hentenryck, "Expecting to be hip: Hawkes intensity processes for social media popularity," in *Proceedings of the 26th International Conference on World Wide Web (WWW)*, 2017.

[65] H. Xu, D. Luo, and H. Zha, "Learning Hawkes processes from short doubly-censored event sequences," in *arXiv preprint arXiv:1702.07013*, 2017.

[66] S. Flaxman, Y. W. Teh, D. Sejdinovic *et al.*, "Poisson intensity estimation with reproducing kernels," in *Electronic Journal of Statistics*, 2017.

[67] R. Lemonnier, K. Scaman, and A. Kalogeratos, "Multivariate Hawkes processes for large-scale inference." in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[68] H. Xu, D. Luo, X. Chen, and L. Carin, "Benefits from superposed Hawkes processes," in *arXiv preprint arXiv:1710.05115*, 2017.

[69] S. Xiao, M. Farajtabar, X. Ye, J. Yan, X. Yang, L. Song, and H. Zha, "Wasserstein learning of deep generative point process models," in *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[70] P. Wang, Y. Fu, G. Liu, W. Hu, and C. Aggarwal, "Human mobility synchronization and trip purpose detection with mixture of Hawkes processes," in *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2017.

[71] A. Zarezade, A. Khodadadi, M. Farajtabar, H. R. Rabiee, and H. Zha, "Correlated cascades: Compete or cooperate." in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[72] J. Yang, V. Rao, and J. Neville, "Decoupling homophily and reciprocity with latent space network models," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

[73] C. R. Shelton, Z. Qin, and C. Shetty, "Hawkes process inference with missing data," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[74] A. K. Menon and Y. Lee, "Proper loss functions for nonlinear Hawkes processes," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[75] H. Xu, X. Chen, and L. Carin, "Superposition-assisted stochastic optimization for Hawkes processes," in *arXiv preprint arXiv:1802.04725*, 2018.

[76] S. Rajaram, T. Graepel, and R. Herbrich, "Poisson-networks: A model for structured point processes," in *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[77] A. Gunawardana, C. Meek, and P. Xu, "A model for temporal dependencies in event streams," in *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[78] U. Nodelman, C. R. Shelton, and D. Koller, "Continuous time Bayesian networks," in *Conference on Uncertainty in artificial intelligence (UAI)*, 2002.

[79] A. Ahmed, L. Hong, and A. Smola, "Nested Chinese restaurant franchise process: Applications to user tracking and document modeling," in *International Conference on Machine Learning (ICML)*, 2013.

[80] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Sharing clusters among related groups: Hierarchical Dirichlet processes," in *Advances in Neural Information Processing Systems (NIPS)*, 2005.

[81] T. L. Griffiths, M. I. Jordan, J. B. Tenenbaum, and D. M. Blei, "Hierarchical topic models and the nested Chinese restaurant process," in *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[82] A. Karr, *Point processes and their statistical inference*, 1991.

[83] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes: volume I: Elementary Theory and Methods*, 2003.

[84] J. Moller and J. G. Rasmussen, "Perfect simulation of Hawkes processes," in *Advances in Applied Probability*, 2005.

[85] ——, "Approximate simulation of Hawkes processes," in *Methodology and Computing in Applied Probability*, 2006.

[86] Y. Ogata, "On lewis' simulation method for point processes," in *IEEE Transactions on Information Theory*, 1981.

[87] A. Dassios and H. Zhao, "Exact simulation of Hawkes process with exponentially decaying intensity," in *Electronic Communications in Probability*, 2013.

[88] T. Ozaki, "Maximum likelihood estimation of Hawkes' self-exciting point processes," in *Annals of the Institute of Statistical Mathematics*, 1979.

[89] C. K. Williams and C. E. Rasmussen, "Gaussian processes for machine learning," 2006.

[90] P. Orbanz, "Lecture notes on bayesian nonparametrics," in *Technical Report*, 2014.

[91] T. L. Griffiths and Z. Ghahramani, "The Indian buffet process: An introduction and review," in *Journal of Machine Learning Research (JMLR)*, 2011.

[92] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," in *Journal of Computational and Graphical Statistics*, 2000.

[93] I. Murray, R. Prescott Adams, and D. J. MacKay, "Elliptical slice sampling," in *Artificial Intelligence and Statistics (AISTATS)*, 2010.

[94] R. M. Neal, "Slice sampling," in *Annals of Statistics*, 2003.

[95] "Enron Email Data Set," https://www.cs.cmu.edu/~/enron/.

[96] "SB Dataset," http://www.linguistics.ucsb.edu/research/santa-barbara-corpus.

[97] C. DuBois and P. Smyth, "Modeling relational events via latent classes," in *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2010.

[98] T. Mihalcea, P. Textrank *et al.*, "Bringing order into texts," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2004.

[99] "NIPS Dataset," http://www.kaggle.com/.

[100] A. Doucet, N. De Freitas, and N. Gordon, "An introduction to sequential Monte Carlo methods," in *Sequential Monte Carlo Methods in Practice*, 2001.

[101] F. Wood and T. L. Griffiths, "Pinproceedings filtering for nonparametric Bayesian matrix factorization," in *Advances in neural information processing systems (NIPS)*, 2007.

VITA

VITA

Xi Tan was born in Zhuzhou, China. He received the B.S. degree in computer science from Northwestern Polytechnical University, China, in July 2007; the M.Phil. degree in computer science from University of Cambridge, England, in July 2008; the M.S. degree in statistics, in May 2014, the M.S. degree in mathematics, in May 2018, and the Ph.D. degree in computer science specializing in machine learning, in May 2018, all from Purdue University. After graduation, he joined Goldman Sachs in New York as a Quantitative Associate.