



CONTINUOUS TIME RELATIONAL NETWORKS INVOLVING HAWKES PROCESSES

Peiju Li

Supervisors: Prof. Scott Sisson, Dr. Xuhui Fan

School of Mathematics and Statistics
UNSW Sydney

November 2020

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF
BACHELOR OF SCIENCE WITH HONOURS

Plagiarism statement

I declare that this thesis is my own work, except where acknowledged, and has not been submitted for academic credit elsewhere.

I acknowledge that the assessor of this thesis may, for the purpose of assessing it:

- Reproduce it and provide a copy to another member of the University; and/or,
- Communicate a copy of it to a plagiarism checking service (which may then retain a copy of it on its database for the purpose of future plagiarism checking).

I certify that I have read and understood the University Rules in respect of Student Academic Misconduct, and am aware of any potential plagiarism penalties which may apply.

By signing this declaration I am agreeing to the statements and conditions above.

Signed: _____

Date: _____

Acknowledgements

By far the greatest thanks must go to my supervisors Prof. Scott Sisson and Dr. Xuhui Fan for the guidance, care and support they provided. They give me a lot encourage when I am stack, some pressure when I am slack and help me understand and overcome several hurdles along the thesis. I express my heartfelt thanks to both of them.

Also great thanks to my friends Phet, Mark, Sean, John, Song who gave me suggestion on writing, helped me in proof-reading and the presentation prepare.

Thanks to my friend Glade who provide me with a seat in her Phd room so I can be safe and productive over this Covid-19 pandemic situation.

Thanks to my family's generous support for me to finish my degree, they might not know much about what I learnt, but they are always my most solid back.

Peiju Flora Li 20 November 2020.

Abstract

Continuous time series data records in relational networks can be modelled to understand the characteristics of latent group communications and underlying mechanisms. These models can flexibly describe latent group allocations and the self-exciting, mutually-exciting features between the clusters by, combining a stochastic block model with the Hawkes process. This thesis demonstrates this by initially implementing a Hawkes model, then gradually refining that to a Poisson stochastic block model and then finally to the reciprocal Hawkes stochastic block model which were tested on generated datasets and real world datasets with the Poisson stochastic block model. This will be achieved by simulating generative models and then presenting a Bayesian inference procedure consisting of a combination of efficient Gibbs and adaptive Metropolis-Hasting sampler steps.

Contents

Chapter 1	Introduction	1
1.1	Background	1
1.2	Research aims	2
1.3	Outline of thesis	2
Chapter 2	Preliminaries and literature reviews	4
2.1	Point processes	4
2.2	Social network graph	8
2.3	Stochastic block model	9
2.4	Bayesian inference	11
2.4.1	Markov Chain Monte Carlo simulation	12
2.5	Useful distributions	13
2.5.1	Dirichlet distribution	13
2.5.2	Categorical distribution	13
2.5.3	Multinomial distribution	14
2.5.4	Chinese Restaurant Process	14
2.6	Literature summary	14
Chapter 3	Simple Hawkes Process between two individuals	15
3.1	Likelihood function of Hawkes process	15
3.2	branching structure	16
3.3	Generated Hawkes process	17
3.4	Bayesian Inferences	19
3.4.1	Joint likelihood for simple Hawkes process	19
Chapter 4	Poisson cluster model	22
4.1	Generative process	23
4.2	Inference	24
4.2.1	Joint distribution	24
4.2.2	Posteriors	24
4.3	Results	25
Chapter 5	Mutually exciting Hawkes interaction model	28
5.1	Reciprocal relation characteristic	28
5.2	Model specification	28
5.2.1	Branching structure A_{ij}	29
5.2.2	The class assignment z_i	29
5.2.3	Categorical distribution parameter π_i	29
5.2.4	Intensity function related parameters	29

5.3	Generative process	30
5.4	Inference	30
5.4.1	Joint distribution	30
5.5	Posteriors	31
5.5.1	Posterior of A	31
5.5.2	Posterior of π	31
5.5.3	Posterior of μ	31
5.5.4	Posterior of β	32
5.5.5	Posterior of ξ	32
5.5.6	Posterior of z_i	32
5.6	Results from generated data	32
5.7	Application on real world dataset	32
Chapter 6	Conclusion and extensions	35
Chapter 7	Appendix	37

CHAPTER 1

Introduction

In this chapter, we will give a general picture of the current situation and the difference between the literature. We first introduce the background of continuous relational data research and social network analysis and then discuss the motivation of the thesis and contributions.

1.1 Background

The dynamic relational network is increasingly attractive to researchers due to the significant benefits it could bring to social science research. Continuous interaction time-series data is being generated every time and everywhere. There are several examples of such interactions: the replies in social media applications; discussions in online communities; email responses and customer service communications during product purchases etc. An individual's behaviour patterns can be influenced by the communities to some extent. Thus the inference of implicit group mechanisms by observing individuals is valuable for social science studies. However, based only on the continuous time-series data and the event activities, there are several challenges to understand those group features.

Despite the importance of studying the interaction features, current literature on evaluating the continuous relational data remains sparse. Previous works either explored the connections between individuals in a static view or considered the node dynamic relationships within a discrete time domain. There are several random graph models constructed by previous authors who have studied this area. Hanneke [13] modeled the social network with the Exponential Random Graph Models; Sarkar [33] explored the models by static graph features with the Hidden Markov Model; Fu [9] constructed the Markov state space combined with the stochastic blocks to model the network. Discrete models are easier to model but continuous models can give greater flexibility in understanding complex network features. We can view the social network as a graph, define the individuals as nodes and the events between any two nodes as an edge. Some studies explored the edge attributes and grouped them by the level of friendships among group members [19]. However the attributes of edges can be vague and at times may not be obtainable. In addition, a lot of activities in social networks are time-variant. One individual's characteristics could depend on the other nodes along time. In this sense, the actor-based continuous model has been more attractive.

In previous literature, the Stochastic block model has been used as an effective model to evaluate the community structures network[25], [26]. Hawkes process models also have been a popular choice for the events in social media [30]. This thesis is inspired by

Blundell et al.[3] and the Stochastic block model. Blundell’s work emphasised the reciprocating relationships in social media and combines the Hawkes process with the Infinite relational model. However, the inference deduction process was only briefly discussed. In this work, we explain in detail how we performed the inference for each of the parameters. For the community assignment, we did not use the infinite relation scheme, which utilises the Chinese Restaurant Process (CRP). The categorical distribution, in contrast to the infinite number of clusters potentially available in the CRP, specifies a fixed number of clusters in the main model. Initially, we were planning to add the Dirichlet Propagated process [6] into the exogenous effect to study more complicated background effects on community, however due to the heavy coding involved like other advanced models the time is limited and we were unable to do this. However, we will briefly discuss this improvement in the conclusion and extension chapter.

1.2 Research aims

This thesis focuses on researching the relational network with point processes and graph model. It gives a clear write up on the simple Hawkes process, the proposed model Poisson stochastic block model and reciprocal Hawkes stochastic block model. Perform step by step inference and applied on the real world dataset. Discover the continuous interaction data, and shows a comprehensive study path from the simple point process to the reciprocal Hawkes stochastic block model. The result shows that the reciprocal Hawkes model is outperforming other models we examined in this thesis.

For the contributions of this work, previous works didn’t have a write up on how Bayesian inference performed on the stochastic block model based reciprocal Hawkes model. We will show the model constructions in full and the most efficient inference algorithms with using Gibbs and optimal Random walk Metropolis Hasting sampling to simulate the parameters. In addition to the inference, for the generative process, there isn’t literature which clearly explains how to generate the reciprocating relational Hawkes synthetic data to validate the model correctness. We will clearly explain the generative process in Chapter 5.2. Compared with the inspired infinite relational model, we use the categorical distribution instead of allowing infinite number of clusters. This will assign a fixed number of clusters and they are easier to be interpreted and understood compare to the CRP.

1.3 Outline of thesis

The thesis is organized as follows:

Chapter 2 introduces the preliminary knowledge in order to construct the model and more relevant literature to the proposed work. This chapter build the connections on several independent topics and this section is finalized by a summary of the first two chapters.

Chapter 3 start to construct a simple pair of interaction time series by discovering the one-dimension Hawkes process model inference and explore the method of generating the synthetic inhomogeneous data.

Then in Chapter 4 stochastic block Poisson relation model is implemented. Compared to the model in Chapter 3, it extends the single-pair events to the graph-liked multiple pairs events and it considers the individuals behavior are influenced by the communities in which it belongs.

However, the model in Chapter 3 still doesn't consider reciprocity of the social network and the mutually exciting characteristic. Thus, following those models in previous chapters, Chapter 5 introduce the stochastic block based reciprocal Hawkes model and how the synthetic data for this kind of model could be generated. Then the real-world dataset is applied and tested.

Chapter 6 concludes the thesis results. Some limitations regarding to current model and the further areas of research are discussed.

CHAPTER 2

Preliminaries and literature reviews

In this chapter, we will outline the background of the Hawkes process, Bayesian inference methods, the stochastic block model and the dynamic network data. Firstly, section 2.1 will review the point processes, from the general counting process to the self-exciting inhomogeneous process. In section 2.2 We will present the literature and definitions of the stochastic block model. In Section 2.3 we will continue on with the history of the dynamic social network research, previously introduced in the Chapter 1 introduction. In Section 2.4 we briefly review the Bayesian theories and the techniques used in the sampling. Then in section 2.5 we introduce some useful distributions and finally, we will summarize the literature to connect them.

2.1 Point processes

Point processes are statistical models which are used to describe the stochastic, countable point patterns in some space.[5] Each point represents the information of an event that happened at a specific time, or location. There are several definitions of point process in different aspects; In this work, we can limit the definition in the temporal domain, which is that the distribution records a sequence of binary events that happened in continuous time.

This section will start with a general introduction of point process then followed by the definition of counting processes. After introducing the basic definitions of point processes, we will raise the family of stochastic counting processes. Then we develop from Poisson process to the in-homogeneous point process talk about the Hawkes process and some real-world applications.

Definition 2.1.1. *Simple point process [34]*

Let $\{\tau_i, i \in \mathbb{N}\}$ be a sequence of non-negative random variables such that $\forall i \in \mathbb{N}, \tau_i < \tau_{i+1}$. A simple point process in \mathbb{R}_+ is the sequence of $\{\tau_i\}_{i=1,2,\dots}$

The point process as a counting measure usually used to model the occurrences of some phenomenon, from the counting number of customers arrival in a given time problem in queuing theory to the application in computer science, financial market, and social science.

Definition 2.1.2. *Counting process*

For $t \in [0, \infty)$, if $N(t)$ is the number of events occurred from time 0 up to and including current time t , then a random process $N(t)$ is said to be a counting process.

Connecting with point process definition, $N(t)$ is a counting process associated with $\{\tau_i, i \in \mathbb{N}\}$

$$N(t) = \sum_{i \in \mathbb{N}} I_{\{\tau_i \leq t\}}$$

Definition 2.1.3. *Counting process follows:*

1. $N(0) = 0$
2. $N(t)$ is non-negative integers for all $t \in [0, \infty)$
3. for $0 \leq s < t$, $N(t) - N(s)$ shows the number of events that occur in the inter-arrival time interval $(s, t]$.

Following the two definitions of the counting process, homogeneous Poisson processes are the most common point process model. It is a random process in which each event occurs independently at a rate per unit time. It can be applied to examples where the interval is a length of time, such as the number of people coming to the supermarket per hour, the number of accidents per day. The rate of those events coming can be defined as $\lambda(t) = \lim_{h \rightarrow 0^+} \frac{P\{N(t, t+h] > 0\}}{h} = \lim_{h \rightarrow 0^+} \frac{E[N(t, t+h)]}{h}$. Which means λ is the expected number of events in the interval. Before going to the definition of the Poisson point process, we first introduce Poisson probability distribution.

Definition 2.1.4. *Poisson probability distribution*

The Poisson probability distribution defines the probability of getting exactly n times successes during a given time interval. The distribution follows

$$\mathcal{P}(X = n) = \frac{e^{-\lambda t} (\lambda t)^n}{n!}$$

Poisson process possess a nice property which is independent increments. This means for any intervals such that $t_0 < t_1 < t_2 \dots < t_n$ then $N(t_1) - N(t_0) \dots N(t_n) - N(t_{n-1})$ are independent random variables. Define from the point process fashion, we can define Poisson process as following [7]:

Definition 2.1.5. *Poisson process*

The point process $\{N(t), t \geq 0\}$ is a Poisson process with rate λ if

1. $N(0) = 0$;
2. $\{N(t), t \geq 0\}$ has independent increments;
3. $N(t + s) - N(s) = \text{Poisson}(\lambda t)$

So far, we have considered a Poisson process with a constant intensity of λ . This can be generalized by allowing the intensity λ to vary according to a deterministic function of t , which now we extend the constant λ to the function $\lambda(t)$.

Definition 2.1.6. *conditional intensity*

Consider the counting process with historical events $H(\cdot)$, if the non-negative intensity function λ^* exists such that.

$$\lambda(t | \mathcal{H}_t) = \lim_{h \rightarrow 0} \frac{E[N(t+h) - N(t) | \mathcal{H}_t]}{h}$$

λ^* is the conditional intensity function of $N(\cdot)$.

Conditional intensity definition explains that the number of events is expected at time t based on the history of the events. The Poisson process is a memory-less point process; however, in the real world, the history of the events could be useful. The time-variant intensity function could provide more information which is crucial in some areas such as financial market, earthquakes and social interactions. However, human activities are more naturally to be bursting. For example, when a message sent, there is more likely to receive a message back from that person compare to those who never sent a message. In-homogeneous Poisson process is the more general point process model. Built the definition on the conditional intensity, we have the following definition for inhomogeneous Poisson process.

Definition 2.1.7. The point process N is said to be an inhomogeneous Poisson process with the integratable intensity function $\{\lambda(t) \geq 0, t \geq 0\}$, if

1. $N(0) = 0$;
2. The process has independent increments;
3. $N(t) - N(s)$ is Poisson with mean $\int_s^t \lambda(r) dr$

Nonhomogeneous Poisson process has attracted several researchers from mid twenties century. Lewis formulated a point process model which was on the direction to model the Hawkes Process. Then Cox developed a doubly stochastic Poisson process in 1955 [4] which defines the conditional intensity itself to be a stochastic process. Following the significant contributions made by several authors such as Cox, Lewis on the research, Hawkes in 1971 first brings up this new type of self-exciting process as a nonhomogeneous Poisson process. The term self-exciting can be explained as the answer to this question: Could one event trigger other events? If so, how the events will distribute? [38] Hawkes process definition is combined with two parts. One is describing the deterministic background intensity; the other part is the triggering function which can reflect the influence of past events on the conditional intensity. For example, how quick the parent event increases the chances of the child events to happen, how long will the parent event influence lasting. The formal definition can be written as following.

Definition 2.1.8. Hawkes process[22]

Let $N(t) : t \geq 0$ be a counting process with associated history timestamps occurs before time t $\{t_k\}_{k=1..N}$. The point process is said to be a Hawkes process if the conditional intensity function $\lambda(t)$ is:

$$\lambda(t | \mathcal{H}_t) = \mu(t) + \sum_{k: t > t_k} \phi(t - t_k) \quad (2.1.1)$$

Where $\mu(t) > 0$ is the base intensity, and $\phi(\tau) > 0$ is the triggering function. If we will it in the sense of Poisson cluster branching problem, then $\mu(t)$ represent the immigrants intensity and $\phi(\tau)$ denote the off-springs intensity function on $[0, \infty)$. There are multiple choices of defining the triggering response function. The most popular form that being used is the exponential distribution. Which is saying the effect of one event appears will decay following the exponential distribution.

$$\phi(t - t_k) = \beta e^{-\xi(t - t_k)} \quad \beta, \xi > 0 \quad (2.1.2)$$

In equation 2.1.2, β also could be called as jump size, as it accounts for the magnitude of influence, and exponential part with ξ is the memory over the time of decay. Besides exponential kernel, power law kernel is also another popular choice, it is also combined by the magnitude of influence and the memory of decay two parts, however just instead of exponential function, we are using the power-law function.

Hawkes process describe the type of event arrival process, which as one event arriving, the likelihood of observing more events coming will increase. This type of point process has been first applied in earthquake events. The aftershocks and subsequent shocks are able to be predicted through the Hawkes model.[27] It also has a wide applications in time series data including social network, criminology, finance and even bioinformatics.

Specifically considering on social network, Hawkes process as the part of inhomogeneous Poisson process is flexible to be applied on several events which the effect could depend on time. Lawrence et al.[21] use the Hawkes process to explore the behavior of viewer on Youtube and connect it with the trend of video. The similar characteristic types are combined with the motivation and the influence corresponding to the base intensity and triggering function. Hasen et al. [14] built the mutually exciting Hawkes models on the dynamic online advertising click to purchase behaviors. In the next year, Hassan et al. then research on the different user operations on twitter and how they influence others. For example, the model that they built has shown the retweet action could excite the reply and like action among the similar type of users.

Hawkes-IRM model built from Blundell et al. [3] as the inspired paper of the thesis, is combine the Hawkes process with infinite relational model [19] together, which gives a nonparametric model for modelling the relational network. He proposed the mutually exciting Hawkes process to model the reciprocity of the social network, and the infinite relational model is aim for partition each type of the individuals into clusters.

The definition of multivariate Hawkes process is also available. The subscription j means the type j rate of point process. The difference of multivariate situation is we assume when one event happened, all the other event point processes will get triggered to increase the likelihood of next few events happen in different rate.

Definition 2.1.9. *Multivariate Hawkes process*

$$\lambda_j(t | H_t) = \mu_j + \sum_{k=1}^d \rho_{kj} \sum_{\substack{v=1 \\ (t_i < t)}}^{N_k(t)} \phi_{kj}(t - t_v^k) \quad j = 1, 2, \dots, d$$

However, in this thesis although there are multiple individuals in the graph, the interactions we are interested in are still in the type of point-to-point. No matter simple

Hawkes process or multivariate, several estimation techniques has been explained in the literature. Both parametric and non-parametric methods work. Maximum likelihood, nonparametric inference, expectation-maximization algorithm. We will mainly perform the Bayesian inference in this work.

2.2 Social network graph

With the increase of mobile devices and the convenience of communication, the lives of millions of people are flooded with online social networks. Especially under this year's pandemic situation, the various interactions between people are highly dependent on various websites and applications. For example, the like, reply and retweet on Tiktok, Facebook, Twitter; the share and reply on Reddit, StackExchange etc. As a result, the analysis of interactive data on social networks has always attracted people's attention. We will explore not only the Hawkes process, but also different graph models and time series analysis techniques.

We would like to introduce some graph theory terminologies to better describe the models appears in the following chapters.

Definition 2.2.1. *Graph Terminology They are very basic terminologies, however define them properly will make sure the readers understand them well.*

- *Node: Let V denotes the set of vertices in the graph. We will also refer to them as nodes.*
- *Edge: Let E refers to the edges of a network graph which connect the nodes in the network.*
- *Graph: In graph theory, the graph G is comprised of the ordered pairs of vertices and edges $G=(V,E)$.*
- *Directed edge: Edges can be directed, which means that it has a direction, from one node to another node.*
- *Undirected edge: These are edges that just represent there is a link between two nodes; it doesn't matter which node is sending the action or which node is receiving.*

Much of the existing work has been done on modeling dynamic interaction data[33] [13], [9] typically assume the input interaction network has been captured as a sequence of snapshots at discrete time-points. Those literature consider the dynamic interaction is an evolution of a series of static graphs, and using the static graph techniques to solve the problem. They view the dynamic feature as the change of state in Markov setting. We have briefly introduce those discrete and static works in the background. However, using those discrete-time approaches and view them as the state transition is not always a good idea since the state models requires several assumptions made on the dataset. Also when the graph is large, the matrix transition representation could be too huge to handle. Constructing continuous models can more precisely describe the occurrence of the events

and the continuous model gives more flexibility on parameter estimation. Hawkes process discussed in previous subsection is a great choice for modelling the characteristic of a continuous time sequence. Stochastic block model introduced in next subsection will be a good idea for communities detection.

Besides the general social network modelling, the reciprocity is an important characteristic of social interaction data.

Definition 2.2.2. Reciprocity

Reciprocity is a measure of the likelihood of nodes in a directed graph to be mutually linked. [10]

In dynamic relational social network, we observed that the reciprocal effect occurred quite often along the time. When individuals are communicating to each other, it will be more likely to get back a message when they send out a message. When we say the reciprocity characteristic of the interaction, it means the directed edges in the graph tends to get back a link from the reverse direction.

The inspired Hawkes-IRM model of this thesis [3] has taken reciprocity into account in Hawkes process. Blundell et al. assumes the intensity of this reverse response depends on the counting process of the forward action. Tan et al. [35] built their model based on this reciprocal Hawkes model, which considered the reciprocity even more realistic by adding the content into consideration. This paper uses Gaussian Process to measure the significance of messages the individuals sent. To evaluate the reciprocity effect, he proposed the scale parameter in the exponential triggering function as $\beta_{uv} = \exp(r_s(x_{vu}(s)) + s_v(x_{vu}(s)))$ instead of just a random variable. Where $r_s \sim GP(0, k_r)$ represents the receptivity of the significance, $s_v \sim GP(0, k_s)$ represents the significance of the messages and $x_{vu}(s)$ is the edge from v to u . Dubois et al. [6] considered the reciprocity features in the graph model, although it is in a bit different aspect, as he classify the linkage between two nodes into seven types including reciprocity. we have introduced in the stochastic block model section.

2.3 Stochastic block model

Previously we had introduced briefly the stochastic block model to discover the latent structure based on the actors. In this section, we will have a more solid understanding of this model.

The stochastic block model (SBM) is a combination of random graph and block structure, first formally introduced by Holland [16]. The main assumption of this model is that the vertices of the graph are partitioned into several amounts of new blocks and that the probability of an edge between two vertices depends only on the blocks to which they belong. Several authors [26] continued on this idea, and they become interested in such block structures as a posterior. Nowicki and Snijders assume that the vertices of the graph are divided into two unknown blocks in their 1997 paper. This model has attracted more attention as well-performed communities detection. For any graph model, the most natural way to represent it is by the adjacency matrix. The initial idea is to reorganise the matrix to several sub-matrices. Given this idea, the Stochastic block model takes the following

parameters, define the graph as G , there are n vertexes and k number of communities. Let A be the adjacency matrix for G , for any node i, j , A_{ij} are statistically independent with probability Q_{g_i, g_j} , where $K \times K$ connectivity matrix Q is symmetric with $\{0, 1\}$.

Let Y be a random vector with i.i.d components distributed under cluster $\{k_k\}$, then for any nodes $i \neq j$ and $i' \neq j'$, if i and i' are in the same block and j and j' in the same block, then the random vectors Y_{ij} and $Y_{i'j'}$ are identically distributed.

This definition is followed by Holland. To define the more general static stochastic block model, we define the latent class assignment z , $\mathbb{P}(z_i = c) = \theta_c$. Then the joint distribution is defined as

$$\mathbb{P}(z_1 = k_1, \dots, z_n = k_n) = \{\theta_1^{\sum_{i=1}^n 1_{z_i=c}} \dots \theta_k^{\sum_{i=1}^n 1_{z_i=c}}\} \quad (2.3.1)$$

Then the edges of the node depend only on the cluster the nodes belong to. The random vector Y_{ij} is independent and the probabilities of Y_{ij} given the latent class vector z are :

$$\mathbb{P}(Y_{ij} = y | Z = z) = \eta_y(z_i, z_j) \quad (2.3.2)$$

which $\eta_y(z_i, z_j)$ can be interpreted as the mixing rate between the two groups z_i and z_j . Then we can find the conditional distribution of Y given the vector of z, Q, η_y . Then using either the frequentist method such as finding the maximum likelihood estimator, EM algorithms or using the Bayesian approach such as Gibbs sampling.

The stochastic block model is initial used to solve for an undirected graph; soon, it has been extended to a directed graph as well. Then several researchers started to think of the development of the Stochastic block model. Instead of each node only being able to be assigned into one group, why not allow one node belonging to multiple clusters? Airoldi et al.[1] developed the mixed membership stochastic block model, which allows each node to belong to several clusters, but assigning different weights to each group. Then computing the $n \times K$ matrix of membership probabilities.

Till now the model is still being developed on static graphs. However, temporal information is available and useful in many of the social networks. Extending Airoldi's mixed-membership idea, Fu et al.[9] further improved the model by extending the membership probability into each time sequence and each nodes. For example, the mixed membership assumes each individual can belong to several clusters with a probability. For example an individual can have $[0.6, 0.3, 0.1]$ as probability of belonging to three different groups. Then Fu extend to the dynamic case. So one could shows the probability of belonging to each of the group is $[0.6, 0.3, 0.1]$ at time t , but then shows $[0.5, 0.4, 0.1]$ at time $t+1$.

However continuous models based on SBM remain scarce. Dubois et al. [6] first proposed the continuous-time SBM for network-based events. This model considers the sequence of events arises from a nonhomogeneous Poisson process, and the latent group of nodes share similar behaviours. The class assignment vector z continues on the idea of static SBM and discrete-time SBM. Then the model assumes that for any two nodes, the intensity functions are defined by the clusters the nodes belong to. Specifically, they

use a log-linear model:

$$\log \lambda_{i,j}(t \mid \mathcal{H}_t, \beta, \mathbf{z}) = \beta'_{z_i, z_j} \mathbf{s}(t, i, j, \mathcal{H}_t) \quad (2.3.3)$$

which β depends on the clusters, \mathcal{H}_t is the historical events and the statistic s is computed from the history of \mathcal{H}_t . Then this model following Bayesian structures, places a hierarchical prior on β with standard normal distribution and inverse gamma distribution. The number of clusters is flexible using CRP distribution. Then Bayesian inference techniques is applied to estimate the posterior distribution. However, compared to the statistic s_0 (including the intercept, reciprocity, sender in and out degrees). Also, similar to the discrete SBM model, the self-exciting nonhomogeneous process could be a better choice. Since the Hawkes process could represent the burst effect more vividly in the social network rather than consider the in and out degrees of the sender. Also, since there are only seven types of statistics available and sometimes it is not easy to get the information. The reciprocity characteristic of the interaction is one of the most important statistics which can be directly modelled by our proposed model in Chapter 5 without knowing the extra node features.

2.4 Bayesian inference

Bayesian inference is an important topic in statistics which can be applied in solving several real-world problems including Machine learning[36], biological modelling[17], social science[18] and a wide range of other subjects[32]. It is popular as it can provide more explainable results and the ability to reallocate the credibility across parameter values [20]. The advantage of Bayesian inference over traditional frequentist inference is that it can make more speculations and conditions on the observed data, also when new data or more evidence is coming, Bayesian statistics provide us the tool to update the beliefs and the confidences mathematically.

In this section, we will first revise the Bayes theorem, extend to the Markov Chain Monte Carlo (MCMC) sampling methods and discuss Gibbs sampling algorithm and Metropolis Hastings algorithm, and then finally we will talk about the literature that connects Bayesian inference with the Hawkes process.

Definition 2.4.1. Bayes theorem

$$\mathbb{P}(\theta|x) = \frac{\mathbb{P}(x \mid \theta)\mathbb{P}(\theta)}{\mathbb{P}(x)}$$

Bayes theorem is the fundamental formula for the inference and simulation process in Bayesian analysis. $\mathcal{P}(\theta|x)$ represent the posterior distribution given the observed data, $\mathcal{P}(x|\theta)$ is the probability distribution of the observed data given the parameter, $\mathcal{P}(\theta)$ is the prior knowledge of the parameter, and $\mathcal{P}(x)$ is the evidence which is a normalized constant can be computed by the integration of the previous two terms. The formula expresses posterior knowledge as the consequences of prior knowledge and the knowledge derived from observation as likelihood function.

2.4.1 Markov Chain Monte Carlo simulation

Monte Carlo simulation methods occurred to be introduced by Stan Ulam in 1946 [31] when he was trying to perform a more practical and straightforward random simulation to avoid exhaustive combination calculation in playing solitaire. Previously there are several pioneers also trying to discover more feasible and easier simulations, however, due to the limitation of computing tools and computation power, the idea did not been well discovered until Ulam and Metropolis [31] first combined Markov chain into the simulation algorithm.

Markov Chain Monte Carlo (MCMC) is now the most common used way in Bayesian Inference. It is a strategy to obtain samples from desired distribution by explore state space using the Markov chain. The chain is constructed so that the generated samples can match the target distribution. The core idea of MCMC is to explore probabilistic behaviour by random sampling to approximate the property of a complex model. The fundamental theory of Monte Carlo is supported by the Law of Large Number and Central Limit Theorem. With the number of simulation iteration goes to infinity, the generated i.i.d. random variables approach the true posterior distribution.

The Monte Carlo method is a general term for a series of algorithms. It assumes that the probability distribution is known, and the numerical results are obtained by repeated random sampling. There are several algorithms to construct the Markov chain, including the Metropolis Hastings algorithm and the Gibbs sampling.

Gibbs sampling is mainly used to infer parameters such as the distribution having a well formed conditional distribution. To satisfy Gibbs sampling, we need to know the marginal conditional probability. In other words, Gibbs sampling simulates the joint distribution by conditional distribution sampling, and then derives the conditional distribution directly through the simulated joint distribution. Geman introduced the Gibbs sampler for optimisation in a discrete image processing problem without completion [12]. In this algorithm a component in each step is updated from the corresponding conditional posterior.

1. Initialize $x_1 \dots x_n$
2. Sample $x_1^{(i+1)}$ based on the conditional probability $x_1^{i+1} \sim p(x_1^{i+1} | x_2^i, x_3^i, x_4^i \dots x_n^i)$

Gibbs sampling is one of the special cases of Metropolis Hastings. The Metropolis Hastings algorithm developed by Hasting in 1970 [15] has been recognised as a significant contribution in MCMC. Generally speaking, it can be divided into five steps:

1. Current status i .
2. A new state j is generated from the proposed distribution $q(j|i)$. Suppose it is a simple distribution, such as Gaussian distribution. In that case, the value of the uniform distribution can be obtained by mapping the value of the uniform distribution through the inverse function of the Gaussian distribution, or use other sampling methods, such as resampling and rejection sampling.
3. Calculate the acceptance probability. $\alpha(j|i) = \min\{\pi(j)q(i|j)/\pi(i)q(j|i), 1\}$
4. Sample a data point (uniform distribution) $u \sim U[0, 1]$ from $[0, 1]$ interval

5. Compare the randomly generated number with the acceptance rate. If $U < \alpha$, the state will jump from i to j . Otherwise, it will remain in state i .

The Random Walk Metropolis Hastings (RMWH) could be low on the acceptance rate if the proposed distribution scale parameter is not good enough. The optimal scaling method [11] by applying the Robbins-Monro searching process can quickly find out the optimal scaling, which relatively guaranteed the sample acceptance to the specified level. Garthwaite et al. show that the Robbins-Monro process could perform the stochastic search and estimate the best scale parameter. The algorithm of performing this optimal scaling RMWH is mostly the same as the standard algorithm except for updating the scale parameter σ each time. For the univariate target distribution, under Gaussian RWMH proposal, the normal RWMH will have the proposal distribution $y \sim N(x, \sigma^2)$ which σ is fixed until the end of the simulation. This strategy is trying to improve the estimation at each step.

$$\sigma_{i+1} = \sigma_i + \frac{c(1-p^*)}{i} \quad \text{accepted} \quad (2.4.1)$$

$$\sigma_{i+1} = \sigma_i + \frac{cp^*}{i} \quad \text{rejected} \quad (2.4.2)$$

where acceptance rate set to $p^* = 0.44$ following Roberts et al. $c = \frac{\sigma_i}{p^*(1-p^*)}$, we will use this method to sample the proposed model, the stochastic block model based Reciprocal Hawkes model.

2.5 Useful distributions

2.5.1 Dirichlet distribution

Dirichlet distributions are generally used as a prior distribution in Bayesian statistics. In face, Dirichwlet distribution is a high dimension generalization of the beta distribution, and the definition of its probability density function is defined as:

$$f(x_1, \dots, x_n \mid \alpha_1, \dots, \alpha_n) = \frac{1}{B(\alpha)} \prod_{i=1}^n x_i^{\alpha_i-1} \quad (2.5.1)$$

where $B(\alpha) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^K \alpha_i)}$ and $\alpha = (\alpha_1, \dots, \alpha_K)$.

And, The likelihood function is given as:

$$f(x_1, x_2, \dots, x_n \mid k, p_1, p_2, \dots, p_n) = \frac{k!}{x_1! \dots x_n!} \prod_{i=1}^n p_i^{x_i} \quad (2.5.2)$$

2.5.2 Categorical distribution

Categorical distribution is a discrete probability distribution with sample space defined as a set of k individually identified items. Categorical distribution is also a generalization of Bernoulli distribution in the domain of the categorical random variable. The probability mass function of Categorical distribution is defined as:

$$f(y; \{\theta_y\}) = \theta_y$$

which θ_y satisfied : $\left\{ \sum_y \theta_y = 1 \right\}$

2.5.3 Multinomial distribution

Multinomial distribution is a generalization of the binomial distribution in higher dimension space. The probability density function is defined as:

$$f(x) = \frac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k} \quad (2.5.4)$$

$$\text{which } p_k, x_k \text{ satisfied } \left\{ \sum_{k=1}^K p_k = 1, \sum_{k=1}^K x_k = n \right\}$$

2.5.4 Chinese Restaurant Process

As name suggested, this distribution described a table sitting problem. The probability of a customer sitting at a table is calculating from the number of customers already sitting at that table.

$$p(z_i = k \mid z_{1:(i-1)}, \alpha) \propto \begin{cases} n_k & \text{for } k \leq K \\ \alpha & \text{for } k = K + 1 \end{cases} \quad (2.5.5)$$

Where z_i means the table assigned for i^{th} individual.[2]

2.6 Literature summary

The first two chapters have briefly discussed the research in the relational data research in which we reviewed the static relational data, and then in the discrete relational network, several graph models have been implemented.

The development of Hawkes process from simple point processes was presented. The research provides a solid understanding in the Hawkes process and is preparation for the models in the following chapters. Then the literature of the Hawkes process combined with social networks was discussed.

Furthermore, the dynamic graph methods were reviewed. We introduced key terminology before going into the stochastic block model. We discussed the history and reviewed several papers in stochastic block models.

Then we reviewed the Bayes inference techniques, including several Markov Chain Monte Carlo (MCMC) simulation algorithms and reviewed the method of an optimal scaling Metropolis Hastings algorithm.

Finally, we listed out the useful distributions which could be used in later model construction and discussion.

CHAPTER 3

Simple Hawkes Process between two individuals

In this chapter, we are considering only a pair of individuals inside the network and assume the interaction data follows Hawkes process. We will first discuss the likelihood formula of Hawkes process. Then we will have a brief discuss on the kernel choices. Afterwards, we introduce the branching structure and introduce the way to compute the joint likelihood for the inhomogeneous process. Finally, we show the simulations we have performed compare to a real world pair of interaction.

3.1 Likelihood function of Hawkes process

To estimate the parameters from Hawkes processes, no matter to perform the Bayesian posterior inference or to go through the maximum likelihood method to find the estimator, we all need to use the likelihood function of Hawkes process. However, the likelihood function directly applied on the nonhomogeneous point process is not appropriate since the form including integration on each time interval could be too complex. Recall that Hawkes process is defined in Preliminary 2.1.8, define $f^*(t) = f(t|\mathcal{H}_t)$ as the conditional probability density function of the next occurrence time. Then the $\lambda(t)$ can be represented as [29]

$$\lambda(t) = \frac{f^*(t)}{1 - F^*(t)}. \quad (3.1.1)$$

$F^*(t)$ is the cdf of $f^*(t)$, In the paper Rasmussen didn't explain in detailed. Intuitively speaking this definition is the ratio of the event happen in the infinitesimal time interval dt probability to the probability of no new event before time t . [3] This could be equivalent to the rate defined by counting process N_t in preliminary section.

Thus we can derive the likelihood function as below [30]:

$$\lambda(t) = \frac{f^*(t)}{1 - F^*(t)} = \frac{\frac{\partial}{\partial t} F^*(t)}{1 - F^*(t)} = -\frac{\partial}{\partial t} \log(1 - F^*(t))$$

Then integrate on both side

$$\int_{t_n}^t \lambda(s) ds = -\log(1 - F^*(t))$$

Rearrange to get the formula of $f^*(t)$

$$f^*(t) = \frac{\partial}{\partial t} \left(1 - \exp \left(- \int_{t_n}^t \lambda(s) ds \right) \right) = \lambda(t) (1 - F^*(t)) = \lambda(t) \exp \left(- \int_{t_n}^t \lambda(s) ds \right)$$

Let θ represent the family of parameters used in the process

$$L(\theta) = \prod_{k=1}^n f^*(t_k) = \prod_{k=1}^n \lambda(t_k) e^{-\int_{t_{k-1}}^{t_k} \lambda(u) ds} = \prod_{i=1}^n \lambda(t_i) \exp \left(- \int_0^{t_n} \lambda(s) ds \right). \quad (3.1.2)$$

Thus, the Likelihood function of Hawkes process is

$$\mathcal{L} = \left(\prod_{k=1}^N \lambda(t_k) \right) \exp \left(- \int_0^T \lambda(t) dt \right) \quad (3.1.3)$$

3.2 branching structure

Hawkes process can be viewed as a branching process on Poisson process, it will be useful when we are computing the joint likelihood. The original purpose of branching process is to solve the population problem. The individuals could give birth to the offsprings and whenever give a birth, the generation number will adding one more. Generation 0 are coming from the immigrants which are moving in directly without being a child of the ancestor.

In the previous studies of Hawkes process with application on earth quake, Marsan [24] first use it to separate the rate of earthquakes events as background rate and exciting rate. Zhou et al. [37] also used this structure when analysis sparse low-rank network. We can define the branching variable $\mathbf{A} = A_1 \dots A_n$ for events at time $t_1 \dots t_n$ in one dimension. Then $A_{ij}^{(n)}$ represent the branching indicator for node i to node j at time t_n , if this event is an offspring of time t'_n then $A_{ij}^{(n)} = n'$, if this event is an immigrant, then $A_{ij}^{(n)} = 0$ indicate it is not an offspring of any previous event time point.

The probability of A's value would be following:

$$P \left(A_{ij}^{(n)} \right) \propto \begin{cases} \mu, & A_{ij}^{(n)} = 0 \\ \beta e^{-\xi(t - t_{ij}^{(n')})}, & A_{ij}^{(n)} = n', n' \text{ is the parent time-point index} \end{cases} \quad (3.2.1)$$

For simple Hawkes process, we don't need to introduce several node, thus no need for using i, j . Most of the other literature is using n 2-dimension matrix representing. However, by using along the timeline, we can reduce the calculation time and make it more effective in both storage and speed of compiling. We can short write it $A^{(n)} = 0$ for it generate from background event and also $A^{(n)} = n'$ when it parent is n' .

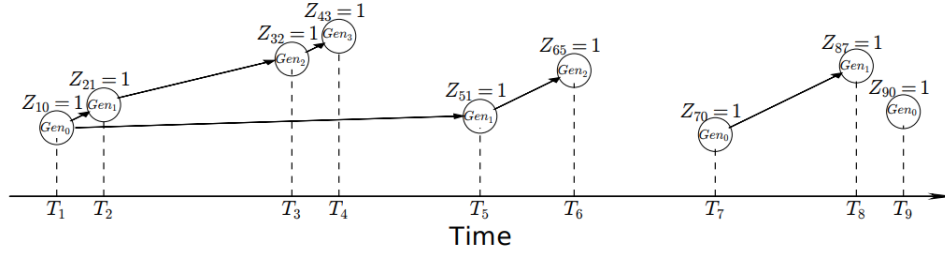


Figure 3.1: Branching process view of Hawkes process [30]

Figure 3.1 can be interpreted as at t_1 , there is an event happened. Then due to t_1 's stimulation, t_2 to t_5 all have some effects from t_1 . They are the generation 1 from event in t_0 . event in t_6 is the offspring of t_5 . At t_7 this is an example of immigrant. Explain in Hawkes process, that means the event in t_7 occurred as purely the exogenous effect. events from t_2 to t_6 is generated by both exogenous effect and the triggering effect from historical events.

3.3 Generated Hawkes process

Lewis and Shedler[23] gives the simulation of inhomogeneous Poisson process algorithm. Which is saying for any bounded intensity, they are simulating points from homogeneous Poisson process and accepting inhomogeneous Poisson process with specific ratio.

The generative process is implemented based on Ogata's modified thinning algorithm [27]. This idea is also coming from branching structure, $\lambda(t)$ is deterministic over period $[t_k, t_{k+1}]$, and at t_{k+1} the point can be viewed as another generation of an inhomogeneous Poisson process.

The algorithm of simulating the simple Hawkes process is as below.

Input: μ, β, δ, T
Initialize $\mathcal{T} = \emptyset, s = 0, n = 0$;
while $s < T$ **do**
 Set $\bar{\lambda} = \lambda(s^+) = \mu + \sum_{\tau \in \mathcal{T}} \beta e^{-\delta(s-\tau)}$;
 Generate $u \sim \text{uniform}(0,1)$;
 Let $w = -\ln u / \bar{\lambda}$;
 Set $s = s + w$;
 Generate $D \sim \text{uniform}(0,1)$;
 if $D\bar{\lambda} \leq \lambda(s) = \mu + \sum_{\tau \in \mathcal{T}} \beta e^{-\delta(s-\tau)}$ **then**
 $n = n + 1$;
 $t_n = s$;
 $\mathcal{T} = \mathcal{T} \cup \{t_n\}$;
 end
if $t_n \leq T$ **then**
 return $\{t_k\}_{k=1,2,\dots,n}$
else
 return $\{t_k\}_{k=1,2,\dots,n-1}$

Algorithm 1: The Ogata's modified thinning algorithm to simulate synthetic data

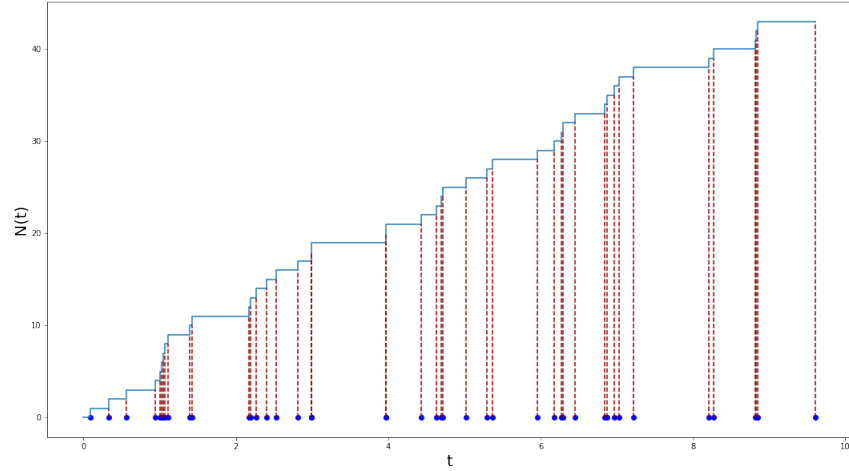


Figure 3.2: Simulated Hawkes process

From figure 3.2, we can see the simulated Hawkes process preserves self-exciting properties. For example at around time point 5, and time point 6, the events appear to be happening clustering together.

3.4 Bayesian Inferences

3.4.1 Joint likelihood for simple Hawkes process

Substitute 2.1.8 and introduce the latent variable 3.2.1 into likelihood formula 3.1.3, we get

$$\begin{aligned}
\mathcal{L} &= \left[\prod_{k=1}^N \lambda(t_k) \right] \exp \left(- \int_0^T \lambda(t) dt \right) \\
&= \left[\prod_{k=1}^N \left(\mu + \sum_{m:t > t_m} \phi(t - t_m) \right) \right] \exp \left(- \int_0^T \left(\mu + \sum_{t > t_m} \phi(t - t_m) \right) dt \right) \\
&= \left[\prod_{k=1}^N \left(\mu^{1_{A^{(k)}=0}} \prod_{m=1}^N \phi(t_k - t_m)^{1_{A^{(k)} \neq 0}} \right) \right] \cdot \\
&\quad \exp \left(- \int_0^T \left(\mu + \sum_{t > t_m} \phi(t, t - t_m) \right) dt \right) \\
&= \left(e^{-\mu T} \mu^{\sum 1_{A^{(k)}=0}} \right) \left(e^{-\frac{\beta}{\xi} \sum_{m=1}^N (1 - e^{-\xi(T-t_m)})} \right)
\end{aligned} \tag{3.4.1}$$

Since $\int_0^T \mu dt = \mu T$ and

$$\begin{aligned}
\int_0^T \sum_{t > t_m} \phi(t - t_m) dt &= \sum_{m=1}^N \int_{t_m}^T \beta e^{-\xi(t-t_m)} dt \\
&= \sum_{m=1}^N \frac{\beta}{\xi} (1 - e^{-\xi(T-t_m)})
\end{aligned} \tag{3.4.2}$$

To compute the posterior distribution, we condition on the joint likelihood for each parameters.

- Parameter μ , given the prior distribution $\mu \sim \text{Gamma}(\alpha_\mu, \beta_\mu)$, then

$$P(\mu | -) \propto \text{Gamma}(\alpha_\mu + \sum 1_{A^{(k)}=0}, \beta_\mu + T) \tag{3.4.3}$$

- given that the jump size parameter β is $\beta \sim \text{Gamma}(\alpha_\beta, \beta_\beta)$, follow from the joint likelihood

$$\begin{aligned}
P(\beta | t, A, \xi) &\propto \left(e^{-\frac{\beta}{\xi} \sum_{m=1}^N (1 - e^{-\xi(T-t_m)})} \right) \left(\beta^{\sum_{m=1}^N \sum_{k=1, k \neq m} 1_{A^{(k)}=m}} \right) \beta_\beta^{\alpha_\beta} \beta^{\alpha_\beta - 1} e^{-\beta_\beta \beta} \\
P(\beta | t, A, \xi) &\sim \text{Gamma}(\sum 1_{A^{(k)} \neq 0} + \alpha_\beta, \frac{1}{\xi} \sum_{m=1}^N (1 - e^{-\xi(T-t_m)}) + \beta_\beta)
\end{aligned} \tag{3.4.4}$$

- The decay rate parameter ξ , given Prior of $\xi \sim \text{Gamma}(\alpha_\xi, \beta_\xi)$ and the joint likelihood,

$$P(\xi) \propto \xi^{\alpha_\xi - 1} \exp \left[-\xi \left(\sum_{k=1, m=1, k \neq m}^N \mathbf{1}(A^{(k)} = m)(t_k - t_m) + \sum_{m=1}^N \frac{\beta}{\xi} (1 - e^{-\delta(T - t_m)}) + \beta_\xi \right) \right] \quad (3.4.5)$$

- Latent parameter A is defined as following categorical distribution with choosing one value among all possible values with given probabilities.

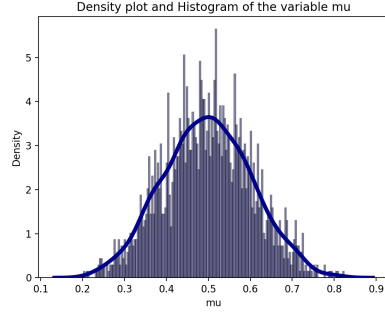
$$P(A^{(n)}) \propto \begin{cases} \mu, & A^{(n)} = 0 \\ \beta e^{-\xi(t - t_{ij}^{(n')})}, & A^{(n)} = n', n' \text{ is the parent time-point index} \end{cases} \quad (3.4.6)$$

Then follows the Gibbs sampling and Metropolis Hastings steps:

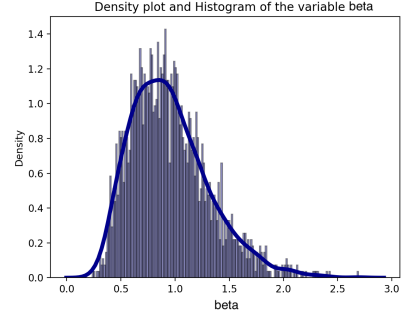
::

```
while  $i \leq N_{simulation}$ :
    Sample new latent variables A using 3.4.6;
    Sample new  $\mu | A_i, \beta_{i-1}, \xi_{i-1}$  using 3.4.3;
    Sample new  $\beta | A_i, \mu_i, \xi_{i-1}$  using 3.4.4;
    Sample new  $\xi | A_i, \mu_i, \beta_i$  using Metropolis-Hastings with ??;
    increment  $i$ .
```

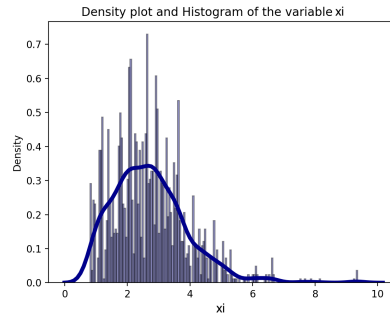
After the Bayesian sampling has been performed, we get the following result plots tested on our synthetic data simulated. From figure 3.3a shows that the histogram of μ is normally allocated around the true value 0.5. The posterior distribution of β in figure 3.3b is also well able to predict. Then this picture 3.3c shows the posterior distribution simulated from Random-walk Metropolis Hastings. The ACF plot 3.3d is trying to see if the data are correlated. The model appear to be free of any autocorrelation after lag 6.



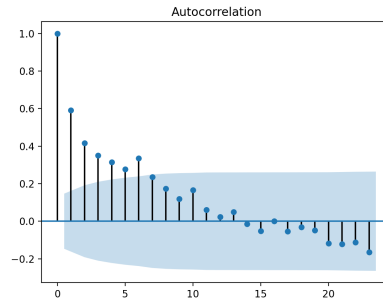
(a) the histogram of μ by burn out first 100, true value = 0.5



(b) The histogram of β by burn out first 100, true value = 1



(c) the histogram of ξ by burn out first 100, true value = 2



(d) the auto correlation plot on decay rate ξ

Figure 3.3: Simple Hawkes process result plots

CHAPTER 4

Poisson cluster model

We know that the Hawkes process on the real line extends the discrete-time series into the continuous setting and could also capture more features of two individuals from the models in the previous chapters. However, it is not that simple as the relational network is not only the interactions between two people. For one individual, we could connect to hundreds of people. Then for everyone inside the network, it will be a huge map. To discover the individual behaviours best reflected by the community effect, we should determine the groups from those instances that have the same behaviours. Thus, instead of building a model between any pairs of individuals, we model the links between any two of the groups, and the instances that belong to that group will interact following that pattern.

As we have discussed in the preliminary and literature chapter, the stochastic block model usually modelled with static and discrete time. The continuous model proposed is still using the node features inherited from the static SBM features. We combine the homogeneous Poisson process with the clustering model together to better model the temporal information. In this chapter we propose the stochastic block model based Poisson model.

Consider we observe in a fixed time interval $(0, T]$, there are N individuals in the graph. The interaction events occurs between N individuals. We assume each node i belongs to one of the existing K clusters and the interactions between any two nodes depend only on the clusters which the individuals belonged to. That is, the event intensity parameters of those interactions are defined based on the links between clusters.

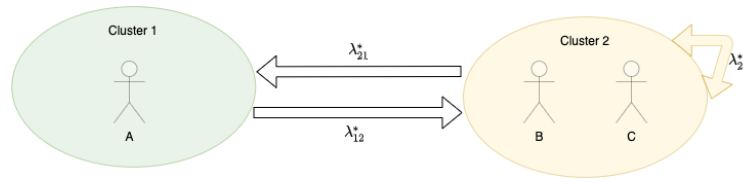


Figure 4.1: SBM based Poisson model

For example, as figure 4.1 show, we have $N = 3$ nodes and $K = 2$ clusters. Cluster 1 contains node 1, cluster 2 contains node 2,3. We assume there are K^2 Poisson processes with given intensity parameters $\lambda_{12}^*, \lambda_{21}^*, \lambda_{22}^*$. (λ_{11}^* is not being used in this example, as we don't interested in the individual text to itself.)

We can now define the model parameters as following:

- Define the time series data between node i and node j is t_{ij} , assume we have n records

and the time series data follows the Poisson process.

$\{t_{ij}\}_{i,j,n} \sim \text{Poisson Process } (\lambda_{z_i z_j});$

- Define the event intensity $\lambda_{k_1 k_2}, \forall k_1, k_2 \in \{1, \dots, K\}$. Then the vector $\lambda = (\lambda_{k_1 k_2} | z_i = k_1, z_j = k_2)$, and the priors are $\lambda_{k_1 k_2} \sim \text{Gamma}(a_\lambda, b_\lambda), \forall k_1, k_2 \in \{1, \dots, K\}$.

- The latent variable z represents the cluster assignment. $z_i = k$ means i^{th} individual is belong to k^{th} community. Then they can be viewed as the probabilities of a K way categorical event, $z_i \sim \text{Categorical}(\pi)$ for all i in N . The $\pi = (\pi_1 \dots \pi_K)$ represent the probability of one specific community to be chosen. π follows $\sum_{k=1}^K \pi_k = 1$.

- The prior of π follows Dirichlet distribution with hyper parameter $\alpha = (\alpha_1 \dots \alpha_K)$, $\pi \sim \text{Dirichlet}(\alpha)$. As $\mathbb{P}(z_i = k) = \pi_k$, as $\sum_{k=1}^K \pi_k = 1$, which satisfies a probability distribution again. In Bayesian setting, π which it could draw from $\text{Dir}(\alpha)$ with hyper-parameter α . A uniform prior can obtained by setting $\alpha = (1, 1 \dots 1)$.

4.1 Generative process

To generate the synthetic data for testing the performance and inference accuracy, we will talk about how the generative process work for this model. We are using the thinning algorithm [27] to generate the time sequences following a Poisson Process. Suppose we have N nodes and K clusters, then we will consider $K * K$ such Poisson processes time series, and randomly assign those time-points to $N * N - N$ pairs. The algorithm pseudo-code is written as following. $k_1, m_1, \dots, k_K, m_K \in (1..K)$

```

input  $K, N, z, \lambda, T$ ;
for each  $(\lambda_{k_1 m_1} \dots \lambda_{k_K m_K})$  do
    Initialize  $n = 0, t_0 = 0$ ;
    while True do
        Generate  $u$  from uniform(0,1);
        Let  $w = -\ln u / \lambda$ ;
        Set  $t_{n+1} = t_n + w$ ;
        if  $t_{n+1} \geq T$  then Store  $\{t_n\}$  in  $t_{km}$ ;
        else set  $n=n+1$ ;
    end
end
for  $i, j$  which  $z_i = k, z_j = m$  do
    | Uniformly random assign  $\{t_p\}_{km}$  to  $\{t_p\}_{ij}$ ;
end

```

Algorithm 2: Simulate synthetic data of Poisson Stochastic block model

We use the same example in figure 4.1, there are 3 nodes and the assignment of z is given as $z_1 = 1, z_2 = 2, z_3 = 2$. Then since both node 2 and node 3 are in cluster 2, we will assign the time points randomly to the $\{t_{ij}\}$ node 1 to 2 and node 1 to 3 in the time period $(0, \dots, T]$. The other node pairs are assigning as table 4.1 below. Similarly, we can assign the time points to cluster 2 to 1 events and the events which cluster 2 connect to itself. For example define cluster 2 to 2 as $\lambda = 1$ and we can randomly assign the time points to node pairs (2,2), (2,3), (3,2), (3,3). Cluster 1 to itself parameter is generated but the

time-stamps corresponding is not been assigned to any pairs appeared. As in this example, we only have one individual group one and it is not useful to text yourself.

sender	receiver	time	cluster(k,m)
1	2	[1.690, 1.993, 2.275, ...]	cluster(1,2)
1	3	[1.605, 3.904, 4.104,...]	cluster(1,2)
2	3	...	cluster(2,2)
3	2	...	cluster(2,2)
2	1	...	cluster(2,1)
3	1	...	cluster(2,1)

Table 4.1: Example node pairs on figure 4.1

The column cluster(k,m) in table 4.1 means the clusters of the senders and receivers belonging to. Which is the representation of the intensity parameter they are generating from.

4.2 Inference

We are using the Gibbs sampling for posterior inferences. In Bayesian inference, we would like to know the joint distribution first then find out the conditional posterior distribution for each parameter.

4.2.1 Joint distribution

$$\begin{aligned}
& \mathcal{L} \left(\{t_{ij}\}_n, \{\lambda_{k_1 k_2}\}_{k_1, k_2}, \{z_i\}_i, \pi \right) \\
&= \left[\prod_{i \neq j} P \left(\{t_{ij}\}_n \mid \{\lambda_{k_1 k_2}\}_{k_1, k_2}, z_i, z_j \right) \right] \left[\prod_{k_1, k_2} P(\lambda_{k_1 k_2}) \right] \left[\prod_i P(z_i \mid \pi) \right] P(\pi \mid \alpha) \\
&\propto \left[\prod_{i \neq j} e^{-\int_0^T \lambda_{z_i z_j} dt} \left(\prod_i^n \lambda_{z_i z_j} \right) \right] \left[\prod_{k_1, k_2} (\lambda_{k_1 k_2})^{a_{\lambda}-1} e^{-b_{\lambda} \lambda_{k_1 k_2}} \right] \left[\prod_k (\pi_k)^{\sum_i^n \mathbf{1}(z_i=k)} \right] \left[\prod_k (\pi_k)^{\alpha_k-1} \right] \\
&\propto \left[\prod_{i \neq j} e^{-\lambda_{z_i z_j} T} (\lambda_{z_i z_j})^{N_{ij}} \right] \left[\prod_{k_1, k_2} (\lambda_{k_1 k_2})^{a_{\lambda}-1} e^{-b_{\lambda} \lambda_{k_1 k_2}} \right] \left[\prod_k (\pi_k)^{m_k + \alpha_k - 1} \right] \quad (4.2.1)
\end{aligned}$$

where N_{ij} represents the number of interactions from node i to node j in the time interval $(0, T]$ and $m_k = \sum_i \mathbf{1}(z_i = k)$ is the number of nodes belonging to the k th cluster.

4.2.2 Posteriors

The posterior full conditional distribution for each parameters is as following:

- Posterior of z

$$\mathcal{P}(z_i = k \mid \{\lambda_{k_1 k_2}\}_{k_1, k_2}, t, z_{-i}, \pi) \propto \pi_k \prod_{j: j \neq i} e^{-\lambda_{k z_j} T} \lambda_{k z_j}^{N_{ij}} e^{-\lambda_{z_j k} T} \lambda_{z_j k}^{N_{ji}} \quad (4.2.2)$$

Define z_{-i} as the vector z by removing the i^{th} element of the vector. N_{ij} and N_{ji} represents the number of interactions happened from node i to node j , and node j to node i in the time T .

The ratio of choosing z_i to be k community is:

$$ratio = \left(\frac{\pi_k \prod_{j,j \neq i} e^{-\lambda_{kz_j} T} \lambda_{kz_j}^{N_{ij}} e^{-\lambda_{z_j k} T} \lambda_{z_j k}^{N_{ji}}}{\sum_k^K \pi_k \prod_{j,j \neq i} e^{-\lambda_{kz_j} T} \lambda_{kz_j}^{N_{ij}} e^{-\lambda_{z_j k} T} \lambda_{z_j k}^{N_{ji}}} \right) \quad (4.2.3)$$

- Posterior of π

$$\mathcal{P}(\pi_k | z, \lambda, \{t_{ij}\}_n) \propto \mathcal{L} \left(\{t_{ij}\}_n \right)_{i,j,n}, \{ \lambda_{k_1 k_2} \}_{k_1, k_2}, \{z_i\}_i, \pi_{-k} \quad (4.2.4)$$

$$\propto \prod_{k=1}^K \pi^{(\sum_{i=1}^n 1_{\{z_i=k\}}) + \alpha - 1} \quad (4.2.5)$$

We can see that the posterior function of π is in the format of Dirichlet distribution, thus π is

$$\pi \sim Dir \left(\sum_i^n (1_{\{z_i=k\}}) + \alpha, \dots \sum_i^n (1_{\{z_i=K\}}) + \alpha \right) \quad (4.2.6)$$

- Posterior of λ

Let $m_{k_1 k_2} = \sum_{i \neq j} \mathbf{1}(z_i = k_1, z_j = k_2)$ Then we get

$$\lambda_{k_1 k_2} \sim \text{Gamma} \left(\sum_{(i,j): z_i=k_1, z_j=k_2} N_{ij} + a_\lambda, m_{k_1 k_2} T + b_\lambda \right).$$

4.3 Results

We first test our results on the generated synthetic data to verify if the parameters has been predict correctly. The data is generated through the algorithms and techniques we go through from 4.1, and the data statistics is shown as bellow.

Synthetic data statistics	
Node number	100
Edge number	7497
Number of clusters	5
The last event time	9.8
The given intensity matrix	Gamma random variables with shape parameter 1

As figure 4.2a shows, although the histogram looks difference between the values, the number of individuals from group counts are look similar. It is just label mismatch as we don't need to specify the index. For example, we say group 0 represents the type

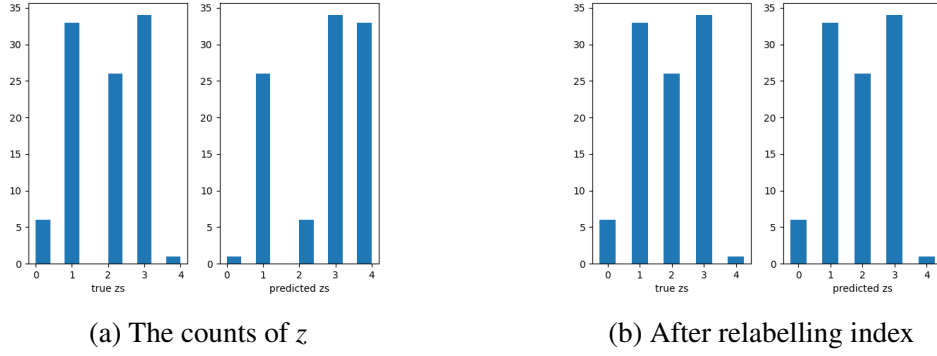


Figure 4.2: The counts for each index in z

of individuals who actively social, then for the prediction shows in figure 4.2b, group 2 represent the same type of the people as group 0 in figure 4.2a did. Thus, after reorganize of the index, figure 4.2b shows that the predicted counts is perfectly match with the true counts. This is the result for the class assignment parameter z after 1000 iterations of MCMC.

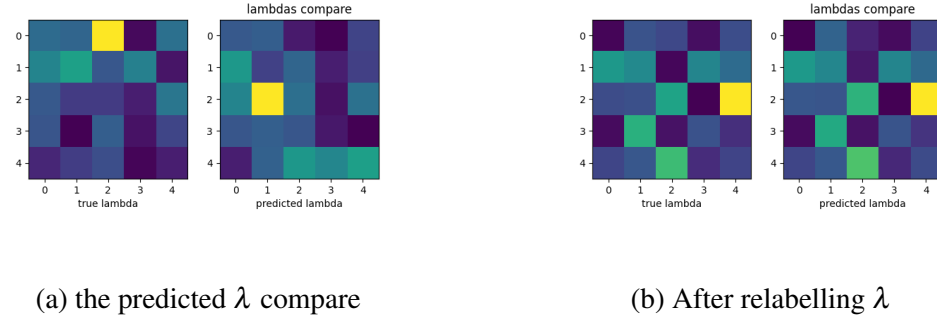


Figure 4.3: λ comparison

Figure 4.3 shows the 5×5 color grid of λ true value and predicted value. The lighter the color it is, the larger the value it represent. Similar to figure 4.2 explained, the label mismatch is fine, thus we swap the columns and rows based on the mismatched z value. We can see that the predicted value is quite similar.

The steady log-likelihood after iteration 10 shows that our model is working well. However the log-likelihood metric and AIC criteria are used in model compare, so we will introduce a bit more in next chapter. The real world dataset applied in this model will also be discussed in next chapter.

However, there are deficits in this model. The homogeneous Poisson process event intensity is fixed rate. However those interaction data are not normally coming with the fixed rate. The message is more likely to be replied from the person who sent that message rather than someone else you still didn't even talk to him before. In this Poisson

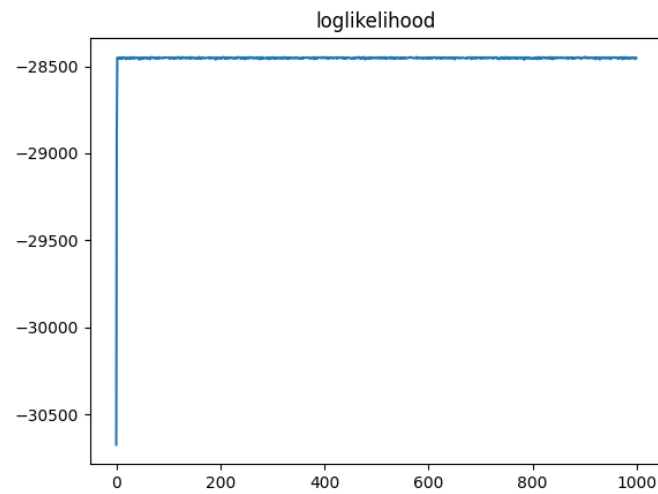


Figure 4.4: log likelihood of the synthetic dataset

stochastic block model it will assume that you will reply them with the same rate if they both belong to a group. We can improve this and we will apply the models on real world datasets and discuss in next chapter.

CHAPTER 5

Mutually exciting Hawkes interaction model

Inspired from the previous chapter, we would like to think: After considering the community effects on the graph, what if those interaction intensities are not a fixed value but depends on the history? Is the interaction from an individual to the other would stimulate the reverse event? In this chapter, we will introduce our primary model: the Reciprocal Hawkes with the stochastic block model. The improvement from last chapter is, instead of homogeneous event intensities between groups, we are taking the reciprocity characteristic into consideration by cooperating with Mutually exciting process. We will start with the model specification, then followed by the generative algorithm. The Bayes Inference then will be fully explained. Finally, we will apply to the real-world dataset and discuss the results.

5.1 Reciprocal relation characteristic

Reciprocity is a measure on how likely the directed edges in the graph get back a link from reverse direction. As Figure 5.1 shows, those lines are the events arriving. Red



Figure 5.1: Reciprocity events example

represents A to B's message, blue represent B to A's message. The actions from A to B is triggering the exciting effect of B to A, and the action of B to A stimulate the events to be more often happened on A to B, which is the idea of mutually exciting. Due to this characteristics obsessed by the interaction data, we would like to introduce mutually exciting Hawkes process which can model such behavior.

Definition 5.1.1. *Mutually exciting Hawkes process*

Consider a series of m counting processes $\{N_1(\cdot), \dots, N_m(\cdot)\}$ denoted N . $N_{k_1 k_2}$ means the counting process of k_1 cluster to k_2 . Then $N_{k_1 k_2}(\cdot)$ has conditional intensity of the form

$$\lambda_{k_1 k_2}^*(t) = \mu_{k_1 k_2} + \int_{-\infty}^t g_{k_1 k_2}(t-u) dN_{k_2 k_1}(u)$$

for some $\lambda_{k_1 k_2} > 0$ and kernel function $g_{k_1 k_2} : (0, \infty) \rightarrow [0, \infty)$, then N is called a mutually exciting Hawkes process.

5.2 Model specification

Similar to the previous model, inside the graph, there are N individuals and the observed time interval is $(0, T]$. The interactions time points between N individuals. We assume

that each node i belongs to one of the existing K clusters, and the interactions between any two nodes are pairwise dependent on the clusters they are being. Moreover, instead of assuming that the events intensity follows a Poisson process, we assume that the events happened between any two nodes follows a mutually exciting Hawkes process.

This mutually exciting Hawkes process can better capture the reciprocity in a social network compare to the memory-less Poisson process and the simple Hawkes process. As the past events affect the probability of future event occurrence based on the series of events interact with each other, the interactions governed by Hawkes process from the node in cluster k_1 to the node in cluster k_2 will depend upon the Hawkes process governing events from cluster k_2 to cluster k_1 .

We assume now that the observations time series $\{t_{ij}^{(n)}\}_{i,j,n} \sim$ mutually exciting Hawkes process.

Let's specify each parameter been used to define the model.

5.2.1 Branching structure A_{ij}

As we introduced in Chapter 3 3.2 The distribution of $A_{ij}^{(n)}$ follows

$$A_{ij}^{(n)} \text{ as } P(A_{ij}^{(n)}) \propto \begin{cases} \lambda_{z_i z_j}, & A_{ij}^{(n)} = 0 \\ \beta e^{-\xi(t_{ij}^{(n)} - t_{ji}^{(n')})}, & A_{ij}^{(n)} = n' (n' : t_{ji}^{(n')} < t_{ij}^{(n)}) \end{cases}$$

Where $\beta, \xi > 0$, the first part $A_{ij}^{(n)} = 0$ represents that the interaction $E_{ij}^{(n)}$ is instantiated by the exogenous rate $\lambda_{z_i z_j}$ and the second part $A_{ij}^{(n)} = n'$ means that $E_{ij}^{(n)}$ is triggered by the historical oppositely-directed interaction $E_{ji}^{(n')}$ with exponential kernel.

5.2.2 The class assignment z_i

Similar to the previous chapter, the latent variable z represents the cluster assignment. $z_i = k$ means i^{th} individual is belong to k^{th} community.

Given the prior $z_i \sim \text{Categorical}(\pi)$ for all i in N .

5.2.3 Categorical distribution parameter π_i

Similar to the model from previous chapter, π follows Dirichlet distribution with hyper parameter α , $\pi \sim \text{Dirichlet}(\alpha)$. As $\mathbb{P}(z_i = k) = \pi_k$, and the π follows $\sum_{k=1}^K \pi_k = 1$. π could draw from $Dir(\alpha)$ with hyper-parameter α . Set $\alpha = 1$ for a uniform prior.

5.2.4 Intensity function related parameters

In mutually exciting Hawkes process with exponential kernel $\phi(t)$, the event intensity $\lambda_{k_1 k_2}$, let $t_{ij,m}$ represent the event occurrence time from node i to node j at time index m . where k_1, k_2 belongs to the K clusters is defined as

$$\lambda_{k_1 k_2} = \mu_{k_1 k_2} + \sum_{m: (t_m < t)} \beta \exp(-\xi(t - t_{k_2 k_1, m}))$$

Which $\mu_{k_1 k_2}$ is the base intensity, β is often accounts for the magnitude of influence and exponential part with ξ is the memory over the time of decay.

Given the priors for μ, β, ξ are Gamma distribution, $\mu_{k_1 k_2} \sim \text{Gamma}(a_\mu, b_\mu), \beta \sim \text{Gamma}(a_\beta, b_\beta), \xi \sim \text{Gamma}(a_\xi, b_\xi), \forall k_1, k_2 \in \{1, \dots, K\}$.

5.3 Generative process

Similar to previous chapter, we first generate $N^2 - N$ time points with Poisson process using thinning algorithm. Then we can view the Hawkes process as a Poisson process adding another Poisson process when it get triggered. Then we compare all candidate occurrence time points to find the smallest value, then this value is set as the occurrence time of the actual new interaction and also the new starting time point t_{new} . Due to the memory-less property of the Poisson process, all these $N^2 - N$ pairs can re-generate new candidate time sequences from the new starting time point. A new membership distribution will also be accordingly generated for the triggering node. Then repeat the steps of selecting minimum value of occurrence time and re-generate with the new triggering effect add on. Initialise $t^{(*)} = 0, \{n_{ij}\}_{i \neq j} = 0$ and generate $\left\{ \left\{ t_{ij}^{(n)} \right\}_{n=1}^{n_{ij}} \right\}_{i \neq j}$ for all the potential nodal pairs $\{(i, j) : i \neq j\}$ according to the following procedure.

- 1. Generate $N^2 - N$ candidate time points $\{\bar{t}_{ij}\}_{i \neq j}$, where each candidate $\bar{t}_{ij} = \min \left(\left\{ \bar{t}_{ij}^{(l)} \right\}_l \right)$ is the minimum of a time sequence $\left\{ \bar{t}_{ij}^{(l)} \right\}_l$ generated from a Poisson point process in the time period $(t_{curr}, T]$, with the rate function as:

$$\left\{ \bar{t}_{ij}^{(l)} \right\}_l \sim \text{PP} \left(\lambda_{z_i z_j} + \sum_{n=1}^{n_{ji}} c \cdot e^{-\xi(t - t_{ji}^{(n)})} \right)$$

If $\{\bar{t}_{ij}\}_{i \neq j} = \emptyset$, stop generating interactions.

- Secondly, we obtain the earliest occurrence time and its corresponding nodal pair:

$$(i^*, j^*) = \underset{(i, j)}{\text{argmin}} \left\{ \bar{t}_{ij} \right\}_{i \neq j}, \text{ set } n_{i^* j^*} = n_{i^* j^*} + 1, t_{i^* j^*}^{(n_{i^* j^*})} = t^{(*)} = \bar{t}_{i^* j^*}$$

and go back to step 1.

5.4 Inference

Similar to the Poisson stochastic block model, we would like to use MCMC sampling techniques to simulate the posteriors. In this chapter, we are incorporated with the more advanced adapted optimal Metropolis Hastings with Gibbs sampling for posterior inferences. In Bayesian inference, we would like to know the joint distribution first then to find out the conditional posterior distribution for each parameter.

5.4.1 Joint distribution

Before writing derivation of the joint distribution, notice that the integrals and the summation can be swapped during calculation

$$\int_0^T \beta e^{-\xi(t - t_{ji}^{(n')})} dt = \frac{\beta}{\xi} (1 - e^{-\xi(T - t_{ji}^{(n')})}) \quad (5.4.1)$$

$$\begin{aligned}
& P\left(\left\{t_{ij}^{(n)}\right\}_{i,j,n}, \left\{A_{ij}^{(n)}\right\}_{i,j,n}, \left\{\mu_{k_1 k_2}\right\}_{k_1, k_2}, \left\{z_i\right\}_i, \pi \mid \beta, \xi, \alpha\right) \\
&= \left[\prod_{i \neq j} P\left(\left\{t_{ij}^{(n)}, A_{ij}^{(n)}\right\}_n \mid \mu_{z_i z_j}, \beta, \xi\right)\right] \left[\prod_{k_1, k_2} P\left(\mu_{k_1 k_2}\right)\right] \left[\prod_i P\left(z_i \mid \pi\right)\right] P(\pi \mid \alpha) \\
&\propto \left[\prod_{i \neq j} e^{-\mu_{z_i z_j} T} \left(\mu_{z_i z_j}\right)^{\sum_n \delta_{A_{ij}^{(n)}=0}}\right] \left[\prod_{k_1, k_2} \left(\mu_{k_1 k_2}\right)^{a_\mu-1} e^{-b_\mu \mu_{k_1 k_2}}\right] \left[\prod_k \left(\pi_k\right)^{m_k+\alpha_k-1}\right] \\
&\cdot \left[\prod_{(i,j,n) \in E_{ij}^{(n)}} \left(e^{-\frac{\beta}{\xi} \left(1-e^{-\xi \left(T-t_{ji}^{(n)}\right)}\right)} \prod_{n': t_{ji}^{(n')} < t_{ij}^{(n)}} \left(\beta e^{-\xi \left(t_{ij}^{(n)}-t_{ji}^{(n')}\right)}\right)^{\delta_{A_{ij}^{(n)}=n'}}\right)\right] \quad (5.4.2)
\end{aligned}$$

$N_{ij}^{(A)} = \sum_n \delta_{A_{ij}^{(n)}=0}$ represents the number of interactions from node i to node j in the time interval $(0, T]$ and is instantiated by the base rate at the same time. $m_k = \sum_i \mathbf{1}(z_i = k)$ is the number of nodes belonging to the k th cluster.

5.5 Posteriors

5.5.1 Posterior of A

The branching variable follows the $A \sim \text{Categorical}(p)$, and the probability of chosen is as following:

$$P\left(A_{ij}^{(n)}\right) \propto \begin{cases} \mu_{z_i z_j}, & A_{ij}^{(n)} = 0 \\ \beta e^{-\xi \left(t_{ij}^{(n)}-t_{ji}^{(n')}\right)}, & A_{ij}^{(n)} = n' \left(n' : t_{ji}^{(n')} < t_{ij}^{(n)}\right) \end{cases}$$

5.5.2 Posterior of π

$$\begin{aligned}
& \mathcal{P}(\pi_k | z, \lambda, \{t_{ij}^{(n)}\}) \propto L\left(\left\{t_{ij}^{(n)}\right\}_{i,j,n}, \left\{\lambda_{k_1 k_2}\right\}_{k_1, k_2}, \left\{z_i\right\}_i, \pi\right) \propto \prod_{k=1}^K \pi^{\left(\sum_{i=1}^n \mathbf{1}_{\{z_i=k\}}\right)+\alpha-1} \\
& \pi \sim \text{Dir}\left(\sum_i^n (1_{\{z_i=k\}}) + \alpha, \dots, \sum_i^n (1_{\{z_i=K\}}) + \alpha\right) \quad (5.5.1)
\end{aligned}$$

5.5.3 Posterior of μ

$$\begin{aligned}
& P\left(\mu_{k_1 k_2} \mid -\right) \propto e^{-\left(\sum_{i \neq j} \mathbf{1}_{z_i=k_1, z_j=k_2} T + b_\mu\right) \mu_{k_1 k_2}} \left(\mu_{k_1 k_2}\right)^{\sum_{(i,j): z_i=k_1, z_j=k_2} \sum_n \left(\delta_{A_{ij}^{(n)}=0}\right) + a_\mu - 1} \\
& \mu_{k_1 k_2} \sim \text{Gamma}\left(\sum_{(i,j): z_i=k_1, z_j=k_2} N_{ij}^{(A)} + a_\mu, \sum_{i \neq j} \mathbf{1}(z_i = k_1, z_j = k_2) T + b_\mu\right) \quad (5.5.2)
\end{aligned}$$

For $N_{ij}^{(A)} = \sum_n \delta_{A_{ij}^{(n)}=0}$.

5.5.4 Posterior of β

$$P(\beta | -) \propto e^{-\left(\sum_{(i,j,n) \in E_{ij}^{(n)}} \frac{1}{\xi} \left(1 - e^{-\xi(T-t_{ij}^{(n)})}\right) + b_\beta\right)} \beta^{\sum_{(i,j,n) \in E_{ij}^{(n)}} \delta_{A_{ij}^{(n)} > 0} + a_\beta - 1}$$

Thus, we get:

$$\beta \sim \text{Gamma} \left(\sum_{(i,j,n) \in E_{ij}^{(n)}} (\delta_{A_{ij}^{(n)} > 0}) + a_\beta, \sum_{(i,j,n) \in E_{ij}^{(n)}} \frac{1}{\xi} \left(1 - e^{-\xi(T-t_{ij}^{(n)})}\right) + b_\beta \right) \quad (5.5.3)$$

5.5.5 Posterior of ξ

$$P(\xi | -) \propto P(\xi) \prod_{(i,j,n) \in E_{ij}^{(n)}} \left(e^{-\frac{\beta}{\xi} \left(1 - e^{-\xi(T-t_{ij}^{(n)})}\right)} - \sum_{n': t_{ji}^{(n')} < t_{ij}^{(n)}} \delta_{A_{ij}^{(n)} = n'} \xi \left(t_{ij}^{(n)} - t_{ji}^{(n')}\right) \right) \quad (5.5.4)$$

$$\propto \xi^{\alpha_\xi - 1} \exp \left(-\xi \left(\sum_{(i,j,n) \in E_{ij}^{(n)}} \delta_{(A_{ij}^{(n)} = n')} (t_{ij}^{(n)} - t_{ji}^{(n')}) + \sum_{n'=1} \frac{\beta}{\xi} (1 - e^{-\xi(T-t_{ij}^{(n)})}) + \beta_\xi \right) \right) \quad (5.5.5)$$

ξ is not a simple distribution, we will use the adapted Metropolis Hastings with Robins Monro algorithm [11] to simulate. Following the typical Metropolis Hastings, we can calculate the acceptance probability $H(x', x_t) = \min \left(1, \frac{\pi(x') q(x_t | x')}{\pi(x_t) q(x' | x_t)} \right)$ and simplified it with logarithm.

5.5.6 Posterior of z_i

The ratio of choosing z_i to be k community is:

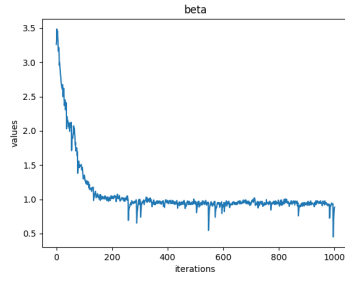
$$ratio = \left(\frac{\pi_k \prod_{j, j \neq i} e^{-\mu_{kz_j} T} \mu_{kz_j}^{N_{ij}(A)} e^{-\mu_{z_j k} T} \mu_{z_j k}^{N_{ji}}}{\sum_k \pi_k \prod_{j, j \neq i} e^{-\mu_{kz_j} T} \mu_{kz_j}^{N_{ij}(A)} e^{-\mu_{z_j k} T} \mu_{z_j k}^{N_{ji}(A)}} \right)$$

5.6 Results from generated data

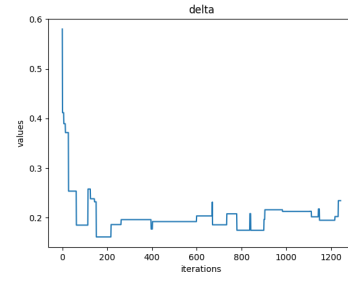
Tested on the synthetic data we generated with parameter $\beta = 1.0, \xi = 0.2, \mu$ as figure 5.2c left hand side shows. Figure 5.2a shows that over the iteration, β is stable after some iterations and gives $\beta = 0.978$. Figure 5.2b trace plot shows that the ξ is also stabilized over around 0.2. Then figure 5.2c compares the true parameter μ with the predicted value. The more light color mean the larger value. We can see that they are mostly same, except the true parameter of $\mu_{00}, \mu_{03}, \mu_{34}, \mu_{44}$ is a bit larger than the predict, but they are inside the tolerate range.

5.7 Application on real world dataset

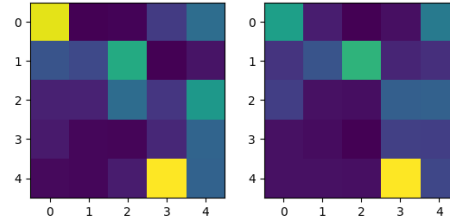
We fit in the College message dataset collected from University of California online social app. The dataset is obtained from the private messages in the online social network



(a) Parameter β trace plot



(b) The optimal Metropolis Hastings ξ trace plot



(c) the base intensity plot

Figure 5.2: The posterior inference plot

from University of California.[28].

College data statistics	
Node number	1899
Number of clusters	Unknown, depends on the problem context setup
Number of interaction pairs	59835
The mean inter-arrival events time	279(s)

There are in total 59835 interactions on the temporal domain and 20296 static edges. Static edge means there are two individuals have a conversation. The users in the College social network can search another users and initiate a conversation. Each entry inside data is combined with the sender, receiver and the UNIX time.

For the purpose of finding out a characteristic belong the latent social structure, for example, as figure 5.3 shows, this could be the after-training example. In which we have fixed the shaded area as three communities. A,B,C,D,E are human in the social network. Then after the training, it shows A and B belong to the green cluster, C is in the purple cluster and D, E is in yellow community. Communications between A and B (including A to B and B to A) will follow the event intensity λ_{11} . Then we fit the college dataset with three groups, then if the result shows that the intensity of λ_{11} , λ_{22} , λ_{13} is all shows to be high, then A and B could be the type of people who always like to initiate a talk

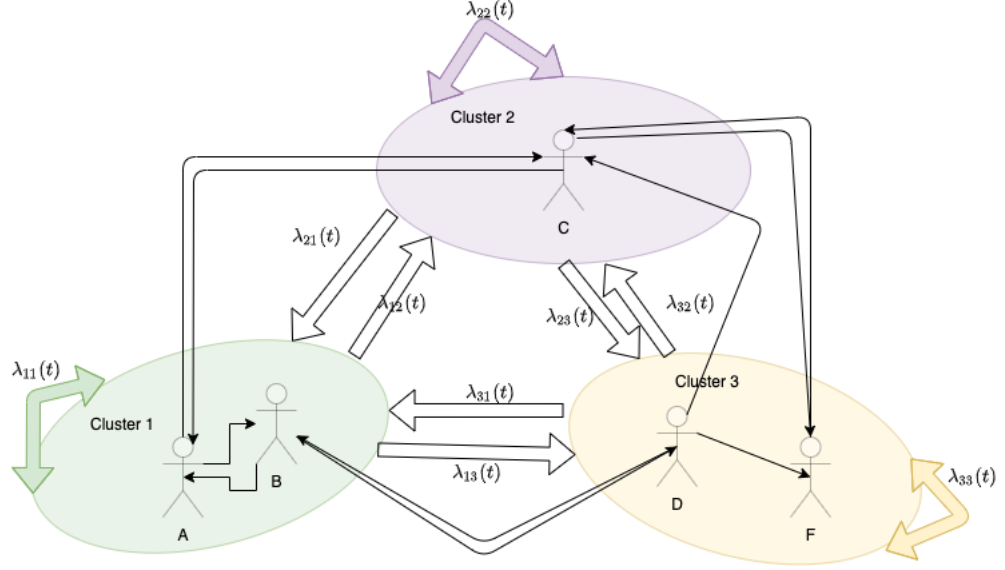


Figure 5.3: Example training of Reciprocal Hawkes Stochastic Block Model

and response messages fast. If λ_{21} appears to be lower than λ_{23} then it could show that the people in group 2 are more interested in being together with individuals coming from group 3 instead of group 1.

We get a value of base intensity 3×3 matrix,

$$\begin{pmatrix} 0.07855476 & 0.050684 & 0.08220117 \\ 0.06710473 & 0.00171408 & 0.06633333 \\ 0.13197345 & 0.07936365 & 0.09334993 \end{pmatrix}$$

Which shows that group 1-1 communicating pattern is similar to the group 3-2, and the communication from 3-1 happened most often, also generally saying, that the row of μ_{3*} gives a highest value compared to the others. Which could probably represent that the individuals from group 3 are more likely to be talkative, and they are more frequently to chat with others.

For the purpose of finding the best number of clusters to partition the students, we fitted this College message dataset on both Poisson SBM model and the reciprocal Hawkes SBM model and tested on the number of latent clusters from 2 to 50. Log-likelihood is a useful metric to compare the models' performances. The log-likelihood is -55279 for Poisson stochastic block model and -542843 for Reciprocal Hawkes stochastic block model. The higher the values are compared to the selected models, the better the model is. The result shows that we can get a better capture of the behaviors between students' messages with the Reciprocal Hawkes stochastic block model. Area under the ROC Curve (AUC) is another performance metric which could measure the predicted power. From 0 to 1, the more closer to 1, the better the model is. We got 0.65 for Poisson stochastic block model, and 0.67 for Reciprocal Hawkes stochastic block model, and find out the cluster number after 10 the AUC will then fluctuate and not gain a higher value.

CHAPTER 6

Conclusion and extensions

In conclusion, this thesis has implemented the simple Hawkes process for one pair, stochastic block model based Poisson model, and the stochastic block model based reciprocal Hawkes model as well as research on the continuous relational network. We developed a better understanding of the individuals behavior and discovered the latent social structure. The generative process algorithm for each case was proposed and implemented. Several Markov Chain Monte Carlo sampling methods were involved to perform an efficient Bayesian posterior inference. The detailed joint likelihood function computation, posterior inference calculation and the highly vectorized implementation written in Python was presented.

We filled the gap between the inspired Blundell et al. paper [3], which proposed an infinite clusters reciprocal Hawkes model. He only briefly talked about using a Bayesian nonparametric method combined with slice sampling to perform the inferences. The inference we proposed with Gibbs sampling and optimal scaling Metropolis hastings gives an efficient and reliable result. Also the generative process for the mutually exciting Hawkes network is not available in the past literature, as they either skipped to test on the synthetic data step or they just specified that it was not implemented yet.

The data required for our model is simple, we only need the interaction data which has a sender, receiver and the time it occurred. We don't need to make further assumptions compared to some other literature which might need to fulfil several conditions. The possible applications of these models is huge. The Poisson stochastic block model can be used in modelling undirected connections between any pairs, the Reciprocal stochastic block model can capture reciprocal relationships and gives good modelling in directed linkages.

For the discussion on the fixed number of clusters, the flexible number of groups and a fixed number of collections is focused on different problem contexts. Both models can give a prediction on different cluster communication frequencies. For the context of the problems which is more clearly on what topic we are going to classify people and what communities we are interested in than the fixed number of communities as proposed in this thesis would be perfect. However, the flexible number of clusters could more suitable for the cases when we don't have an idea of how many communities we are interested in. Still, we are more interested in giving those nodes a community assignment. We initially planned to have a comparison between the Blundell et al. implement a flexible number of clusters and the fixed number of clusters as proposed in this thesis, however, we didn't

have time to implement the Hawkes-IRM to perform the comparison.

For the further extension with regards to the model, a possibility could be adding more information on background effect such as adding the propagating Dirichlet distribution [8] with the community effect taken into consideration. Also currently the reciprocity is considered on the base intensity. We can extend this into the triggering kernel to see how it performs.

These models are flexible enough to be applied on any relational time-series, not only limited to social network. There are more areas that can be explored, for example, on the medical image data. Currently another project I am on is to research on cognitive impairment for the elderly. In an anatomical view, the brain can be parsed into different Region of Interests (ROIs). The assumption is that the temporal similarity between the blood-oxygen-level-dependent (BOLD) signals in each region, as an indicator of synchronous neural activity, demonstrates functional connectivities in the human connectome. We are investigating those brain regions interaction networks which can be viewed as a relational network. We are trying to use the model this thesis's proposed model to apply it on Alzheimer's disease classification problems by finding out the latent groups of brain ROIs to see if some specific group appears to indicate the cognitive impairment in this way to improve the prediction accuracy.

CHAPTER 7

Appendix

There are a lot of coding involved for plotting, generative process, and the modelling program, and the execute function for synthetic data and real-world data. Below is some of my coding work, but there are 10 more files like this. Thus this will be just as a proof of the work and to avoid over-crowded of the Appendix.

```
from scipy.stats import dirichlet, gamma, poisson
from scipy.special import logsumexp

def generative_process(N, K, T):
    alphas = np.ones(K)*1
    pis = dirichlet.rvs(alpha = alphas)[0]
    zs = np.random.choice(a = K, p = pis, size = N)

    lambdas = gamma.rvs(a = 1, size = (K, K))
    #counts_mat counting the interaction
    counts_mat = np.zeros((N, N), dtype = int)

    for i in range(N):
        for j in range(N):
            if i != j:
                counts_mat[i, j] = poisson.rvs(mu = lambdas[zs[i], zs[j]])

    print("zs {}".format(zs))

    return alphas, pis, zs, lambdas, counts_mat

class PoissonSBM:
    def __init__(self, alphas, N, K, T, counts_mat):
        self.alphas = alphas
        self.N = N
        self.K = K
        self.T = T
        self.counts_mat = counts_mat

        self.pis = dirichlet.rvs(alpha = alphas)[0]
        self.zs = np.random.choice(a = K, p = self.pis, size = N)
        self.lambdas = gamma.rvs(a = 1, size = (K, K))
        pre_count = np.ones((self.N, self.N))
        np.fill_diagonal(pre_count, 0)
        self.pre_count = pre_count

        self.ll_val = []

    def sample_lambda(self):
        N_ij = np.zeros((self.K, self.K))
        m_k1k2 = np.zeros((self.K, self.K))

        for k1 in range(self.K):
            for k2 in range(self.K):
                N_ij[k1, k2] = np.sum(self.counts_mat[self.zs==k1][:, self.zs==k2])
                m_k1k2[k1, k2] = np.sum(self.pre_count[self.zs==k1][:, self.zs==k2])

        self.lambdas = gamma.rvs(a = N_ij+1, scale = 1)/(self.T*m_k1k2+1)

    def sample_z(self):
        for ii in range(self.N):
            ll_k = 0
            val1 = np.sum(self.lambdas[:, self.zs], axis=1)-self.lambdas[:, self.zs[ii]]
            val2 = np.sum(self.lambdas[self.zs, :], axis=0)-self.lambdas[self.zs[ii], :]
            ll_k = -self.T*(val1 + val2)

            val3 = np.sum(self.counts_mat[ii, :][np.newaxis, :]*np.log(self.lambdas[:, self.zs]), axis=1)
            val4 = np.sum(self.counts_mat[:, ii][:, np.newaxis]*np.log(self.lambdas[self.zs, :]), axis=0)

            ll_k = ll_k + val3+val4+np.log(self.pis)
            self.zs[ii] = np.random.choice(self.K, p = np.exp(ll_k-logsumexp(ll_k)))

    def sample_pi(self):
        m_counts = np.zeros(self.K)
        [unique_val, return_counts] = np.unique(self.zs, return_counts=True)
        m_counts[unique_val] = return_counts
        self.pis = dirichlet.rvs(alpha = 1 + m_counts)[0]
```

```

def cal_ll(self):
    lambda_ij = self.lambdas[self.zs][:, self.zs]
    v = -self.T*(np.sum(lambda_ij)-np.sum(np.diag(lambda_ij)))+np.sum(self.counts_mat*np.log(lambda_ij))
    print(v)
    self.ll_val.append(v)

data = (pd.read_csv("sx-mathoverflow.txt", sep=" ", header=None).values).astype(float)
NumEdge = 10000 #take first 10000 edges
# preprocess data
data = data[: NumEdge]
# not consider the interaction to myself
data = data[(data[:, 0] != data[:, 1])]
# unique_id = all sender and receiver id, sorted increasing
unique_id = np.unique(data[:, :2]) # np.sort(np.unique(data[:, :2]))
# data_id = first two column of data, id of sender and receiver
data_id = data[:, :2]
# change all the id to the range of 0-len(unique_id)
for ii in range(len(unique_id)):
    data_id[data_id == unique_id[ii]] = ii
# normalize time
data[:, 2] = (data[:, 2] / (24 * 3600.0))
data[:, 2] = data[:, 2] - np.min(data[:, 2])

KK = 10
dataNum = len(unique_id)

nodepair = data[:, :2].astype(int)
eventtime = data[:, 2]

index = np.argsort(eventtime)
eventtime = np.asarray(eventtime)[index]
nodepair = np.asarray(nodepair)[index]
alphas = np.ones(KK) * 1
counts_mat = np.zeros((dataNum, dataNum), dtype=int)

for p in range(len(nodepair)):
    counts_mat[nodepair[p][0]][nodepair[p][1]] += 1

PIRM = PoissonSBM(alphas, dataNum, KK, eventtime[-1], counts_mat)

# PIRM = PoissonIRM(alphas, N, K, T, counts_mat)

IterationTime = 1000

hist_lambda = []
hist_z = []

for ite in range(IterationTime):
    PIRM.sample_lambda()
    hist_lambda.append(PIRM.lambdas)
    PIRM.sample_z()
    hist_z.append(PIRM.zs)
    # hist_z.append([np.count_nonzero(PIRM.zs==0), np.count_nonzero(PIRM.zs==1), np.count_nonzero(PIRM.zs==2), np.count_nonzero(PIRM.zs==3)])
    PIRM.sample_pi()
    PIRM.cal_ll()
import numpy as np
import pandas as pd
import logging
import matplotlib.pyplot as plt
import seaborn as sns
import pymc3 as pm
from statsmodels.graphics.tsaplots import plot_acf

logging.basicConfig(level=logging.DEBUG, format='%(asctime)s - %(levelname)s - %(message)s')
logging.getLogger('matplotlib.font_manager').disabled = True
logging.debug('This is a log message.')

hyper_alpha_d = 1
hyper_beta_d = 1

hyper_alpha_y = 1
hyper_beta_y = 1

hyper_alpha_m = 1
hyper_beta_m = 1

class Model:
    def __init__(self, timepoints=[0.0]):
        # some initial values
        self.delta_vec = [2] # true val 3 2
        self.mu_vec = [0.5] # true val 0.5 .5
        self.y_vec = [1] # true val 0.2 .5*2
        self.time = timepoints
        self.child = [[] for i in range(len(timepoints))] # each time can have multiply childs

        # time differences matrix
        x = np.reshape(self.time, (len(self.time), 1))
        timediff = x - x.transpose()
        retDiff = timediff.transpose()
        retDiff = np.triu(retDiff, k=1)

```

```

        self.timeDiffMat = retDiff.transpose()

# branching structure A
# probability generated with base_intensity/total_intensity , triggering_kernel/total_intensity
def sample_A_Val(self):
    # initial value
    numBackgroundEvents = 0
    child_events = [[] for i in range(len(self.time))]
    background = self.mu_vec[-1]

    for i in range(0, len(self.time)):
        child_triggering_effect = self.y_vec[-1] * self.delta_vec[-1] * np.exp(- np.dot(self.delta_vec[-1], self.timeDiffMat[i][:i]))
        #child_triggering_effect = self.y_vec[-1]* np.exp(- np.dot(self.delta_vec[-1], self.timeDiffMat[i][:i]))

        un_normalized_prob = np.concatenate([background], child_triggering_effect)
        normalized_prob = un_normalized_prob / np.sum(un_normalized_prob)
        # this is the categorical value selected based on the probability to assign the parent

        val = np.random.choice(range(0, i+1), p=normalized_prob)
        if val == 0:
            # 0 can be seen as event generated from the base intensity
            numBackgroundEvents += 1
        else:
            # parent event append one more child event count
            # val-1 because value 0 means generated from base intensity , so when val = 1 is actually the t.0
            child_events[val-1].append(i)

    self.child = child_events
    return numBackgroundEvents

def sample_mu(self, numBackground):
    new_mu = np.random.gamma(shape=(hyper_alpha_m + numBackground), scale=1.0/(hyper_beta_m + max(self.time)))
    logging.debug("new_mu {}".format(new_mu))
    return new_mu

def sample_y(self):
    child_count = sum([len(c) for c in self.child])
    sum_cdf = sum([1.0-np.exp(-self.delta_vec[-1]*(self.time[-1]-t)) for t in self.time])
    logging.debug("sum_cdf {}".format(sum_cdf))
    logging.debug("Num of child = {} sum_cdf {} shape = {} scale = {}".format(child_count, sum_cdf, (hyper_alpha_y + child_count)))
    # new_y = np.random.gamma(shape=(hyper_alpha_y + child_count), scale = 1/((1/self.delta_vec[-1])*sum_cdf + hyper_beta_y))
    new_y = np.random.gamma(shape=(hyper_alpha_y + child_count), scale = 1/(sum_cdf + hyper_beta_y))
    logging.info("new_y {}".format(new_y))
    return new_y

def delta_target_log_likelihood(self, delta):
    sum_cdf = sum([1.0 - np.exp(-delta * (self.time[-1] - t)) for t in self.time])

    #second_cdf = sum([(self.y_vec[-1]) * (1 - np.exp(-delta * (self.time[-1] - t))) for t in self.time])
    logging.debug("second cdf {}".format(sum_cdf))
    cum_time_diff = 0 # t.k-t.m
    num_child = 0
    for k in range(len(self.time)):
        num_child_k = len(self.child[k])
        num_child += num_child_k
        if num_child_k > 0:
            # k th time point have a parent as p th time parent
            cum_time_diff += sum([self.time[self.child[k][m]]-self.time[k] for m in range(num_child_k)])

    #res = (num_child+hyper_alpha_d - 1) * np.log(delta) - delta * (cum_time_diff + hyper_beta_d)
    res = (num_child + hyper_alpha_d - 1) * np.log(delta) - delta * (cum_time_diff + hyper_beta_d + self.y_vec[-1]*sum_cdf)
    #logging.info("delta {}, cumTimeDiff {}, second exponential term {}".format(delta, cum_time_diff, second_cdf))
    return res

def sample_delta_metropolis(self, sigma):
    # Sample a candidate from Normal(mu, sigma) as Q is symmetric so hasting ratio is 1
    delta_current = self.delta_vec[-1]
    delta_proposed = np.random.normal(loc=delta_current, scale=sigma)

    if delta_proposed > self.y_vec[-1]:
        p = self.delta_target_log_likelihood(delta_proposed)-self.delta_target_log_likelihood(delta_current)

        u = np.random.rand()
        if np.log(u) <= p:
            logging.info("new_delta {} accepted".format(delta_proposed))
            # accept and make candidate the new sample
            self.delta_vec.append(delta_proposed)
        else:
            self.delta_vec.append(self.delta_vec[-1])

def simulation(self, nSim):
    for i in range(1, nSim):
        numBackground = self.sample_A_Val()

        self.mu_vec.append(0.5)
        #self.mu_vec.append(self.sample_mu(numBackground))

        self.y_vec.append(0.5)
        #self.y_vec.append(self.sample_y())

        #self.delta_vec.append(2)
        self.sample_delta_metropolis(2)

def plot_his(self, variable, name):
    # Density Plot and Histogram of all arrival delays

```

```

sns.distplot(variable, hist=True, kde=True, bins=180, color='darkblue',
              norm.hist = True, hist.kws={'edgecolor': 'black'},kde.kws={'linewidth': 4})

# Add labels
plt.title('Density plot and Histogram of the variable {}'.format(name))
plt.xlabel('{}'.format(name))
plt.ylabel('Density ')
plt.show()

if __name__ == '__main__':
    df = pd.read_csv("hawkes_time_points_by_lib_hawkes_15.52 alpha.csv", header=None)
    df.columns = ['time']
    time = np.array(df.get("time")).astype(np.float64)
    a = Model(timepoints=time)
    logging.info("start simulation")
    a.simulation(3000)
    print(a.delta_vec)
    print("mu mean {}".format(np.mean(a.mu_vec)))
    print(np.quantile(a.mu_vec, (0.025, 0.5, 0.975)))
    print("y mean {}".format(np.mean(a.y_vec)))
    print(np.quantile(a.y_vec, (0.025, 0.5, 0.975)))
    print("delta mean {}".format(np.mean(a.delta_vec)))
    print(np.quantile(a.delta_vec, (0.025, 0.5, 0.975)))
    a.plot_his(a.mu_vec, 'mu')
    a.plot_his(a.y_vec, 'y')
    a.plot_his(a.delta_vec[0::5], 'delta')
    plot_acf(np.array(a.delta_vec[0::10]))
    pm.traceplot(np.array(a.mu_vec))
    pm.traceplot(np.array(a.y_vec))
    pm.traceplot(np.array(a.delta_vec[0::5]))
    plt.show()

```

References

- [1] E. M. Airoldi, T. B. Costa, and S. H. Chan. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. page 9.
- [2] D. Blei, P. Frazier, and I. Mukherjee. COS 597C: Bayesian nonparametrics. page 7.
- [3] C. Blundell, J. Beck, and K. A. Heller. Modelling Reciprocating Relationships with Hawkes Processes. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems* 25, pages 2600–2608. Curran Associates, Inc., 2012. URL <http://papers.nips.cc/paper/4834-modelling-reciprocating-relationships-wit>
- [4] D. R. Cox. Some Statistical Methods Connected with Series of Events. *Journal of the Royal Statistical Society. Series B (Methodological)*, 17(2):129–164, 1955. ISSN 0035-9246. URL <https://www.jstor.org/stable/2983950>. Publisher: [Royal Statistical Society, Wiley].
- [5] D. J. Daley and D. Vere-Jones. *An introduction to the theory of point processes*. Springer, New York, 2nd ed edition, 2003. ISBN 978-0-387-95541-4 978-0-387-21337-8 978-0-387-49835-5.
- [6] C. DuBois, C. T. Butts, and P. Smyth. Stochastic blockmodeling of relational event dynamics. page 9.
- [7] R. Durrett. Essentials of Stochastic Processes. page 226.
- [8] X. Fan, B. Li, C. Li, S. Sisson, and L. Chen. Scalable Deep Generative Relational Model with High-Order Node Dependence. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems* 32, pages 12658–12668. Curran Associates, Inc., 2019. URL <http://papers.nips.cc/paper/9428-scalable-deep-generative-relational-model>
- [9] W. Fu, L. Song, and E. P. Xing. Dynamic mixed membership block-model for evolving networks. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pages 329–336, New York, NY, USA, June 2009. Association for Computing Machinery. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553416. URL <https://doi.org/10.1145/1553374.1553416>.
- [10] D. Garlaschelli and M. I. Loffredo. Patterns of Link Reciprocity in Directed Networks. *Physical Review Letters*, 93(26):268701, Dec. 2004. doi: 10.1103/PhysRevLett.93.268701. URL <https://link.aps.org/doi/10.1103/PhysRevLett.93.268701>. Publisher: American Physical Society.
- [11] P. H. Garthwaite, Y. Fan, and S. A. Sisson. Adaptive Optimal Scaling of Metropolis-Hastings Algorithms Using the Robbins-Monro Process. *arXiv:1006.3690 [stat]*, June 2010. URL <http://arxiv.org/abs/1006.3690>. arXiv: 1006.3690.

- [12] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, Nov. 1984. ISSN 1939-3539. doi: 10.1109/TPAMI.1984.4767596. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.
- [13] S. Hanneke and E. P. Xing. Discrete Temporal Models of Social Networks. In E. Airoldi, D. M. Blei, S. E. Fienberg, A. Goldenberg, E. P. Xing, and A. X. Zheng, editors, *Statistical Network Analysis: Models, Issues, and New Directions*, Lecture Notes in Computer Science, pages 115–125, Berlin, Heidelberg, 2007. Springer. ISBN 978-3-540-73133-7. doi: 10.1007/978-3-540-73133-7₉.
- [14] S. S. Hansen, J. K. Lee, and S.-Y. Lee. CONSUMER-GENERATED ADS ON YOUTUBE: IMPACTS OF SOURCE CREDIBILITY AND NEED FOR COGNITION ON ATTITUDES, INTERACTIVE BEHAVIORS, AND EWOM. 15(3):13, 2014.
- [15] W. K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, 1970. ISSN 0006-3444. doi: 10.2307/2334940. URL <https://www.jstor.org/stable/2334940>. Publisher: [Oxford University Press, Biometrika Trust].
- [16] P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic block-models: First steps. *Social Networks*, 5(2):109–137, June 1983. ISSN 03788733. doi: 10.1016/0378-8733(83)90021-7. URL <https://linkinghub.elsevier.com/retrieve/pii/0378873383900217>.
- [17] J. P. Huelsenbeck. Bayesian Inference of Phylogeny and Its Impact on Evolutionary Biology. *Science*, 294(5550):2310–2314, Dec. 2001. ISSN 00368075, 10959203. doi: 10.1126/science.1065889. URL <https://www.sciencemag.org/lookup/doi/10.1126/science.1065889>.
- [18] D. Kaplan. *Bayesian statistics for the social sciences*. Bayesian statistics for the social sciences. Guilford Press, New York, NY, US, 2014. ISBN 978-1-4625-1651-3 978-1-4625-1667-4. Pages: xviii, 318.
- [19] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning Systems of Concepts with an Infinite Relational Model. page 8.
- [20] J. Kruschke and H. Joo. The Time Has Come Bayesian Methods for Data Analysis in the Organizational Sciences. *Organizational Research Methods*, 15:722–752, Oct. 2012. doi: 10.1177/1094428112457829.
- [21] B. Lawrence, S. Fournier, and F. Brunel. When Companies Don’t Make the Ad: A Multimethod Inquiry Into the Differential Effectiveness of Consumer-Generated Advertising. *Journal of Advertising*, 42(4):292–307, Oct. 2013. ISSN 0091-3367. doi: 10.1080/00913367.2013.795120. URL <https://doi.org/10.1080/00913367.2013.795120>. Publisher: Routledge _eprint: <https://doi.org/10.1080/00913367.2013.795120>.
- [22] Y. Lee. Hawkes Processes with Stochastic Excitations - Supplemental Materials. page 4.
- [23] P. A. W. Lewis. Simulation of Nonhomogeneous Poisson Processes by Thinning. page 44.
- [24] D. Marsan and O. Lengliné. Extending Earthquakes’ Reach Through Cascading. *Science*, 319(5866):1076–1079, Feb. 2008. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1148783. URL <https://science.sciencemag.org/content/319/5866/1076>. Publisher: American Association for the Advancement of Science Section: Report.

- [25] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, Feb. 2004. ISSN 1539-3755, 1550-2376. doi: 10.1103/PhysRevE.69.026113. URL <http://arxiv.org/abs/cond-mat/0308217>. arXiv: cond-mat/0308217.
- [26] K. Nowicki and T. A. B. Snijders. Estimation and Prediction for Stochastic Block-structures. *Journal of the American Statistical Association*, 96(455):1077–1087, Sept. 2001. ISSN 0162-1459, 1537-274X. doi: 10.1198/016214501753208735. URL <http://www.tandfonline.com/doi/abs/10.1198/016214501753208735>.
- [27] Y. Ogata. On Lewis’ simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31, Jan. 1981. ISSN 1557-9654. doi: 10.1109/TIT.1981.1056305. Conference Name: IEEE Transactions on Information Theory.
- [28] P. Panzarasa, T. Opsahl, and K. Carley. Patterns and Dynamics of Users’ Behavior and Interaction: Network Analysis of an Online Community. *JASIST*, 60:911–932, May 2009. doi: 10.1002/asi.21015.
- [29] J. G. Rasmussen. Bayesian Inference for Hawkes Processes. *Methodology and Computing in Applied Probability*, 15(3):623–642, Sept. 2013. ISSN 1387-5841, 1573-7713. doi: 10.1007/s11009-011-9272-5. URL <http://link.springer.com/10.1007/s11009-011-9272-5>.
- [30] M.-A. Rizoiu, Y. Lee, S. Mishra, and L. Xie. A Tutorial on Hawkes Processes for Events in Social Media. *arXiv:1708.06401 [cs, stat]*, Oct. 2017. URL <http://arxiv.org/abs/1708.06401>. arXiv: 1708.06401.
- [31] C. Robert and G. Casella. A Short History of Markov Chain Monte Carlo: Subjective Recollections from Incomplete Data. *Statistical Science*, 26(1):102–115, Feb. 2011. ISSN 0883-4237. doi: 10.1214/10-STS351. URL <http://arxiv.org/abs/0808.2902>. arXiv: 0808.2902.
- [32] P. E. Rossi and G. M. Allenby. Bayesian Statistics and Marketing. *Marketing Science*, 22(3):304–328, 2003. ISSN 0732-2399. URL <https://www.jstor.org/stable/4129743>. Publisher: INFORMS.
- [33] P. Sarkar and A. W. Moore. Dynamic Social Network Analysis using Latent Space Models. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1145–1152. MIT Press, 2006. URL <http://papers.nips.cc/paper/2879-dynamic-social-network-analysis-using-latent->
- [34] K. Sigman. Stationary Marked Point Processes. In H. Pham, editor, *Springer Handbook of Engineering Statistics*, pages 137–152. Springer London, London, 2006. ISBN 978-1-84628-288-1. doi: 10.1007/978-1-84628-288-1₈.URL