Variational Bayesian Inference of Point Processes for Time-Sequence Modeling
（時系列モデリングのための点過程の変分ベイズ推論）

by

Hongyi Ding
丁　弘毅

A Doctor Thesis
博士論文

Submitted to
the Graduate School of the University of Tokyo
on December 7, 2018
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Information Science and Technology
in Computer Science

Thesis Supervisor: Masashi　Sugiyama　杉山　将
Professor of Computer Science

ABSTRACT

A time-sequence consists of a set of time-stamps, each of which records the arrival time of an event. Time-sequence data can generally be classified into two types. One is from experiments that monitor subjects in a continuous fashion; and thereby the exact timestamps of all occurrences of the events are fully observable. These data are usually referred to as *recurrent event data*. On the other hand, we have the so-called *panel count data*, in which only the numbers of occurrences of the events between subsequent observation times. In real-world problems arising in areas such as social science, health care and crime prevention, time-sequence modeling is extremely useful since it can help us in predicting future events and understanding the reasons behind them.

A common approach to time-sequence modeling is to assume a time-sequence is generated by a temporal point process. Cox processes are widely used in the models of temporal point processes. A Cox process is defined via a stochastic intensity function. The stochastic process to generate the intensity function is usually chosen to be a Gaussian process (GP) and the model using a GP is called a *Gaussian-process-modulated Poisson process* (GP3) model. For the recurrent event data, GP3 models have been studied extensively. Among all approaches which try to solve the inference problem, the variational inference method provides a computationally efficient estimate of the intensity function and does not require a careful discretization of the underlying space.

In order to retain the scalability and computation efficiency of the variational inference approach and model the uncertainty of the intensity function when we only observe panel count data, we present the first Bayesian inference framework for panel count data. We assume that all time-sequences in the data set share the same intensity function, which is generated by a GP3 model. The method of conducting computationally efficient variational inference is presented. We derive a tractable lower bound to alleviate the problem of the intractable evidence lower bound inherent in the variational inference framework. Our model, the *Gaussian-process-modulated Poisson process for panel count data* (GP4C), outperforms a non-Bayesian method in terms of the test likelihood and achieves comparable results in computation time.

For multiple time-sequences, it is often cumbersome to assume all time-sequences share the same intensity function since we may overlook the variety for different time-sequences. A key idea to model the heterogeneity is to cluster the data into groups while allowing the groups to remain linked to share the latent functions. Several models have been proposed on the basis of this simple idea, e.g., the convolution process, nonnegative matrix factorization (NMF), and latent Poisson process allocation (LPPA). These models employ latent factors to share statistical strengths and combine these functions to model the correlations within and among time-sequences. Among these models, LPPA is a powerful approach because it uses latent functions obtained from a GP, which is a flexible prior for a random function. However, a limitation of LPPA is that the number of latent functions needs to be set beforehand. If the chosen number is much larger than the actual number of latent functions required to explain the data, LPPA will still use all the latent functions and over-fit on the training data set.

To automatically infer the number of basis functions for multiple time-sequences, we present the *Bayesian nonparametric Poisson process allocation* (BaNPPA), a latent-function model for time-sequences. We model the intensity of each sequence as an infinite mixture of latent functions, each of which is obtained using a function drawn from a GP. We show that a technical challenge for the inference of such mixture models is the un-identifiability of the weights of the latent functions. We propose to cope with the issue by regulating the volume of each latent function within a variational inference algorithm. Our algorithm is computationally efficient and scales well to large data sets. We demonstrate the usefulness of our proposed model through experiments on both synthetic and real-world data sets.

In summary, we proposed two computationally efficient variational Bayesian inference algorithms for time-sequence modeling. In the first algorithm GP4C, we quantified the average arrival rate for multiple time-sequences and provided the additional uncertainty, which helps illustrate the difficulty of the prediction. For the second algorithm BaNPPA, we automatically inferred the number of basis functions to model the variety for multiple

time-sequences, which could provide insights into the understanding of social networks and human activities.

Acknowledgements

Pursuing a Ph.D. degree in the University of Tokyo is absolutely an unexpected journey for me. I will cherish every moment of happiness or sorrow during this journey since they have become the most valuable treasure of my life.

At this moment when I am about to finish my Ph.D. study, I would like to thank Prof. Masashi Sugiyama, who is the first and by far the most influential mentor during my journey. Prof. Masashi Sugiyama has never explicitly shown me any specific research direction. However, he taught me something much more important, that is, to think independently and act on my own initiative. Moreover, he paid consistent attention on my progress. Whenever I went to a dead end, he would always point it out and provide me with insightful suggestions to help me get back to the main avenue. Above all, he has created an ideal environment for us machine learning researchers in the University of Tokyo, where we could freely discuss every topic in any field and pursue anything we want.

Next, I would like to express my special thanks of gratitude to Dr. Young Lee, Prof. Issei Sato and Prof. Mohammad Emtiyaz Khan. I greatly appreciate their advices and I learned a lot from cooperating with them, for they have provided me with numerous valuable paper writing skills as well as research methods. Without them, I would never have accomplished my Ph.D. research.

Furthermore, my gratitude goes out to all the members of Sugiyama-Sato-Honda Laboratory. Especially, I am grateful to Prof. Hiroaki Sasaki, Prof. Hideko Kawakubo, Dr. Gang Niu, Dr. Yao Ma, Dr. Hao Zhang, Dr. Kishan Wimalawarne, Dr. Kiyoshi Irie, Dr. Voot Tangkaratt, Nontawat Charoenphakdee, Nan Lu, Futoshi Futami, Jie Luo, Masayoshi Hayashi, Yusuke Konno, Chenri Ni, Jiangqi Dong, Chenri Ni, Masahiro Kato, Han Bao, Tianyi Zhang, Jongyeong Lee, Zeke Xie and Zhenghang Cui. The secretary Ms. Yuko Kawashima kindly helped me with the Japanese documents. I deeply appreciate her help.

My research project is financially supported by the University of Tokyo Foreign Students Special Scholarship Program. The financial support helped me in every aspect: getting used to the new environment, focusing on my research and attending workshops and conferences. I would like to thank the University of Tokyo. Without the scholarship, my life in Japan would have been much harder.

Last but not least, I would like to thank my parents who have always been by my side and supported me during the darkest moment in my Ph.D. studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Internet age has made it possible to collect a huge amount of temporal data available in the form of time-sequences. Each time-sequence consists of time-stamps which record the arrival times of events, e.g., postings of tweets on Twitter or announcements of life events on Facebook. In real-world problems arising in areas such as social science [32], health care [57] and crime prevention [59], time-sequence modeling is extremely useful since it can help us in predicting future events and understanding the reasons behind them. This thesis is devoted to the application of the variational inference method, one branch of the Bayesian inference methods in machine learning, on modeling the time-sequence data.

1.1 Machine Learning

Machine learning is a natural outgrowth in the intersection of computer science and statistics, which seeks to answer the following questions [67]:

"*How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?*"

While statistics primarily discusses how to collect, store, analyze and present the data mathematically [62] and computer science has mainly focused on how to manually program computers, machine learning considers both what computational structures and algorithms can be utilized for computationally-efficient statistical analysis and how to let the computers learn by themselves [67].

1.1.1 Machine Learning and Human Learning

When talking about machine learning, one may wonder what the difference between human learning and machine learning is and whether it is really necessary to teach machines to learn.

There are two distinctive features about the human learning from machine learning [64]. First of all, the human learning process is incredibly slow. Generally speaking, it takes a human being nearly thirty years to learn to become a systematist (a specialist in taxonomy) from a baby in the cradle. On the other hand, machine learning models can be trained much more efficiently provided with sufficient resources. Recently, researchers from Sony have trained a popular machine learning model (the ResNet-50 neural network model) on ImageNet, a 1,000-class image classification data set, in 224 seconds and achieved an accuracy of 75.03% [65]. Secondly, the learning process of the human can not be copied. Each human being has to struggle to learn by himself/herself and repeat the

same learning process from generation to generation. In contrast, once you have a debugged machine learning program, it can be easily copied as many times as you want and can be transplanted from one computer to another with almost no effort.

Therefore, it is desirable to develop machine learning programs to avoid the inefficiencies of human being. Needless to say, robots and advanced automation aided by machine learning have eased the burden on human beings who are required to work in tedious assembly lines or dangerous coal mines. Moreover, the development of machine learning can potentially speed up the human learning process. For example, machine learning techniques were utilized in designing effective instructional systems which aim at providing personalized interactions to an individual learner [37].

We should also notice that there is a rather unsettling open question on whether machine will replace human beings completely in the future. Over the past decades, we have witnessed the superiority of machines over human experts in more and more domains. Taking the game as an example, machine learning programs beat human champions in the game of chess in 1997 [14] and Go in 2016 [87]. For a thorough discussion about the influence of machine learning on the employment rate, the readers are referred to Rifkin [79] and the references thereafter.

1.1.2 Types of Machine Learning

Depending on the differences in the problem settings, common machine learning algorithms can usually be classified into the three main categories: supervised learning, unsupervised learning and reinforcement learning [70].

- *Supervised learning*: The goal is to learn a mapping from inputs to outputs given a data set containing input and output pairs. The training data set is usually presented in the following form:

  $$\{(x_1, y_1), \ldots, (x_N, y_N)\},$$

  where N is the number of training examples, $\{x_i\}$ are inputs called features or covariates and $\{y_i\}$ are the outputs or labels given by a human expert. When each y is a categorical variable from a given finite set, e.g., $y = \{1, 2, \ldots, C\}$, the problem is usually known as classification. When y is a real number, the problem is called regression [71]. Supervised learning plays an important role in applications such as face recognition and object detection.

- *Unsupervised learning*: The goal here is to find "interesting or meaningful" patterns from only the input data in the following form:

  $$\{x_1, \ldots, x_N\}.$$

  This type of learning is also referred to as knowledge discovery. Unlike supervised learning, we are not provided with the desired output and therefore there is no obvious error metric to use. However, unsupervised learning is argued to be more applicable than supervised learning since the human expert is not required in this scenario [70]. Typical examples of unsupervised learning are clustering and latent factor analysis [71].

- *Reinforcement learning*: The task is concerned with learning how to act or behave in an unknown environment so as to maximize the cumulative rewards [93]. The learning algorithm is presented with occasional reward or punishment signals. Reinforcement learning has been successfully applied to various problems, such as automatic driving and robot control. More recently, a reinforcement learning algorithm, AlphaGo, has defeat a world champion in the game of Go [87].

The studies of this thesis fall into the category of unsupervised learning as we are mainly concerned with discovering meaningful patterns from the unlabeled time-sequence data.

1.2 Bayesian Statistics

Bayesian statistics is a theory in statistics on the basis of the Bayesian interpretation of probability in which probability represents the degree of belief in an event and the belief can change as new information is gathered [1]. In this section, we will briefly discuss the difference between Bayesian statistics and frequentist statistics, the most widely-applied statistics in machine learning and then introduce two inference methods in Bayesian statistics.

1.2.1 Bayesian Statistics and Frequentist Statistics

There has been a debate among statisticians for nearly a century over the issue of whether the Bayesian or frequentist paradigm is superior. The debate is still ongoing, since these two paradigms do not share the same philosophical and pedagogical foundation. However, methodologically there is an agreement that both approaches contribute to the statistical practice to a great extent and each is indispensable for full development of the other approach [5].

Let $\mathcal{D}$ and x be the data and the parameter. We now briefly discuss some of the basic assumptions in Bayeisan and frequentist statistics.

- In Bayesian statistics [13], $\mathcal{D}$ is viewed as fixed after the data generation process and x is treated as a random variable. According to Bayes' theorem, we have the posterior probability of x:

$$P(x|\mathcal{D}) = \frac{P(\mathcal{D}|x)P(x)}{P(\mathcal{D})}, \tag{1.1}$$

  where $P(x)$ is the prior representing the initial degree of belief about the variable and $P(x|\mathcal{D})$ is the posterior representing the degree of belief after seeing the data set $\mathcal{D}$. The uncertainty of the parameter can be directly obtained by computing the posterior distribution $P(x|\mathcal{D})$.

- In frequentist statistics, the parameter x is viewed as fixed and the data $\mathcal{D}$ are still treated as random after the data generation process. This setting is opposite to Bayesian statistics [70]. An estimate of the parameter x is conducted by applying an estimator δ to the data:

$$\hat{x} = \delta(\mathcal{D}).$$

  There is no automatic way of deriving an optimal estimator δ and we are free to choose any estimator δ as we want. The uncertainty about the

[1] https://deepai.org/machine-learning-glossary-and-terms/bayesian-statistics

parameter estimate $\hat{x}$ can be measured by calculating the sampling distribution of the estimator. The sampling distribution can be obtained by sampling many different data sets from the true model. The bootstrap [26] is one commonly-used Monte Carlo technique which can approximate the sampling distribution. The posterior distribution in Bayesian statistics and the sampling distribution by the bootstrap in frequentist statistics are quite similar [30]. However, the bootstrap is not that direct since we need to sample the data set multiple times [70].

1.2.2 Variational Inference and Markov Chain Monte Carlo Inference

One fundamental problem in the modern statistics is approximating the probability densities which are difficult to compute [12]. This problem arises naturally in Bayesian inference. Bayesian inference [13] is a method of Bayesian statistical inference where Bayes' theorem is utilized to update the posterior probability of a hypothesis as more evidence or data become available. More specifically, in Equation (1.1) the posterior probability $P(x|\mathcal{D})$ sometimes can not be analytically computed and has to be approximated.

Variational Bayesian inference [48] is one method to approximate the posterior probability in a Bayesian model. One alternative and competitive strategy of this task is Markov Chain Monte Carlo (MCMC) inference.

- *MCMC inference* [80, 63]: The basic idea in MCMC inference is to construct a Markov chain on the state space whose stationary distribution is the target posterior distribution $P(x|\mathcal{D})$. That is, the random walk generated by the Markov chain visits any possible state x with the frequency proportional to $P(x|\mathcal{D})$. This inference method can provide asymptotically exact samples of the target posterior probability [80]. However, the computation cost of MCMC inference is rather intensive since there might be a large amount of rejected samples if the proposal distribution is not chosen properly and a short mixing time is usually hard to obtain [33].

- *Variational inference* [48]: The basic idea in the variational inference is to pick an approximation distribution $q(x)$ from some tractable distribution family and to try to make this approximation as close as possible to the true posterior distribution $P(x|\mathcal{D})$ [70]. Then the inference problem is reduced to an optimization problem. The variational inference method does not have the theoretical guarantee as the MCMC inference and there is possibly a model bias since the tractable distribution family may not contain the true posterior distribution. However, variational inference enjoys a much faster convergence rate than MCMC inference and can be easily adapted to very large data sets [44].

Our studies in this thesis fall into the category of variational inference in Bayesian statistics. The reason is that we would like to obtain a direct and computationally-efficient estimate of the uncertainty of the underlying parameters. We should note that the studies of time-sequence data with MCMC inference or in the frequentist paradigm are equally intriguing and worth further research effort.

1.3 Time-Sequence Data

A time-sequence consists of a set of time-stamps, each of which records the arrival time of an event. Time-sequence data can generally be classified into two types:

the recurrent event data and the panel count data [91]. In this section, we briefly introduce these two types of data which we are going to study in this thesis.

1.3.1 Recurrent Event Data

The first type of time-sequence data arise from experiments that monitor subjects in a continuous fashion; and thereby the exact timestamps of all occurrences of the events are fully observable. These data are usually referred to as *recurrent event data* [15].

As a preliminary example of the recurrent event data, we introduce the coal-mining disaster data set [46]. This data set records the time intervals between successive coal-mining explosions involving 10 or more men killed from 1851 to 1962 in Britain. The entire data set is shown in Table 1.1. We notice that it is difficult to examine the arrival rate of events by directly looking at the numbers in Table 1.1.

Table 1.1: **Coal-mining disaster data set**. This data set records the time intervals in days between successive explosions in mines, from 15th March 1851 to 22 March 1962. The numbers in the table are listed column-wisely.

157	65	53	93	127	176	22	1205	1643	312
123	186	17	24	218	55	61	644	54	536
2	23	538	91	2	93	78	467	326	145
124	92	187	143	0	59	99	871	1312	75
12	197	34	16	378	315	326	48	348	364
4	431	101	27	36	59	275	123	745	37
10	16	41	144	15	61	54	456	217	19
216	154	139	45	31	1	217	498	120	156
80	95	42	6	215	13	113	49	275	47
12	25	1	208	11	189	32	131	20	129
33	19	250	29	137	345	388	182	66	1630
66	78	80	112	4	20	151	255	292	29
232	202	3	43	15	81	361	194	4	217
826	36	324	193	72	286	312	224	368	7
40	110	56	134	96	114	354	566	307	18
12	276	31	420	124	108	307	462	336	1358
29	16	96	95	50	188	275	228	19	2366
190	88	70	125	120	233	78	806	329	952
97	225	41	34	203	28	17	517	330	632

One popular approach to modeling and visualizing the variation of the arrival rate of events is via the intensity function in the inhomogeneous Poisson process [52, 15]. In this thesis, we restrict ourselves to the recurrent event data in which the arrival rate of events varies smoothly over time. For point processes which allow "spiky" patterns and the arrival rate can be non-smooth functions, the readers are referred to the Hawkes processes [39].

The traditional point-estimate approach to modeling the smoothly-varying intensity function is based on the smoothing kernels [18]. Diggle [18] proposed to utilize Rosenblatt's density kernel estimate [81] on the intensity estimation problem and to optimize the bandwidth of the kernel function via the empirical

Ripley's function. The local likelihood method [103, 9, 43] can be seen as the generalization of the kernel intensity estimate [18]. When using a zero-order polynomial approximation in the local likelihood, the estimate is reduced to the kernel smoothing estimate. However, when using a higher-order approximation, we can only obtain the estimate of the intensity value at a given time. More recently, Flaxman et al. [28] exploited the properties of the reproducing Hilbert space to estimate the intensity function within the empirical risk minimization framework.

Another branch of the intensity estimation methods is the Bayesian estimate. Within the Bayesian framework, the prior of the intensity function is constructed by passing a random function drawn from a Gaussian process, through a proper transformation. This type of inhomogeneous Poisson processes is also known as *Gaussian-process modulated Poisson processes* [60]. However, the Bayesian inference of the intensity function is intractable since we need to integrate an infinite dimensional random function over the domain of the intensity function [2]. Various approximation methods have been proposed to perform the tractable inference, including the MCMC sampling [2, 84], the Laplace approximation method [98] and the variational inference method [60, 47, 22].

A closely-related research direction circumvented the intractable inference problem by introducing a computational grid [20] to discretize the domain of the intensity function [68, 66, 29].

1.3.2 Panel Count Data

The second type of time-sequence data is due to the lack of supervision over the recurrent events and only the numbers of the events between subsequent observation times are recorded. This type of data is commonly referred to as *panel count data* [91]. For example, it is sometimes too expensive for a patient to pay for the continuous follow-up observation in the hospital after the treatment. He/She is then allowed to go home after a certain treatment and is required to go back to the hospital and report the symptoms. Therefore, only the numbers of symptoms between subsequent visits are recorded, such as the number of vomits or new tumors.

An illustrative example to show the difference between the recurrent event data and the panel count data is given in Figure 1.1. In the example shown in Figure 1.1, the patient vomited three times after the treatment. In the recurrent event data, we have access to the exact time-stamp when the patient vomited and the time-stamps are 60 minutes (1 hour), 121 minutes (2 hours 1 minutes) and 190 minutes (3 hours and 10 minutes). However, in the case of panel count data, the exact time-stamps of the vomits are no longer available. The patient visited the hospital twice after the treatment and only the numbers of vomits between subsequent visits are recorded.

Several maximum likelihood point-estimates have been proposed on the basis of the likelihood of the panel count data or its variants. Wellner and Zhang [100] proposed the nonparametric maximum likelihood estimator (NPMLE), assuming that the underlying intensity function is a positive piece-wise constant function. Based on NPMLE, Zhang and Jamshidian [105] added an additional Gamma-distributed random variable to the intensity function to model the individual effect among multiple time-sequences. Similar to the recurrent event data, the local likelihood method has also been exploited by Betensky et al. [9] and Fan et al. [25].

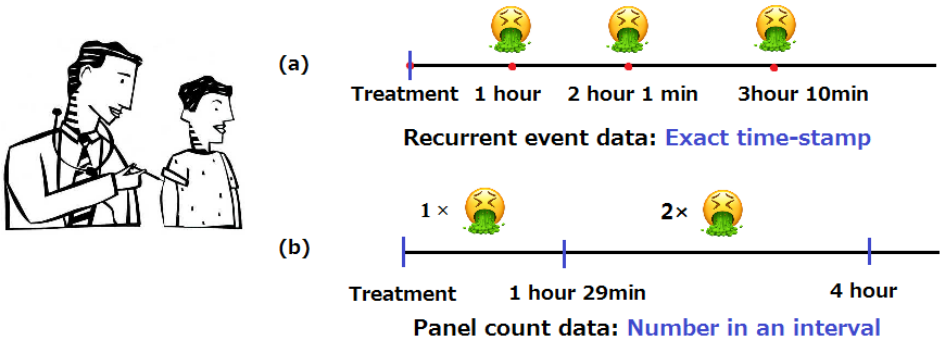

Figure 1.1: The time-sequence of the vomits after the treatment. (a) **Recurrent event data**. The exact time-stamps of three times of vomits can be obtained. (b) **Panel count data**. The patient reported that in two intervals he/she vomited once and twice respectively.

1.4 Contributions

This thesis is devoted to the application of the variational inference method, one branch of the Bayesian inference methods in machine learning, on modeling the time-sequence data. The contributions in this thesis are mainly on Chapters 3 and 4. We briefly introduce the contributions below.

1.4.1 Chapter 3: Variational Inference for Panel Count Data with Gaussian Processes

Time-sequence data can be generally divided into the recurrent event data and panel count data [91]. This chapter focuses on modeling multiple time-sequences in the form of panel counts. The technical contributions of this chapter are three-fold.

1. In the first place it undertakes to construct a variational inference procedure for the Gaussian-process-modulated Poisson process model for panel count data (GP4C).

2. To carry out a variational inference in this setting, we derive a simple and tractable lower bound of the intractable evidence lower bound and demonstrate through empirical evidence that with this lower bound, GP4C outperforms a non-Bayesian method.

3. To model the diversity among multiple time-sequences, we proposed the Gaussian-process-modulated Poisson process model for panel count data with individual weight (GP4CW) model. Experiments show that this model further improves the performance of test likelihood.

Generally speaking, this chapter presents two useful Bayesian time-sequence modeling methods, GP4C and GP4CW. These models serve as an alternative to the current mainstream point-estimates for the machine learning researchers and practitioners who are interested in modeling and understanding panel count data.

1.4.2 Chapter 4: Bayesian Nonparametric Poisson Process Allocation

This chapter focuses modeling the diversity of time-sequence data. We choose recurrent event data rather than panel count data because we could easily have access to massive recurrent event data. The technical contributions of this chapter are two-fold.

1. In the first place, we present a scalable and accurate Bayesian nonparametric approach for time-sequence modeling, that is, Bayesian Nonparametric Poisson Process Allocation (BaNPPA).

2. We propose a computationally efficient variational inference algorithm for BaNPPA and solve the un-identifiability issue by adding a constraint within the inference algorithm to regulate the volume of each latent function.

Generally speaking, this chapter presents the challenges and possible solutions when applying Bayesian nonparametric techniques on multiple time-sequences. This discussion might provide insights on future Bayesian nonparametric researches. For medical practitioners, BaNPPA can automatically identify different patterns of symptoms and help develop individual treatments for each patient.

1.5 Organization

The thesis consists of five chapters and an illustrative flow chart is shown in Figure 1.2. In this section, we will introduce the organization of each chapter.

In Chapter 2, we give the preliminaries of this thesis. The general notations as well as the notations on the time-sequence data are listed in Section 2.1. In Section 2.2, we introduce some of the basic concepts related to stochastic processes. Several stochastic processes, including temporal point processes, Gaussian processes and stick-breaking processes are briefly discussed in Section 2.3, 2.4 and 2.5, respectively. We review the intensity estimation for the recurrent event data in Section 2.6 and the intensity estimation for the panel count data in Section 2.7.

In Chapter 3, we present the first Bayesian inference framework for Gaussian process-modulated Poisson processes when the temporal data appear in the form of panel counts. In Section 3.1 and 3.2, we introduce and discuss the background of this research. In Section 3.3, we provide the variational inference framework for panel count data with Gaussian processes. We also briefly discuss how to model the diversity among multiple time-sequences with the GP4CW model in Section 3.4. The experiments on GP4C and GP4CW are shown in Section 3.5. Section 3.6 contains the supplementary materials for the proof of one lemma in the Section 3.3.

In Chapter 4, we focus on how to model the diversity among multiple time-sequences with latent functions. As a beginning, we study the case of recurrent event data. In Section 4.1 and 4.2, we introduce and discuss the problem of factor analysis for time-sequence data. In Section 4.3, we provide the BaNPPA model. Then the variational inference for the BaNPPA model can be found in Section 4.4. Experiments of the BaNPPA model is then shown in Section 4.5. Section 4.6 contains the supplementary materials for the derivations related to the evidence lower bound (ELBO).

Finally in Chapter 5, we conclude this thesis and present several directions for future work.

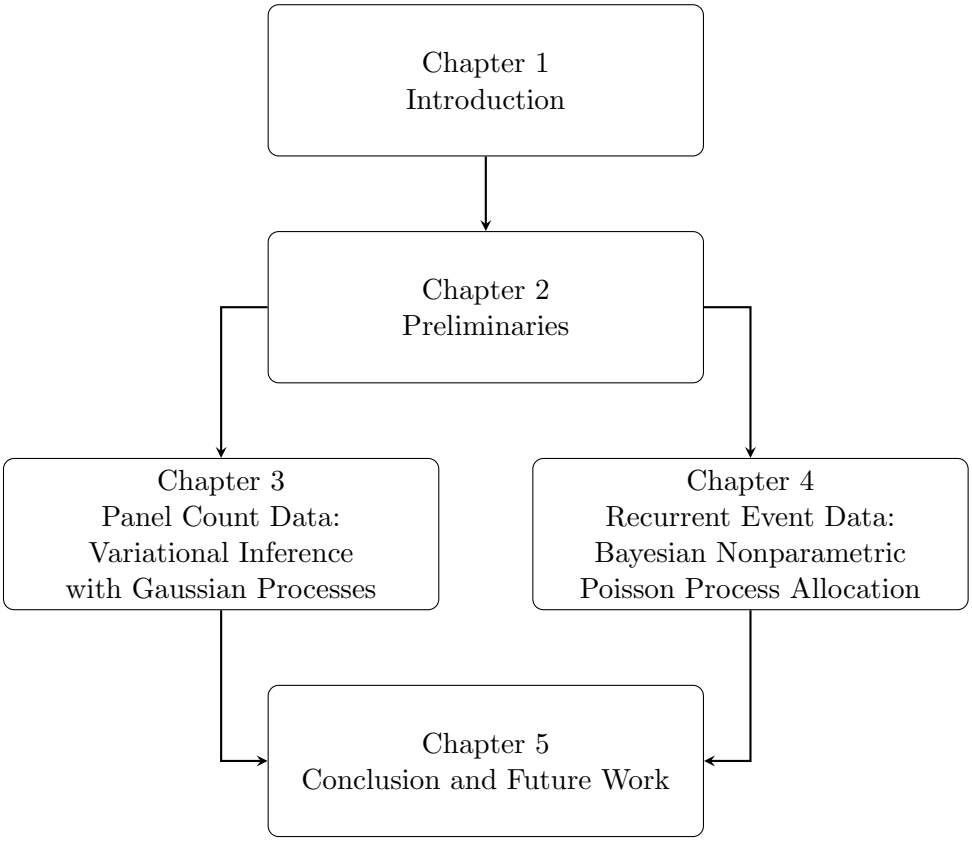

Figure 1.2: Structure of the thesis. The contributions in this thesis are mainly on Chapters 3 and 4.

Chapter 2

Preliminaries and Previous Work

In this chapter, we will introduce the notations used in this thesis. A brief introduction of several stochastic processes, mainly on temporal point processes, Gaussian processes and stick-breaking processes, will also be provided. Finally we will review the previous studies in the intensity estimation for both recurrent event data and panel count data.

2.1 Notations

In this section, we will introduce the general notations used in this thesis as well as the notations for the time-sequence data.

2.1.1 General Notations

We denote a random variable with an uppercase letter, such as X, Y, or Z. A scalar or an experimental observation of a random variable is denoted with a lowercase letter x, y, or z. We denote the distribution of a discrete random variable or the probability of an event with $P(\cdot)$ and the probability density function of a continuous random variable as $p(\cdot)$. We denote a vector with a lowercase letter in bold, such as $\boldsymbol{x}$ or $\boldsymbol{y}$ and a matrix with an uppercase letter in bold, such as $\boldsymbol{X}$ or $\boldsymbol{Y}$. $\boldsymbol{y} - x$ means that the scalar x is subtracted from all elements in the vector $\boldsymbol{y}$.

2.1.2 Notations on the Time-Sequence Data

Throughout this thesis, we denote the set of time-sequence data from $K \in \mathbb{N}^+$ independent subjects as $\mathcal{D} = \{\boldsymbol{d}_k\}_{k=1}^K$. $\boldsymbol{d}_k$ is the time-sequence data which we collected from the kth subject. Each subject will generate a sequence of events in an observation window $\mathcal{X}^{(k)} \subset \mathbb{R}$. Since each subject will join and drop out from the experiment at different time-stamps, $\mathcal{X}^{(k)}$ are not necessarily the same.

In the *recurrent event data*, the time-stamp of each event is a scalar and is fully observable. We denote the number of events observed from the kth subject as $N_k \in \mathbb{N}^+$. The time-sequence data from the kth subject can be represented as follows:

$$\boldsymbol{d}_k \triangleq \left\{x_j^{(k)} \in \mathcal{X}^{(k)}\right\}_{j=1}^{N_k}. \tag{2.1}$$

In the *panel count data*, we do not know the exact time-stamp for each event and the kth subject is assessed in $N_k \in \mathbb{N}^+$ intervals $\{\mathcal{X}_i^{(k)}\}_{i=1}^{N_k}$, and these intervals

satisfy the following condition:

$$\mathcal{X}_i^{(k)} \bigcap \mathcal{X}_j^{(k)} = \emptyset, i \neq j, \quad \bigcup_{i=1}^{N_k} \mathcal{X}_i^{(k)} = \mathcal{X}^{(k)}.$$

For the kth subject, we have access to each interval $\mathcal{X}_i^{(k)}$ and the number of events observed in this interval $m_i^{(k)} = |\{x_j^{(k)} \in \mathcal{X}_i^{(k)}\}|$. The panel count data from each subject can be represented as follows:

$$\boldsymbol{d}_k \triangleq \left\{(\mathcal{X}_i^{(k)}, m_i^{(k)})\right\}_{i=1}^{N_k}. \tag{2.2}$$

2.2 Basic Concepts of Stochastic Processes

We briefly review the concepts of the probability space, the random variable and the stochastic process. For more rigorous discussions, the readers are referred to Durrett [23], Stark and Woods [90] and Gallager [31]. These basic concepts are essential in understanding concrete examples of stochastic processes in the following sections.

Definition 2.2.1. *A* probability space $(\Omega, \mathcal{F}, P)$ *is a triple of the sample space* Ω*, the* σ*-field* $\mathcal{F}$ *and the probability measure* P*.*

The sample space Ω defines the possible outcomes in the experiment. The σ-field $\mathcal{F}$ is a non-empty collection of the subsets of Ω and satisfies the following properties:

- $\Omega \in \mathcal{F}$.
- If $A \in \mathcal{F}$, then $\Omega \setminus A \in \mathcal{F}$.
- For a countable collection of the subsets $\{A_i \in \mathcal{F}\}_{i=1}^{\infty}$, then

$$\bigcup_{i=1}^{\infty} A_i \in \mathcal{F}.$$

These properties indicate that $\mathcal{F}$ contains the universal set Ω, is closed under the complementation and is closed under the countable union. Each element in the σ-field is called an event and the σ-field defines the collection of events we are interested in from the sample space Ω. The probability measure P is defined on the σ-field $\mathcal{F}$ to measure the probability of each event. More specifically, the probability measure is a function $P : \mathcal{F} \to \mathbb{R}$ which satisfies the following properties:

- $P(\Omega) = 1$.
- $P(A) \geq 0, \forall A \in \mathcal{F}$.
- For a countable collection of the disjoint subsets $\{A_i \in \mathcal{F}\}_{i=1}^{\infty}$, $A_i \cap A_j = \emptyset, \forall i \neq j$,

$$P\Big(\bigcup_{i=1}^{\infty} A_i\Big) = \sum_{i=1}^{\infty} P(A_i).$$

These properties guarantee that a probability measure is normalized, non-negative and countably additive.

Next we use the experiment of tossing a fair coin as an example to explain the concepts of the probability space. The possible outcome when tossing a fair coin forms the sample space $\Omega = \{\text{HEAD}, \text{TAIL}\}$. One σ-field $\mathcal{F}$ is $\{\emptyset, \{\text{HEAD}\}, \{\text{TAIL}\}, \{\text{HEAD}, \text{TAIL}\}\}$. Here the empty set $\emptyset$ represents the situation when the outcome is neither head nor tail and the subset $\{\text{HEAD}, \text{TAIL}\}$ represents the situation when the outcome is either head or tail. Since the coin is fair, we can then assign a probability to all events in $\mathcal{F}$ with the probability measure P.

$$P(\emptyset) = 0,\ P(\{\text{HEAD}\}) = P(\{\text{TAIL}\}) = 1/2,$$
$$P(\{\text{HEAD}, \text{TAIL}\}) = P(\{\text{HEAD}\}) + P(\{\text{TAIL}\}) = 1.$$

Based on the concepts of the probability space, we are ready to define the random variable and the stochastic process.

Definition 2.2.2. *A* random variable X *is a function* $X : \Omega \to A$*, where* Ω *is the sample space and* A *is a measurable space. Usually* A *is the space of real numbers, i.e.,* $A = \mathbb{R}$*. The mapping satisfies the property that for every subset* $S \in A$*, the set* $\{\omega | X(\omega) \in S\}$ *is an event in the* σ*-field* $\mathcal{F}$*.*

When we perform the function mapping $X : \Omega \to A$, the σ-field $\mathcal{B}$ in the original probability space is also mapped to a new σ-field, $\mathcal{B}_A$. The mapping property ensures that the output of a random variable will inherit its own probability measure [36]. For example, we can define the probability measure P_X on the σ-field $\mathcal{B}_A$.

$$P_X(B) \triangleq P(X^{-1}(B)) = P(\{\omega | X(\omega) \in \mathcal{B}\}), \forall B \in \mathcal{B}_A. \tag{2.3}$$

We can show that P_X is a probability measure from the elementary set theory and the space $(A, \mathcal{B}_A, P_X)$ is a probability space.

As an example of the random variable, we re-consider the experiment of tossing a coin. We can define the following random variable X based on the outcome $\omega \in \Omega = \{\text{HEAD}, \text{TAIL}\}$. X is also called a Bernoulli random variable.

$$X(\omega) = \begin{cases} 1 & \text{if } \omega = \text{HEAD}, \\ 0 & \text{if } \omega = \text{TAIL}. \end{cases}$$

In this case, $A = \{1, 0\}$.

Definition 2.2.3. *A* stochastic process *is an infinite set of random variables defined on the same probability space* $(\Omega, \mathcal{F}, P)$*.*

A discrete stochastic process is denoted with $\{X_i\}_{i \in I}$ with I indicating a countable index set, while a continuous stochastic process is usually denoted as a function $X(t)$, where t is a point in the continuous space $\mathcal{T}$. Note that each random variable $X(t)$ is a function of the experiment outcome $w \in \Omega$.

2.3 Dirichlet Processes and Stick-breaking Processes

We discuss the concept of a Dirichlet process, which is introduced by Ferguson [27], as the first useful tool from the arsenal of stochastic processes. A Dirichlet process can be used to automatically determine the number of components in a mixture model such as the Gaussian mixture model [71]. It has been widely used in the document modeling [99] and the factor analysis [94].

2.3.1 Definition of a Dirichlet Process

Following the routine used in Orbanz and Teh [73], we first define the Dirac measure and then define the Dirichlet process through the stick-breaking process.

Definition 2.3.1. *Let Ω be the sample space and $\mathcal{F}$ be a σ-field on Ω. A* Dirac measure *δ_ϕ is the probability measure which satisfies the property $\forall A \in \mathcal{F}$,*

$$\delta_\phi(A) = \begin{cases} 1 & \text{if } \phi \in A, \\ 0 & \text{if } \phi \notin A. \end{cases}$$

The Dirac measure assigns the mass 1 to the single point ϕ in the sample space and is used as the basic element (atom) in the Dirichlet process.

Definition 2.3.2. *If $\alpha > 0$ and if G is a probability measure on a sample space of models Ω, the random discrete probability measure Θ which is generated by*

$$V_k \sim \text{Beta}(1, \alpha), \ k = 1, 2, \ldots, \infty, \tag{2.4}$$

$$Y_k = V_k \prod_{i=1}^{k-1} (1 - V_i), \ k = 1, 2, \ldots, \infty, \tag{2.5}$$

$$\Phi_k \sim G, \ \Theta = \sum_{k=1}^{\infty} Y_k \delta_{\Phi_k},$$

is called a Dirichlet process *(DP) with the base measure G and the concentration parameter α. We denote a sample from a DP as $\Theta \sim \text{DP}(\alpha, G)$. The sampling procedure for the random variables $\{Y_k\}_{k=1}^{\infty}$ is called a* stick-breaking process.

The name "stick-breaking" comes from the construction procedure. We can imagine that we originally have a stick with length 1 and at each step we repeatedly break off a portion V_k of the remaining stick. Each piece has a length of Y_k. Finally we obtain a set of pieces $\{Y_k\}_{k=1}^{\infty}$. The sum of the first K pieces can be computed as follows:

$$\sum_{k=1}^{K} Y_k = \sum_{k=1}^{K} V_k \prod_{j=1}^{k-1} (1 - V_j) = 1 - \prod_{k=1}^{K} (1 - V_k).$$

As $K \to \infty$, the expected sum of the all the pieces yields

$$\lim_{K \to \infty} \mathbb{E}\Big[\sum_{k=1}^{K} Y_k\Big] = 1 - \lim_{K \to \infty} \prod_{k=1}^{K} \mathbb{E}(1 - V_k) = 1 - \lim_{K \to \infty} \Big(\frac{\alpha}{1 + \alpha}\Big)^K = 1.$$

Two examples of the stick-breaking process with different hyper-parameters α are shown in Figure 2.1.

An interesting property [27] is that the length of the pieces are ordered in the way that on average a piece with a smaller index k will have a larger length Y_k than a piece with a larger index. More formally, this property is given by the following theorem. This property can alleviate the identifiability issue [71] in the mixture models.

Theorem 2.3.1 (Ferguson [27])**.** *For an experimental outcome from a stick-breaking process described in Equations* (2.4) *and* (2.5)*, the following inequality holds.*

$$\mathbb{E}[Y_i] > \mathbb{E}[Y_j], \quad \forall i > j.$$

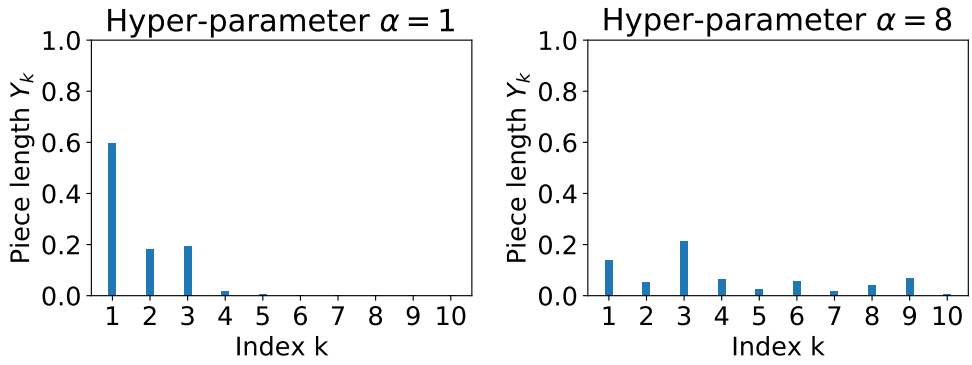

Figure 2.1: The illustration of the lengths of the first 10 pieces $\{Y_i\}_{i=1}^{10}$ from two stick-breaking processes. (Left) A stick-breaking process with the concentration parameter $\alpha = 1$. (Right) A stick-breaking process with the concentration parameter $\alpha = 8$. Notice that the process does not end within 10 steps and the sum of the first 10 pieces is smaller than 1.

Proof. Since $\{V_i\}$ are identical independent beta-distributed random variables, we can obtain the expectation of Y_k.

$$\mathbb{E}[Y_k] = \mathbb{E}\Big[\sum_{k=1}^{\infty} V_k \prod_{i=1}^{k-1}(1 - V_i)\Big] = \mathbb{E}[V_k] \prod_{i=1}^{k-1} \mathbb{E}[1 - V_i] = \frac{1}{1+\alpha}\Big(\frac{\alpha}{1+\alpha}\Big)^{k-1}.$$

This expectation will decrease with the increase of the index k. □

2.3.2 Variational Inference with a Dirichlet Process

To perform the inference for the probabilistic models with a Dirichlet process, different approaches including the Markov Chain Monte Carlo (MCMC) sampling with the Chinese restaurant process [10] and the variational inference approach [11] have been proposed.

Next we discuss the variational inference method with a truncated stick-breaking representation [11] below. In the variational inference framework, the variational distribution $q(\{V_k\})$ that we use to approximate the stick-breaking process in Equations (2.4) and (2.5) is defined as follows:

$$q(V_k) = \begin{cases} \text{Beta}(\tau_{k1}, \tau_{k2}) & \text{if } k < K, \\ \delta_1 & \text{if } k = K. \end{cases}$$

$$Y_k = V_k \prod_{i=1}^{k-1}(1 - V_i), \;\; k = 1, 2, \ldots, K.$$

τ_{k1}, τ_{k2} are two positive real numbers. The variational distribution is a truncation of the original Dirichlet process and the value K as the maximum number of components which can be used by the model. Also notice that the original Dirichlet process prior is not truncated, and the truncation is designed for the feasible inference [11].

Let $\tau_{k1} = 1$ and $\tau_{k2} = \alpha$ and the maximum number of components be $K = 10$. We illustrate two random draws from the truncation procedure with two different settings of the parameter α in Figure 2.2. We can notice that the sum of the 10 pieces is exactly 1.

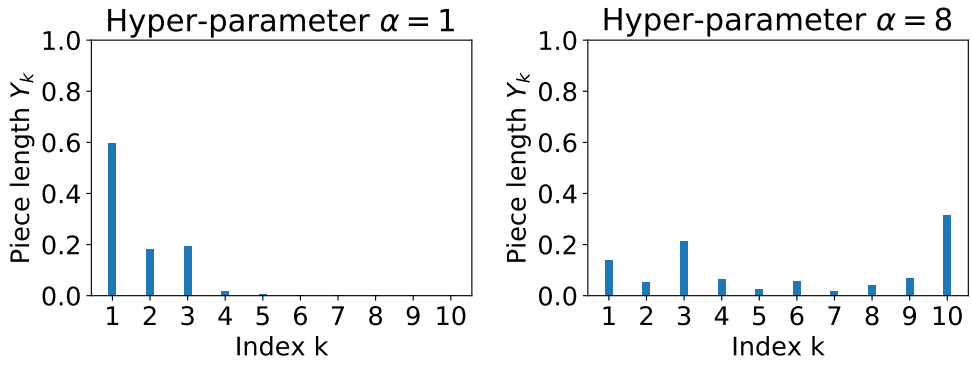

Figure 2.2: The illustration of the lengths of the pieces $\{Y_i\}_{i=1}^{10}$ from two truncated stick-breaking procedure with the maximum number of components $K = 10$. (Left) The concentration parameter $\alpha = 1$. (Right) The concentration parameter $\alpha = 8$.

2.3.3 An Example: Dirichlet Process Gaussian Mixture Model

As a concrete example, we briefly review the variational inference method for the Dirichlet process Gaussian mixture model (DPGMM) [11, 92, 35]. In a DPGMM, the data points are assumed to be generated from an infinite mixture of Gaussian distributions with unknown parameters. The DPGMM provides a solution to jointly infer the number of Gaussian distributions and their parameters.

We consider the simple case where we have a data set $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ with each data point $\boldsymbol{x}_i \in \mathbb{R}^2$. Let $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$ be a Gaussian distribution with the mean vector $\boldsymbol{\mu}$ and the precision matrix $\boldsymbol{\Lambda}$. Let $\mathcal{NW}(\boldsymbol{\mu}_0, \lambda, \boldsymbol{W}, \rho)$ be the normal-Wishart distribution [71] with the parameters $\{\boldsymbol{\mu}_0, \lambda, \boldsymbol{W}, \rho\}$.

$$\begin{aligned}\mathcal{NW}(\boldsymbol{\mu}, \boldsymbol{\Lambda}; \boldsymbol{\mu}_0, \lambda, \boldsymbol{W}, \rho) \propto & |\boldsymbol{\Lambda}|^{\frac{1}{2}} \exp\Big(-\frac{1}{2}(\boldsymbol{\mu} - \boldsymbol{\mu_0})^\top \lambda \boldsymbol{\Lambda}(\boldsymbol{\mu} - \boldsymbol{\mu_0})\Big) \\ & \times |\boldsymbol{\Lambda}|^{\frac{\rho-3}{2}} \exp\Big(-\frac{1}{2}\mathrm{tr}(\boldsymbol{W}^{-1}\boldsymbol{\Lambda})\Big).\end{aligned}$$

The generative process in a DPGMM is given in Algorithm 1.

Algorithm 1: The generative process of the DPGMM.

Input : The hyper-parameters $\{\alpha, \boldsymbol{\mu}_0, \lambda, \boldsymbol{W}, \rho\}$ and the data set size N.
Output: A data set $\{\boldsymbol{x}_n\}_{n=1}^N$.

for *each component* $k = 1, 2, \ldots, \infty$ **do**
  Sample $V_k \sim \mathrm{Beta}(1, \alpha)$ and obtain the observation v_k.
  Caluclate $y_k = v_k \prod_{i=1}^{k-1}(1 - v_i)$.
  Sample $(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \sim \mathcal{NW}(\boldsymbol{\mu}_0, \lambda, \boldsymbol{W}, \rho)$.
end
Set $\boldsymbol{y} = [y_1, \ldots, y_\infty]$.
for *each data point* $n = 1, \ldots, N$ **do**
  Sample $c_n \sim \mathrm{Multinomial}(\boldsymbol{y})$.
  Sample $\boldsymbol{x}_n \sim \mathcal{N}(\boldsymbol{\mu}_{c_n}, \boldsymbol{\Lambda}_{c_n}^{-1})$.
end

Based on Algorithm 1, the joint distribution of the data and the hidden vari-

ables is

$$
\begin{aligned}
&p(\{v_k\}_{k=1}^{\infty}, \{\boldsymbol{\mu}\}_{k=1}^{\infty}, \{\boldsymbol{\Lambda}\}_{k=1}^{\infty}, \{c_n\}_{n=1}^{N}, \{\boldsymbol{x}\}_{n=1}^{N}) \\
=&\prod_{k=1}^{\infty} \mathrm{Beta}(v_k; 1, \alpha) \prod_{k=1}^{\infty} \mathcal{NW}(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k; \boldsymbol{\mu}_0, \lambda, \boldsymbol{W}, \rho) \prod_{n=1}^{N} \prod_{k=1}^{\infty} y_k^{\mathbb{I}(c_n=k)} \\
&\times \prod_{n=1}^{N} \mathcal{N}(\boldsymbol{x}_n; \boldsymbol{\mu}_{c_n}, \boldsymbol{\Lambda}_{c_n}^{-1}).
\end{aligned}
$$

Algorithm 2: The variational inference of the DPGMM.

Input : The hyper-parameters $\{\alpha, \boldsymbol{\mu}_0, \lambda, \boldsymbol{W}, \rho\}$ and the data set $\{\boldsymbol{x}_n\}_{n=1}^{N}$.
Output: The parameters in the variational distribution $\{\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \Phi, B\}$.

Initialize $\{\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \Phi, B\}$.
while *not converge* **do**
  for *each data point* $n = 1, \ldots, N$ *and each component* $k = 1, \ldots, T$ **do**
    Update β_{nk}.

$$\beta_{nk} \propto \exp\left(\mathbb{E}_q[\ln v_k] + \sum_{j=1}^{k-1} \mathbb{E}_q[\ln(1 - v_j)] + \mathbb{E}_q \ln \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})\right). \tag{2.6}$$

  end
  for *each component* $k = 1, \ldots, K$ **do**
    Calculate $\bar{x} = \left(\sum_{n=1}^{N} \beta_{nk} \boldsymbol{x}_n\right) / \sum_{n=1}^{N} \beta_{nk}$ and $\hat{\beta} = \sum_{n=1}^{N} \beta_{nk}$.
    Update the parameters for this component.

$$
\begin{aligned}
&\tau_{k1} = 1 + \sum_{n=1}^{N} \beta_{nk}, \ \tau_{k2} = \alpha + \sum_{n=1}^{N} \sum_{j=k+1}^{K} \beta_{nj}, \\
&\tilde{\lambda}_k = \lambda + \sum_{n=1}^{N} \beta_{nk}, \ \tilde{\rho}_k = \rho + \sum_{n=1}^{N} \beta_{nk}, \\
&\tilde{\boldsymbol{\mu}}_{k0} = \frac{\lambda \boldsymbol{\mu}_0 + \sum_{n=1}^{N} \beta_{nk} \boldsymbol{x}_n}{\lambda + \hat{\beta}}, \\
&\tilde{\boldsymbol{W}}_k^{-1} = \boldsymbol{W}^{-1} + \frac{\lambda \hat{\beta}}{\lambda + \hat{\beta}} (\boldsymbol{\mu} - \bar{\boldsymbol{x}})(\boldsymbol{\mu} - \bar{\boldsymbol{x}})^{\top} + \sum_{n=1}^{N} \beta_{nk} (\boldsymbol{x} - \bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^{\top}.
\end{aligned}
$$

  end
end

The variational distribution with the truncation level K is chosen as follows:

$$
\begin{aligned}
&q(\{v_k\}_{k=1}^{K}, \{\boldsymbol{\mu}\}_{k=1}^{K}, \{\boldsymbol{\Lambda}\}_{k=1}^{K}, \{c_n\}_{n=1}^{N}) \\
=&q(v_K) \prod_{k=1}^{K-1} \mathrm{Beta}(v_k; \tau_{k1}, \tau_{k2}) \prod_{k=1}^{K} \mathcal{NW}(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k; \tilde{\boldsymbol{\mu}}_{k0}, \tilde{\lambda}_k, \tilde{\boldsymbol{W}}_k, \tilde{\rho}_k) \prod_{n=1}^{N} \prod_{k=1}^{K} \beta_{nk}^{\mathbb{I}(c_n=k)},
\end{aligned}
$$

where $q(v_K) = \delta_1$ and $\mathbb{I}(c_n = k)$ is an indicator function with the property

$$\mathbb{I}(c_n = k) = \begin{cases} 1 & \text{if } c_n = k, \\ 0 & \text{if } c_n \neq k. \end{cases}$$

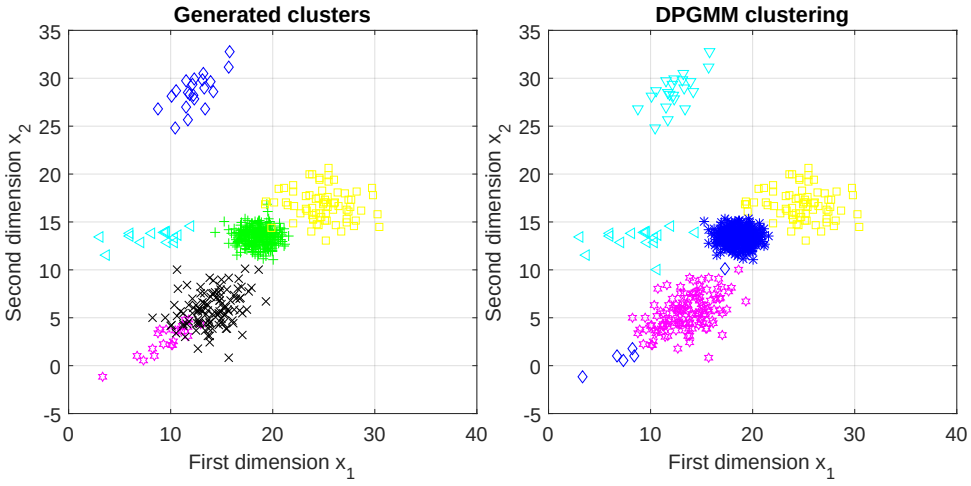

Figure 2.3: An illustration of the variational inference for the DPGMM. Data points are two-dimensional, i.e., $\boldsymbol{x} = [x_1, x_2] \in \mathbb{R}^2$. (Left) One generated data set with 6 components. (Right). The result from DPGMM with the truncation level $K = 50$ identifies 5 clusters.

In the framework of the variational inference, we optimize the variational distribution by maximizing the evidence lower bound of the data likelihood.

$$\begin{aligned}\ln p(\{\boldsymbol{x}\}_{n=1}^N) \geq \mathbb{E}_q\Big[\ln p(\{v_k\}_{k=1}^\infty, \{\boldsymbol{\mu}\}_{k=1}^\infty, \{\boldsymbol{\Lambda}\}_{k=1}^\infty, \{c_n\}_{n=1}^N, \{\boldsymbol{x}\}_{n=1}^N)\Big] \\ - \mathbb{E}_q[\ln q(\{v_k\}_{k=1}^K, \{\boldsymbol{\mu}\}_{k=1}^K, \{\boldsymbol{\Lambda}\}_{k=1}^K, \{c_n\}_{n=1}^N)].\end{aligned} \tag{2.7}$$

Let $\boldsymbol{\tau}_1 = \{\tau_{k1}\}_{k=1}^{K-1}$, $\boldsymbol{\tau}_2 = \{\tau_{k2}\}_{k=1}^{K-1}$, $\Phi = \{\tilde{\boldsymbol{\mu}}_{k0}, \tilde{\lambda}_k, \tilde{\boldsymbol{W}}_k, \tilde{\rho}_k\}_{k=1}^K$ and $B = \{\beta_{nk}\}$. The parameters to be optimized in the rightmost side of Equation (2.7) are $\{\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \Phi, B\}$. Since the normal-Wishart distribution is conjugate to the data likelihood, we can obtain the following updating rule [11] in Algorithm 2. Let $\Psi(\cdot)$ be the digamma function. In Equation (2.6), the three expectations can be analytically computed as follows:

$$\begin{aligned}\mathbb{E}_q[\ln v_k] &= \Psi(\tau_{k1}) - \Psi(\tau_{k1} + \tau_{k2}), \\ \mathbb{E}_q[\ln(1 - v_j)] &= \Psi(\tau_{j2}) - \Psi(\tau_{j1} + \tau_{j2}), \\ \mathbb{E}_q \ln \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}) &= \frac{1}{2}\Big(\Psi\Big(\frac{\tilde{\rho}}{2}\Big) + \Psi\Big(\frac{\tilde{\rho} - 1}{2}\Big)\Big) + \frac{1}{2}\ln|\tilde{\boldsymbol{W}}_k| - \frac{1}{\tilde{\lambda}_k} \\ &\quad - \frac{\tilde{\rho}_k}{2}(\boldsymbol{\mu}_{k0} - \boldsymbol{x}_n)^\top \tilde{W}_k(\boldsymbol{\mu}_{k0} - \boldsymbol{x}_n).\end{aligned}$$

An illustration of one sampling data set and the variational inference result of the data set are provided in Figure 2.3. We can observe that even with a large $K = 50$, the variational inference algorithm can still identify a small number of clusters.

2.4 Gaussian Processes

A Gaussian process can be used as a prior for an unknown function and is applied widely in Bayesian non-linear nonparametric regression and classification [77]. The definition of a Gaussian process is given as follows.

Definition 2.4.1. *A* Gaussian process *is a collection of random variables, any finite of which follows a joint Gaussian distribution.*

In Definition 2.2.3, a stochastic process is a set of random variables and for a Gaussian process, the set is the input space $\mathcal{X} \subset \mathbb{R}^D$, with D indicating the number of the inputs. A Gaussian process is specified by the mean function and the covariance function. We denote the mean function $m(\boldsymbol{x})$ and the covariance function $\kappa(\boldsymbol{x}, \boldsymbol{x}')$ of a real-valued process $f(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}$ as follows:

$$\begin{aligned} m(\boldsymbol{x}) &= \mathbb{E}[f(\boldsymbol{x})], \\ \kappa(\boldsymbol{x}, \boldsymbol{x}') &= \mathbb{E}[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x}') - m(\boldsymbol{x}'))]. \end{aligned}$$

We denote a function f drawn from a Gaussian process with the mean function $m(\boldsymbol{x})$ and the covariance function $\kappa(\boldsymbol{x}, \boldsymbol{x}')$ as follows:

$$f \sim \mathcal{GP}(m(\boldsymbol{x}), \kappa(\boldsymbol{x}, \boldsymbol{x}')).$$

For any set of points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ with each $\boldsymbol{x}_i \in \mathcal{X}$, we denote the input matrix as $\boldsymbol{X} \triangleq [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$. According to Definition 2.4.1, we have a joint Gaussian distribution for the output vector $\boldsymbol{f} \triangleq [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]^\top \in \mathbb{R}^{N \times 1}$.

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}), \tag{2.8}$$

where the mean vector $\boldsymbol{m} \triangleq [m(\boldsymbol{x}_1), \ldots, m(\boldsymbol{x}_N)]^\top \in \mathbb{R}^{N \times 1}$ and the covariance matrix $\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}$ can be computed as follows:

$$\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} \triangleq \begin{bmatrix} \kappa(\boldsymbol{x}_1, \boldsymbol{x}_1) & \ldots & \kappa(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ \kappa(\boldsymbol{x}_N, \boldsymbol{x}_1) & \ldots & \kappa(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}. \tag{2.9}$$

A covariance matrix of this kind is also named a Gram matrix in the kernel methods [70]. For the covariance function, we use the automatic relevance determination (ARD) kernel in the following discussion. Let x_j be the j-th element of the vector $\boldsymbol{x}$.

$$\kappa_{\text{ARD}}(\boldsymbol{x}, \boldsymbol{x}') = c \exp\Big(-\frac{1}{2} \sum_{j=1}^{D} b_j (x_j - x_j')^2 \Big), b_j \geq 0. \tag{2.10}$$

The hyper-parameters in the ARD kernel are $\{c, \boldsymbol{b}\}$. This kernel is also known as the squared exponential (SE) kernel.

2.4.1 Gaussian Processes for Regression

In a regression task, we are given a data set consisting of input-response tuples $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ with each $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^D$ and each $y_i \in \mathcal{Y} \subset \mathbb{R}$ is a real number. We denote $\boldsymbol{y} = \{y_i\}_{i=1}^N$. Our task is to learn a mapping $f : \mathcal{X} \to \mathcal{Y}$.

In the standard Gaussian Process regression method [102], we place a Gaussian process prior on the function f and $f \sim \mathcal{GP}(\boldsymbol{m}(\boldsymbol{x}), \kappa(\boldsymbol{x}, \boldsymbol{x}'))$ and the mean function $\boldsymbol{m}(\boldsymbol{x})$ is usually chosen to be a constant $m(\boldsymbol{x}) = m_0$. We place a Gaussian noise model for the observed data tuple $(\boldsymbol{x}, y)$ and $p(y|f; \boldsymbol{x}, \sigma) = \mathcal{N}(f(\boldsymbol{x}), \sigma^2)$ where $\mathcal{N}(\cdot)$ is a normal distribution. Assuming the noises are added independently to the function values $\{f(\boldsymbol{x}_i)\}$ conditioned on the latent function f, the joint likelihood is computed as follows:

$$p(\boldsymbol{f}, \boldsymbol{y}) = p(\boldsymbol{f}) p(\boldsymbol{y}|\boldsymbol{f}) = p(\boldsymbol{f}) \prod_{i=1}^{N} p(y_i | f(\boldsymbol{x}_i)).$$

The marginal likelihood $p(\boldsymbol{y})$ is can be computed below.

$$\begin{aligned}
p(\boldsymbol{y}) =& \int p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f})d\boldsymbol{f} \\
=& \int \Big[\prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{(y_i - f(\boldsymbol{x}_i))^2}{2\sigma^2}\Big)\Big] \\
& \times \frac{1}{(2\pi)^{N/2}|\boldsymbol{K}_{\boldsymbol{XX}}|^{1/2}} \exp\Big(-\frac{1}{2}(\boldsymbol{f} - m_0)^\top \boldsymbol{K}_{\boldsymbol{XX}}^{-1}(\boldsymbol{f} - m_0)\Big)d\boldsymbol{f} \\
=& \mathcal{N}\Big(\boldsymbol{y}\Big|\boldsymbol{m}_0, \boldsymbol{K}_{\boldsymbol{XX}} + \sigma^2 I\Big),
\end{aligned}$$

where we denote $\boldsymbol{m}_0 \in \mathbb{R}^N$ as a vector with all elements equal to m_0. The learning in the Gaussian process regression method is conducted by maximizing the marginal likelihood $p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}\big|\boldsymbol{m}_0, \boldsymbol{K}_{\boldsymbol{XX}} + \sigma^2 I)$ with respect to the hyper-parameters in the covariance function (such as the hyper-paramters $\{c, \boldsymbol{b}\}$ in the ARD covariance function), m_0 in the mean function and the noise level σ.

To predict the value of the function at an arbitrary point $\boldsymbol{x}^*$ in the Gaussian process regression, we will examine the posterior process when we have the data likelihood $p(y|f; \boldsymbol{x}, \sigma) = \mathcal{N}(f(\boldsymbol{x}), \sigma^2)$.

Theorem 2.4.1 (Rasmussen [77])**.** *Assuming that the prior is a Gaussian process $f \sim \mathcal{GP}(m(\boldsymbol{x}),\ \kappa(\boldsymbol{x}, \boldsymbol{x}'))$ and the data likelihood is $p(\boldsymbol{y}) = \prod_{i=1}^{N} \mathcal{N}(0, \sigma^2)$, the posterior of this Gaussian process is still a Gaussian process $\mathcal{GP}(\hat{m}(\boldsymbol{x}), \hat{\kappa}(\boldsymbol{x}, \boldsymbol{x}'))$, with the mean function $\hat{m}(\boldsymbol{x})$ and the covariance function $\hat{\kappa}(\boldsymbol{x}, \boldsymbol{x}')$ given by the following equations.*

$$\begin{aligned}
\hat{m}(\boldsymbol{x}) &= m_0 + \boldsymbol{\kappa}(\boldsymbol{x}, \boldsymbol{X})(\boldsymbol{K}_{\boldsymbol{XX}} + \sigma^2 I)^{-1}(\boldsymbol{y} - m_0), \\
\hat{\kappa}(\boldsymbol{x}, \boldsymbol{x}') &= \boldsymbol{\kappa}(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{\kappa}(\boldsymbol{x}, \boldsymbol{X})(\boldsymbol{K}_{\boldsymbol{XX}} + \sigma^2 I)^{-1}\boldsymbol{\kappa}(\boldsymbol{X}, \boldsymbol{x}').
\end{aligned}$$

Proof. According to Definition 2.4.1, the joint distribution of an observed data matrix $\boldsymbol{X}$ and any un-observed data matrix $\boldsymbol{X}^*$ is Gaussian-distributed.

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}^* \end{bmatrix} = \mathcal{N}\Big(\begin{bmatrix} \boldsymbol{m}_0 \\ \boldsymbol{m}_0^* \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{XX}} & \boldsymbol{K}_{\boldsymbol{XX}^*} \\ \boldsymbol{K}_{\boldsymbol{X}^*\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X}^*\boldsymbol{X}^*} \end{bmatrix}\Big).$$

After observing the output $\boldsymbol{y}$, the joint distribution of $\boldsymbol{y}$ and $\boldsymbol{f}^*$ can be derived by integrating out $\boldsymbol{f}$ in $p(\boldsymbol{y}, \boldsymbol{f}, \boldsymbol{f^*})$ and the result is:

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}^* \end{bmatrix} = \mathcal{N}\Big(\begin{bmatrix} \boldsymbol{m}_0 \\ \boldsymbol{m}_0^* \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{XX}} + \sigma^2 I & \boldsymbol{K}_{\boldsymbol{XX}^*} \\ \boldsymbol{K}_{\boldsymbol{X}^*\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X}^*\boldsymbol{X}^*} \end{bmatrix}\Big).$$

The posterior distribution of any un-observed data $p(\boldsymbol{f}^*|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}^*)$ can be calculated from the joint distribution and the result is given as follows:

$$p(\boldsymbol{f}^*|\boldsymbol{y}; \boldsymbol{X}, \boldsymbol{X}^*) = \mathcal{N}(\hat{\boldsymbol{m}}(\boldsymbol{X}^*), \hat{\boldsymbol{K}}_{\boldsymbol{X}^*\boldsymbol{X}^*}). \tag{2.11}$$

If we take the observed data into concern, the joint distribution is as follows:

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f} \\ \boldsymbol{f}^* \end{bmatrix} = \mathcal{N}\Big(\begin{bmatrix} \boldsymbol{m}_0 \\ \boldsymbol{m}_0 \\ \boldsymbol{m}_0^* \end{bmatrix}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{XX}} + \sigma^2 I & \boldsymbol{K}_{\boldsymbol{XX}} & \boldsymbol{K}_{\boldsymbol{XX}^*} \\ \boldsymbol{K}_{\boldsymbol{XX}} & \boldsymbol{K}_{\boldsymbol{XX}} & \boldsymbol{K}_{\boldsymbol{XX}^*} \\ \boldsymbol{K}_{\boldsymbol{X}^*\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X}^*\boldsymbol{X}} & \boldsymbol{K}_{\boldsymbol{X}^*\boldsymbol{X}^*} \end{bmatrix}\Big).$$

Similarly, when there are a part of the observed data in the joint distribution, the result is still the same. From Definition 2.4.1, we conclude that the posterior process is still a Gaussian process.

□

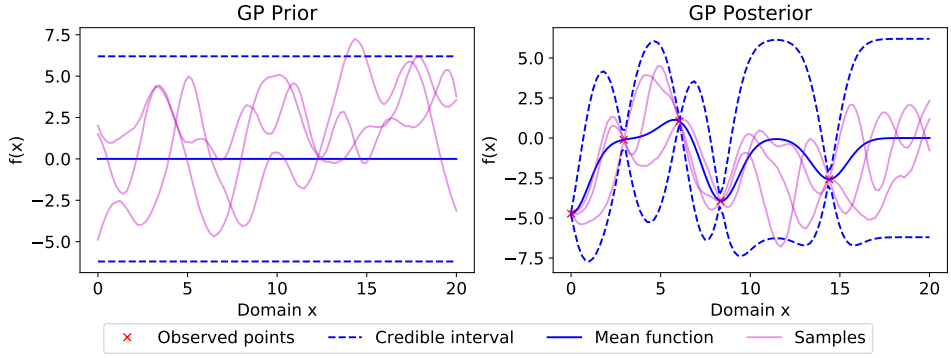

Figure 2.4: The illustration of the prior distribution and the posterior distribution of the Gaussian process in a regression task. A 95% credible interval is also provided for both the prior and the posterior distribution. We also plot three sampled random functions for both the prior and the posterior distribution.

An illustration of the prior distribution and the posterior distribution after we observer several data tuples are shown in Figure 2.4. We notice that the uncertainty about the shape of the function is reduced after observing the data tuples.

Given the above theorem, the prediction of the function value at an arbitrary point $\boldsymbol{x}^*$ is given by a Gaussian distribution since according Definition 2.4.1, the function value at any given point is a Gaussian distributed random variable.

$$f(\boldsymbol{x}^*) \sim \mathcal{N}(\hat{m}(\boldsymbol{x}), \hat{\kappa}(\boldsymbol{x}^*, \boldsymbol{x}^*)). \tag{2.12}$$

During the training of the Gaussian process regression, the bottleneck of the computational complexity [1] lies in the inversion of the covariance matrix $\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} + \sigma I$ and is $O(N^3)$ [88]. Once we obtain the inversion of the covariance matrix, the computational complexity for one test point is $O(N)$ for computing the mean value $\hat{m}(\boldsymbol{x}^*)$ and $O(N^2)$ for computing the variance $\hat{\kappa}(\boldsymbol{x}^*, \boldsymbol{x}^*)$.

2.4.2 Sparse Gaussian Processes for Regression

In order to reduce the computational complexity in the standard Gaussian process regression, various methods have been proposed. One approach [88, 86] is to modify the prior of the Gaussian process with pseudo inputs and then learn the hyper-parameters by maximizing the marginal likelihood with the modified prior. However, the number and the positions of the pseudo inputs in these methods are additional hyper-parameters in the marginal likelihood to be optimized. It can lead to over-fitting in the experiments [96]. Another line of researches [96, 40] utilizes the framework of the variational inference and selects the pseudo inputs and the hyper-parameters by maximizing a lower bound of the exact marginal likelihood.

In the sparse Gaussian process method (sGP)[96], a set of pseudo inputs $\bar{\boldsymbol{X}} = \{\bar{\boldsymbol{x}}_m\}_{m=1}^M, M < N$ and their corresponding function values $\bar{\boldsymbol{f}} = \{f(\bar{\boldsymbol{x}}_m)\}_{m=1}^M$ are considered. The name "pseudo input" indicates that the function values at these points are not directly observed.

The first intuition is that we can augment the joint distribution $p(\boldsymbol{y}, \boldsymbol{f})$ with the additional function values $\bar{\boldsymbol{f}}$ without changing the marginal distribution of

[1] To the best of our knowledge, the current state of the art for the inversion of a matrix is $O(N^{2.3728639})$ [54].

$p(\boldsymbol{y}, \boldsymbol{f})$. The distribution of the function values of the pseudo inputs $p(\bar{\boldsymbol{f}}; \bar{\boldsymbol{X}})$ and the conditional distribution $p(\boldsymbol{f}|\bar{\boldsymbol{f}}; \boldsymbol{X})$ are defined as follows:

$$\hat{p}(\boldsymbol{f}|\bar{\boldsymbol{f}}; \boldsymbol{X}) = \mathcal{N}(\boldsymbol{f}; m_0 + \boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}(\bar{\boldsymbol{f}} - m_0), \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} - \boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{K}_{\bar{\boldsymbol{X}}\boldsymbol{X}}),$$
$$\hat{p}(\bar{\boldsymbol{f}}; \bar{\boldsymbol{X}}) = \mathcal{N}(\bar{\boldsymbol{f}}; \boldsymbol{m}_0, \boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}).$$

Theorem 2.4.2 (Snelson and Ghahramani [88])**.** *The marginal distribution $\hat{p}(\boldsymbol{y}, \boldsymbol{f})$ equals the marginal distribution constructed by a Gaussian process prior.*

$$p(\boldsymbol{y}, \boldsymbol{f}) = \int p(\boldsymbol{y}|\boldsymbol{f})\hat{p}(\boldsymbol{f}|\bar{\boldsymbol{f}}; \boldsymbol{X})\hat{p}(\bar{\boldsymbol{f}}; \bar{\boldsymbol{X}})d\bar{\boldsymbol{f}}.$$

Proof. We can marginalize the distribution $\hat{p}(\boldsymbol{f}, \bar{\boldsymbol{f}})$ to obtain the marginal distribution of $\hat{p}(\boldsymbol{f})$. Let $\boldsymbol{A} = \boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}$, $\boldsymbol{B} = \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} - \boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{K}_{\bar{\boldsymbol{X}}\boldsymbol{X}}$ and $C = \boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}$.

$$\begin{aligned}\hat{p}(\boldsymbol{f}) &= \int \hat{p}(\boldsymbol{f}|\bar{\boldsymbol{f}}; \boldsymbol{X})\hat{p}(\bar{\boldsymbol{f}}; \bar{\boldsymbol{X}})d\bar{\boldsymbol{f}} \\ &\propto \int \exp\Big(-\frac{1}{2}(\bar{\boldsymbol{f}} - m_0)^\top \boldsymbol{C}^{-1}(\bar{\boldsymbol{f}} - m_0) - \frac{1}{2}(\boldsymbol{f} - \boldsymbol{A}\bar{\boldsymbol{f}})^\top \boldsymbol{B}^{-1}(\boldsymbol{f} - \boldsymbol{A}\bar{\boldsymbol{f}})\Big)d\bar{\boldsymbol{f}}.\end{aligned} \tag{2.13}$$

This indicates that $\hat{p}(\boldsymbol{f})$ is also a Gaussian distribution and we can directly compute the mean vector and the covariance matrix.

$$\begin{aligned}\mathbb{E}[\boldsymbol{f}] &= \mathbb{E}_{\hat{p}(\bar{\boldsymbol{f}})}[\mathbb{E}_{\hat{p}(\boldsymbol{f}|\bar{\boldsymbol{f}})}(\boldsymbol{f})] = \mathbb{E}_{\hat{p}(\bar{\boldsymbol{f}})}[m_0 + \boldsymbol{A}(\bar{\boldsymbol{f}} - m_0)] = \boldsymbol{m}_0 \\ \mathrm{Var}[\boldsymbol{f}] &\triangleq \mathbb{E}[(\boldsymbol{f} - \mathbb{E}[\boldsymbol{f}])(\boldsymbol{f} - \mathbb{E}[\boldsymbol{f}])^\top] = \mathbb{E}[\boldsymbol{f}\boldsymbol{f}^\top] = \mathbb{E}_{\hat{p}(\bar{\boldsymbol{f}})}[\mathbb{E}_{\hat{p}(\boldsymbol{f}|\bar{\boldsymbol{f}})}(\boldsymbol{f}\boldsymbol{f}^\top)] \\ &= \mathbb{E}_{\hat{p}(\bar{\boldsymbol{f}})}[\boldsymbol{B} + \boldsymbol{A}\bar{\boldsymbol{f}}\bar{\boldsymbol{f}}^\top\boldsymbol{A}^T] = \boldsymbol{B} + \boldsymbol{A}\boldsymbol{C}\boldsymbol{A}^\top \\ &= \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} - \boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{K}_{\bar{\boldsymbol{X}}\boldsymbol{X}} + \boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{K}_{\bar{\boldsymbol{X}}\boldsymbol{X}} = \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}.\end{aligned}$$

Since $\hat{p}(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}; \boldsymbol{m}_0, \boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}}) = p(\boldsymbol{f})$, the joint distribution $\hat{p}(\boldsymbol{y}, \boldsymbol{f})$ is the same as the joint distribution with the original Gaussian process prior. □

Within the framework of the variational inference, the variational distribution $q(\boldsymbol{f}, \bar{\boldsymbol{f}})$ is chosen as follows:

$$q(\boldsymbol{f}, \bar{\boldsymbol{f}}) = q(\boldsymbol{f}|\bar{\boldsymbol{f}})q(\bar{\boldsymbol{f}}) = p(\boldsymbol{f}|\bar{\boldsymbol{f}}; \boldsymbol{X})\mathcal{N}(\bar{\boldsymbol{f}}; \boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

Then we can derive a lower bound of the marginal likelihood with the aid of $q(\boldsymbol{f}, \bar{\boldsymbol{f}})$. We call the lower bound the evidence lower bound (ELBO). Let $\boldsymbol{\theta}$ denote all hyper-parameters including the hyper-parameters in the covariance function and the positions of the pseudo inputs. The ELBO can be derived as follows:

$$\begin{aligned}\ln p(\boldsymbol{y}) &= \ln \iint p(\boldsymbol{y}, \boldsymbol{f}, \bar{\boldsymbol{f}}; \boldsymbol{X}, \bar{\boldsymbol{X}})d\boldsymbol{f}d\bar{\boldsymbol{f}} \\ &= \ln \iint \Big(q(\boldsymbol{f}, \bar{\boldsymbol{f}})\frac{p(\boldsymbol{y}, \boldsymbol{f}, \bar{\boldsymbol{f}}; \boldsymbol{X}, \bar{\boldsymbol{X}})}{q(\boldsymbol{f}, \bar{\boldsymbol{f}})}\Big)d\boldsymbol{f}d\bar{\boldsymbol{f}} \\ &\geq \iint q(\boldsymbol{f}, \bar{\boldsymbol{f}}) \ln \frac{p(\boldsymbol{y}, \boldsymbol{f}, \bar{\boldsymbol{f}}; \boldsymbol{X}, \bar{\boldsymbol{X}})}{q(\boldsymbol{f}, \bar{\boldsymbol{f}})}d\boldsymbol{f}d\bar{\boldsymbol{f}} \qquad \text{(Jensen's inequality.)} \\ &= \iint q(\boldsymbol{f}, \bar{\boldsymbol{f}}) \ln \frac{p(\boldsymbol{y}|\boldsymbol{f})p(\boldsymbol{f}|\bar{\boldsymbol{f}}; \boldsymbol{X})p(\bar{\boldsymbol{f}}; \bar{\boldsymbol{X}})}{p(\boldsymbol{f}|\bar{\boldsymbol{f}}; \boldsymbol{X})q(\bar{\boldsymbol{f}})}d\boldsymbol{f}d\bar{\boldsymbol{f}} \\ &= \mathbb{E}_{q(\boldsymbol{f})}[\ln p(\boldsymbol{y}|\boldsymbol{f})] + \mathbb{E}_{q(\bar{\boldsymbol{f}})}\Big[\ln \frac{p(\bar{\boldsymbol{f}}; \bar{\boldsymbol{X}})}{q(\bar{\boldsymbol{f}})}\Big] \triangleq \mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \theta).\end{aligned} \tag{2.14}$$

For the regression task, the optimal variational distribution to maximize $\mathcal{L}_{\text{ELBO}}$ can be calculated analytically by the following theorem. A more detailed derivation can be found in Titsias [96].

Theorem 2.4.3 (Titsias [96])**.** *Assuming that the prior is a sparse Gaussian process and the data likelihood is* $p(\boldsymbol{y}) = \prod_{i=1}^{N} \mathcal{N}(0, \sigma^2)$*, the optimal variational distribution* $q^*(\bar{\boldsymbol{f}})$ *to maximize the ELBO is a Gaussian distribution* $q^*(\bar{\boldsymbol{f}}) = \mathcal{N}(\bar{\boldsymbol{f}}; \boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*)$*, where*

$$\boldsymbol{\mu}^* = \boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}\Big(\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} + \frac{1}{\sigma^2}\boldsymbol{K}_{\bar{\boldsymbol{X}}\boldsymbol{X}}\boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\Big)^{-1}\Big(\frac{1}{\sigma^2}\boldsymbol{K}_{\bar{\boldsymbol{X}}\boldsymbol{X}}\boldsymbol{y} + m_0\Big),$$
$$\boldsymbol{\Sigma}^* = \boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}\Big(\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} + \frac{1}{\sigma^2}\boldsymbol{K}_{\bar{\boldsymbol{X}}\boldsymbol{X}}\boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\Big)^{-1}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}.$$

Proof. We can write $\mathcal{L}_{\text{ELBO}}$ as a function of $q(\bar{\boldsymbol{f}})$.

$$\mathcal{L}_{\text{ELBO}}(q(\bar{\boldsymbol{f}})) = \int p(\boldsymbol{f}|\bar{\boldsymbol{f}})q(\bar{\boldsymbol{f}}) \ln p(\boldsymbol{y}|\boldsymbol{f})d\boldsymbol{f}d\bar{\boldsymbol{f}} + \int q(\bar{\boldsymbol{f}}) \ln \frac{p(\bar{\boldsymbol{f}})}{q(\bar{\boldsymbol{f}})}d\bar{\boldsymbol{f}}.$$

Taking the gradient with respect to the function $q(\bar{\boldsymbol{f}})$, we could obtain the optimal distribution $q^*(\bar{\boldsymbol{f}})$.

$$q^*(\bar{\boldsymbol{f}}) \propto p(\bar{\boldsymbol{f}}) \exp\Big(\int p(\boldsymbol{f}|\bar{\boldsymbol{f}}) \ln p(\boldsymbol{y}|\boldsymbol{f})d\boldsymbol{f}\Big). \tag{2.15}$$

In this equation, the integral in the exponential term can be computed as follows:

$$\begin{aligned}&\int p(\boldsymbol{f}|\bar{\boldsymbol{f}}) \ln p(\boldsymbol{y}|\boldsymbol{f})d\boldsymbol{f}\\ &= \ln[\mathcal{N}(\boldsymbol{y}; \boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\bar{\boldsymbol{f}}, \sigma^2 I)] - \frac{1}{2\sigma^2}\text{tr}(\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} - \boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{K}_{\bar{\boldsymbol{X}}\boldsymbol{X}})\end{aligned} \tag{2.16}$$

Let $Q = \boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}$. Then the optimal distribution $q^*(\bar{\boldsymbol{f}})$ is

$$\begin{aligned}&q^*(\bar{\boldsymbol{f}}) \propto p(\bar{\boldsymbol{f}})\mathcal{N}(\boldsymbol{y}; Q\bar{\boldsymbol{f}}, \sigma^2 I)\\ &\qquad = \exp\Big(-\frac{1}{2}(\bar{\boldsymbol{f}} - m_0)\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}(\bar{\boldsymbol{f}} - m_0) - \frac{1}{2\sigma^2}(Q\bar{\boldsymbol{f}} - \boldsymbol{y})^\top(Q\bar{\boldsymbol{f}} - \boldsymbol{y})\Big).\end{aligned}$$

We can recognize that this is also a Gaussian distribution with the mean vector $\boldsymbol{\mu}^*$ and the covariance matrix $\boldsymbol{\Sigma}^*$. □

Inserting the optimal variational distribution $q^*(\bar{\boldsymbol{f}})$ back into $\mathcal{L}_{\text{ELBO}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \theta)$, we obtain the final objective function $\mathcal{L}^*_{\text{ELBO}}$.

$$\begin{aligned}\mathcal{L}^*_{\text{ELBO}} &= \mathbb{E}_{q^*(\bar{\boldsymbol{f}})}[\mathbb{E}_{p(\boldsymbol{f}|\bar{\boldsymbol{f}})}[\ln p(\boldsymbol{y}|\boldsymbol{f})]] + \mathbb{E}_{q^*(\bar{\boldsymbol{f}})}\Big[\ln \frac{p(\bar{\boldsymbol{f}}; \bar{\boldsymbol{X}})}{q^*(\bar{\boldsymbol{f}})}\Big]\\ &= \ln\Big[\int p(\bar{\boldsymbol{f}}) \exp\Big(\int p(\boldsymbol{f}|\bar{\boldsymbol{f}}) \ln p(\boldsymbol{y}|\boldsymbol{f})d\boldsymbol{f}\Big)d\bar{\boldsymbol{f}}\Big]\\ &= \ln\Big[\mathbb{E}_{p(\bar{\boldsymbol{f}})}\mathcal{N}(\boldsymbol{y}; Q\bar{\boldsymbol{f}}, \sigma^2 I)\Big] - \frac{1}{2\sigma^2}\text{tr}(\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} - \boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{K}_{\bar{\boldsymbol{X}}\boldsymbol{X}})\\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{(Equation (2.16))}\\ &= \ln \mathcal{N}(\boldsymbol{y}; Q\boldsymbol{m}_0, Q\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}Q^\top + \sigma^2 I) - \frac{1}{2\sigma^2}\text{tr}(\boldsymbol{K}_{\boldsymbol{X}\boldsymbol{X}} - \boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{K}_{\bar{\boldsymbol{X}}\boldsymbol{X}}).\end{aligned}$$

The training of the hyper-parameters is conducted by maximizing $\mathcal{L}^*_{\text{ELBO}}$ with respect to the hyper-parameters $\boldsymbol{\theta}$.

For the computational complexity of using a sparse Gaussian process for the regression task, since $M < N$ the computational complexity when performing the inversion of the matrix is reduced to $O(M^3)$ while the computational complexity is $O(N^3)$ in the standard Gaussian process regression. The bottleneck now lies in the matrix-matrix multiplication $\boldsymbol{K}_{\boldsymbol{X}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}$ and this operation costs $O(NM^2)$. The final computational complexity is $O(NM^2)$.

2.4.3 Sparse Gaussian Processes for Non-conjugate Models

The closed-form solution of $q(\bar{\boldsymbol{f}})$ for the regression task relies on the form of the data likelihood, which is a Gaussian distribution and is conjugate to the GP prior. However, in other tasks, such as the classification [49, 50] or the intensity estimation [60], the data likelihood is no longer conjugate to the Gaussian process prior and the exact inference is no longer available.

When we are using a non-conjugate data likelihood, however, the ELBO $\mathcal{L}_{\text{ELBO}}$ in Equation (2.14) is still applicable.

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \boldsymbol{\theta}) &\triangleq \mathbb{E}_{q(f(\boldsymbol{x}))}[\ln p(\boldsymbol{y}|f(x))] + \mathbb{E}_{q(\bar{\boldsymbol{f}})}\Big[\ln \frac{p(\bar{\boldsymbol{f}}; \bar{\boldsymbol{X}})}{q(\bar{\boldsymbol{f}})}\Big], \\ q(\bar{\boldsymbol{f}}) &= \mathcal{N}(\bar{\boldsymbol{f}}; \boldsymbol{\mu}, \boldsymbol{\Sigma}).\end{aligned} \tag{2.17}$$

The first part in $\mathcal{L}_{\text{ELBO}}$ depends on the form of the data likelihood. For example, in the binary classification task [49], each $y_i \in \{0, 1\}$ and we assume the data likelihood $p(y_i = 1|f(\boldsymbol{x}_i)) = 1/(1 + \exp(-f(\boldsymbol{x}_i)))$. The first term can be decomposed into the sum of N terms by the independence assumption.

$$\mathbb{E}_{q(f(\boldsymbol{x}))}[\ln p(\boldsymbol{y}|f(x))] = \sum_{i=1}^{N} \mathbb{E}_{q(f(\boldsymbol{x}))}[\ln p(y_i|f(\boldsymbol{x}_i))] = \sum_{i=1}^{N} \mathbb{E}_{q(f(\boldsymbol{x}_i))}[\ln p(y_i|\boldsymbol{f}; \boldsymbol{x}_i)].$$

The last equation comes from the fact that the marginal of a Gaussian distribution is still a Gaussian distribution. The variational distribution $q(f(x))$ can be computed similarly to Equation (2.13) and the posterior $q(f(\boldsymbol{x}))$ is a Gaussian process $\mathcal{GP}(\bar{m}(\boldsymbol{x}), \bar{\kappa}(\boldsymbol{x}, \boldsymbol{x}'))$.

$$\begin{aligned}\bar{m}(\boldsymbol{x}) &= m_0 + \boldsymbol{\kappa}_{\boldsymbol{x}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}(\boldsymbol{\mu} - m_0), \\ \bar{\kappa}(\boldsymbol{x}, \boldsymbol{x}') &= \kappa_{\boldsymbol{x}\boldsymbol{x}'} - \boldsymbol{\kappa}_{\boldsymbol{x}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\kappa}_{\boldsymbol{x}'\bar{\boldsymbol{X}}}^{\top} + \boldsymbol{\kappa}_{\boldsymbol{x}\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\Sigma}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\kappa}_{\boldsymbol{x}'\bar{\boldsymbol{X}}}.\end{aligned}$$

The second part in $\mathcal{L}_{\text{ELBO}}$ is the negative Kullback-Leibler divergence between two Gaussian distributions.

$$\begin{aligned}&\mathbb{E}_{q(\bar{\boldsymbol{f}})}\Big[\ln \frac{p(\bar{\boldsymbol{f}}; \bar{\boldsymbol{X}})}{q(\bar{\boldsymbol{f}})}\Big] = -\text{KL}(q(\bar{\boldsymbol{f}})||p(\bar{\boldsymbol{f}}; \bar{\boldsymbol{X}})) \\ &= -\text{KL}(\mathcal{N}(\bar{\boldsymbol{f}}; \boldsymbol{\mu}, \boldsymbol{\Sigma})||\mathcal{N}(\bar{\boldsymbol{f}}; \boldsymbol{0}, \boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}})) \\ &= \frac{1}{2}\ln \frac{|\boldsymbol{\Sigma}|}{|\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}|} + \frac{1}{2}\mathbb{E}_{q(\bar{\boldsymbol{f}})}\Big[(\bar{\boldsymbol{f}} - \boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\bar{\boldsymbol{f}} - \boldsymbol{\mu}) - (\bar{\boldsymbol{f}} - m_0)^{\top}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}(\bar{\boldsymbol{f}} - m_0)\Big] \\ &= \frac{1}{2}\ln \frac{|\boldsymbol{\Sigma}|}{|\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}|} + \frac{M}{2} - \frac{1}{2}\text{tr}\Big(\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}(\boldsymbol{\Sigma} + (\boldsymbol{\mu} - m_0)(\boldsymbol{\mu} - m_0)^{\top}\Big).\end{aligned} \tag{2.18}$$

The training process when we have a non-conjugate data likelihood can be conducted by two different approaches. In the first approach, we use the grid-search [51] or the Bayesian optimization [89] to choose the hyper-parameters $\boldsymbol{\theta}$.

When the hyper-parameters $\boldsymbol{\theta}$ are all fixed, we solve the following optimization problem.

$$(\boldsymbol{\mu}^*, \boldsymbol{\Sigma}^*) = \arg\max_{\boldsymbol{\mu}, \boldsymbol{\Sigma}} \mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \boldsymbol{\theta}).$$

The second approach is the variational Bayesian expectation-maximization (VB-EM) framework [7] which is given in Algorithm 3. This can be seen as an algorithm to alternately maximize $\mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \boldsymbol{\theta})$ with respect to $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ and $\boldsymbol{\theta}$. We use the expectation-maximization since some parameters in the variational distribution can have a closed-form update. Although we avoid the grid-search in the training process, we may suffer from the bias since the optimal hyper-parameters should be found by maximizing the marginal likelihood. However, in each M-step we are maximizing the ELBO and ELBO is an inexact lower bound of the marginal data likelihood [97].

Algorithm 3: The VB-EM framework for the sparse Gaussian process with a non-conjugate model.

Input : The training data set $\mathcal{D}$.
Output: An estimation of the parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ in the variational distribution $q(\bar{\boldsymbol{f}})$ and the hyper-parameters $\boldsymbol{\theta}$.

Initialize $k = 0, x^{(k)} = \mathrm{Inf}$.
Initialize the parameters $\boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)}$ and the hyper-parameters $\boldsymbol{\theta}^{(k)}$.
while *True* **do**
    $k = k + 1$.
    E-step: Update $\boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)}$ to increase $\mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \boldsymbol{\theta}^{(k-1)})$.
    M-step: Update $\boldsymbol{\theta}^{(k)}$ to increase $\mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)}; \boldsymbol{\theta})$.
    Calculate the current $x^{(k)} = \mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{\mu}^{(k)}, \boldsymbol{\Sigma}^{(k)}; \boldsymbol{\theta}^{(k)})$.
    if $|x^{(k)} - x^{(k-1)}| < 10^{-6}|x^{(k)}|$ **then**
        Break.
    end
end

2.5 Temporal Point Processes

A temporal point pattern [78] is ubiquitous in everyday life. Let us assume that you arrive at a bus stop and begin to examine the arrival time of the bus. The arrival time of a specific bus has a pattern, for instance, arriving almost every 15 minutes and may be subject to the delay on the road. This is a temporal point pattern. Some other examples are the occurrence time of vomit symptoms when a patient suffers from the nausea and the occurrence time of the earthquake occurs in Japan.

2.5.1 Three Views of a Temporal Point Process

A temporal point process, or an arrival process [31] can be specified by three different views, which are the arrival time-stamps of events, the inter-arrival time of two consecutive events and the number of events before a given time. Hereafter, we use the name event to indicate the occurrence of an incidence and it is different from the event in the probability space. First we give the definition of a temporal point process based on the arrival times of events $\{X_i\}$.

Definition 2.5.1. *A* temporal point process *is a sequence of random variables* $\{X_i\}_{i=1}^{\infty}$ *with the following property:*

$$0 < X_i < X_j,\ \forall i < j.$$

Notice that this is a very informal definition and a more formal one can be found in Daley and Vere-Jones [16] and Kingman [52]. A temporal point process will start at the time 0 and due to the increasing constraint on the random variables, multiple events will not occur simultaneously. If we would like to study the phenomenon of multiple arrivals at the same time, we could associate an additional random variable to each arrival time to model the number of the arrivals Daley and Vere-Jones [16].

The second way to specify a temporal point process is by the sequence of random variables $\{Z_i\}$ which represent the inter-arrival times of any two consecutive events by the following equation:

$$Z_1 = X_1,\ Z_i = X_i - X_{i-1},\ i \geq 1. \tag{2.19}$$

The third way to specify a temporal point process is by a counting process $N(t), t \in \mathbb{R}_{>0}$. A counting process is a right-continuous function defined by the following equation:

$$N(t) = \#\{X_i \in (0, t]\}, t \in \mathbb{R}_{>0}.$$

Here # denotes the number of elements in the set and $\mathbb{R}_{>0}$ is the space of all positive real numbers. $N(0)$ is usually defined to be 0 which indicates that there is no event when the experiment starts. Finally, we can define the intensity function by the counting process.

Definition 2.5.2. *The intensity function* $\lambda(t)$ *is defined as the limit of the rate of events at a given time.*

$$\lambda(t) \triangleq \lim_{\Delta t \to 0^+} \mathbb{E}\Big[\frac{N(t + \Delta t) - N(t)}{\Delta t}\Big|\mathcal{H}_t\Big]. \tag{2.20}$$

Here $\mathcal{H}_t \triangleq \{X_i \in (0, t)\}$ *is the history of the events before time* t*.*

These three different views are closely related and we could obtain the other two views from any single view. We illustrate the relationship among these three views in Figure 2.5.

2.5.2 Homogeneous Poisson Processes

A Poisson process [52, 31] is a special case of the temporal point process and is also widely used in the various disciplines. Since there are three different views of a temporal point process, we can define a Poisson process in three different ways. Here we use the definition from the view of inter-arrival random variables.

Definition 2.5.3. *Let* λ *be a positive real number. A* homogeneous Poisson process *is a temporal point process in which the inter-arrival random variables* $\{Z_i\}$ *are i.i.d. exponential random variables with the parameter* λ*, that is*

$$p(Z_i = z; \lambda) = \lambda \exp(-\lambda z),\ \lambda, z \geq 0. \tag{2.21}$$

Based on Definition 2.5.3, we can study the distribution of the arrival time X_i and the counting process $N(t)$. First we introduce the following lemma.

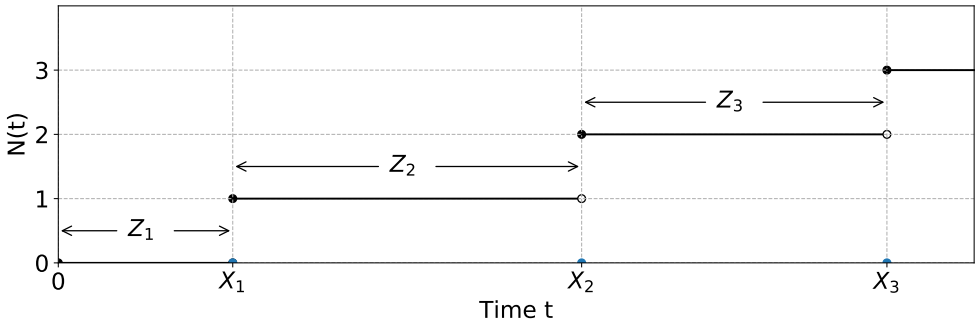

Figure 2.5: The illustration of the three views of a temporal point process. A realization of the arrival times $\{X_i\}$ are marked with blue points in the horizontal axis. A realization of the inter-arrival times $\{Z_i\}$ are marked with the lengths of the double arrows. A realization of the counting process is shown by the right-continuous function with a black point indicating that the function takes the corresponding value.

Lemma 2.5.1. *Let $\{Z_i\}$ be i.i.d. exponential-distributed random variables in Equation (2.21). Then $X_k = \sum_{i=1}^{k} Z_j$ and Z_{k+1} are independent random variables.*

Proof. First we use the law of total probability on the joint distribution.

$$\begin{aligned}
&P(X_k = a, Z_{k+1} = b) \\
&= \int_{z_1+\cdots+z_k=a} P(Z_1 = z_1, \ldots, Z_k = z_k, Z_{k+1} = b) dz_1 \cdots dz_k.
\end{aligned}$$

Since $\{Z_i\}$ are independent random variables, we have the decomposition.

$$\begin{aligned}
&P(X_k = a, Z_{k+1} = b) \\
=&\Big(\int_{z_1+\cdots+z_k=a} P(Z_1 = z_1, \ldots, Z_k = z_k) dz_1 \ldots dz_k \Big) P(Z_{k+1} = b) \\
=&P(X_k = a) P(Z_{k+1} = b).
\end{aligned}$$

□

This lemma indicates that for a homogeneous Poisson process, the previous arrival time will not affect the arrival of the next point. The distributions of X_i and $N(t)$ are given by the following two theorems.

Theorem 2.5.1 (Gallager [31])**.** *Let $\{Z_i\}$ be i.i.d. exponential random variables in Equation (2.21). The distribution of $X_i = \sum_{j=1}^{i} Z_j$ is an Erlang distribution.*

$$P(X_i = x; \lambda) = \frac{\lambda^i x^{i-1} \exp(-\lambda x)}{(i-1)!}, x \in \mathbb{R}_{>0}. \tag{2.22}$$

Proof. We use the mathematical induction method to prove this. For $i = 1$,

$$P(X_1 = Z_1 = x) = \lambda \exp(-\lambda x).$$

Since the exponential distribution is the Erlang distribution when $i = 1$, the proposition holds true. Next assuming that the theorem is true when $i = k$, since $X_{k+1} = X_k + Z_{k+1}$, we have

$$\begin{aligned}
P(X_{k+1} = x) &= \int_0^x P(X_k = s, Z_k = x - s)ds \quad \text{(The law of total probability)} \\
&= \int_0^x P(X_k = s)P(Z_k = x - s)ds \quad \text{(Lemma 2.5.1)} \\
&= \int_0^x \frac{\lambda^k s^{k-1} \exp(-\lambda s)}{(k-1)!} \lambda \exp(-\lambda(x-s))ds \\
&= \frac{\lambda^{k+1} \exp(-\lambda x)}{(k-1)!} \int_0^x s^{k-1} ds \\
&= \frac{\lambda^{k+1} \exp(-\lambda x)}{(k-1)!} \frac{x^k}{k} = \frac{\lambda^{k+1} x^k \exp(-\lambda x)}{k!}.
\end{aligned}$$

This means that the theorem holds true when $i = k + 1$ and according to the mathematical induction method, the theorem is proved. □

Theorem 2.5.2 (Gallager [31])**.** *Let $\{Z_i\}$ be i.i.d. exponential random variables in Equation* (2.21) *and $X_i = \sum_{j=1}^{i} Z_j$. The distribution of each random variable $N(t) = \#\{X_i, X_i \in (0, t]\}, t \in \mathbb{R}_{>0}$ is a Poisson distribution.*

$$P(N(t) = n; \lambda) = \frac{\lambda^n t^n \exp(-\lambda t)}{n!}, n \in \mathbb{N}^+. \tag{2.23}$$

Proof. We can represent the event $\{N(t) = n\}$ with X_n and Z_{n+1} using the law of total probability and then calculate the probability.

$$\begin{aligned}
P(N(t) = n) &= \int_0^t P(X_n = s, Z_{n+1} > t - s)ds \quad \text{(The law of total probability)} \\
&= \int_0^t \int_{t-s}^{\infty} P(X_n = s)P(Z_{n+1} = r)drds \quad \text{(Lemma 2.5.1)} \\
&= \int_0^t \frac{\lambda^n s^{n-1} \exp(-\lambda s)}{(n-1)!} \exp(-\lambda(t-s))ds \quad \text{(Theorem 2.5.1)} \\
&= \frac{\lambda^n \exp(-\lambda t)}{(n-1)!} \int_0^t s^{n-1} ds = \frac{\lambda^n t^n \exp(-\lambda t)}{n!}.
\end{aligned}$$

□

Utilizing the distribution of $N(t)$, we can calculate the intensity function $\lambda(t)$ of a homogeneous Poisson process. First we review the memoryless property of the exponential distribution.

Lemma 2.5.2 (Memoryless)**.** *Let Z_i be an exponential-distributed random variable in Equation* (2.21)*. Then*

$$P(Z_i > t + s | Z_i > t) = P(Z_i > s). \tag{2.24}$$

Proof.

$$\begin{aligned}
P(Z_i > t + s | Z_i > t) &= \frac{P(Z_i > t + s, Z_i > t)}{P(Z_i > t)} \\
&= \frac{\exp(-\lambda(t+s))}{\exp(-\lambda(t))} = \exp(-\lambda s) = P(Z_i > s).
\end{aligned}$$

□

One metaphor for this lemma is given as follows. Let us assume that the arrival time of a bus follows a Poisson process. This lemma tells us that if a person has waited at a bus stop for one hour, this one hour will not change the expected time he/she has to wait till the bus comes.

Corollary 2.5.1. *Let $\{Z_i\}$ be i.i.d. exponential random variables in Equation (2.21). $X_i = \sum_{j=1}^{i} Z_j$ and $N(t) = \#\{X_i, X_i \in (0, t]\}, t \in \mathbb{R}_{>0}$.*

$$P(N(t+s) - N(s) = n | N(s) = m) = P(N(t) = n), n \in \mathbb{N}^+. \tag{2.25}$$

Proof. Let us consider the arrival time of the first event after time s,

$$\begin{aligned} P(X_{m+1} > r + s | N(s) = m) &= P(Z_{m+1} > r + s - u | N(s) = m, X_m = u) \\ &= P(Z_{m+1} > r + s - u | Z_{m+1} > s - u, X_m = u) \\ &= P(Z_{m+1} > r). \qquad \text{(Lemma 2.5.2)} \end{aligned}$$

This implies that the distribution of the first event after time s is still the exponential distribution in Equation (2.21). Reusing Lemma 2.5.2, we may conclude that the all events after time s follows the same pattern as if the process started from 0. □

Corollary 2.5.2. *The intensity function of a homogeneous Poisson process is a constant function $\lambda(t) \equiv \lambda$.*

Proof. First we recall the distribution of $N(t + \Delta t) - N(t)$ from Lemma 2.5.1 as follows:

$$P(N(t + \Delta t) - N(t) = n) = \frac{(\lambda \Delta t)^n \exp(-\lambda \Delta t)}{n!}.$$

Then the intensity function can be computed according to Equation (2.20).

$$\begin{aligned} \lambda(t) &\overset{\Delta}{=} \lim_{\Delta t \to 0^+} \mathbb{E}\Big[\frac{N(t + \Delta t) - N(t)}{\Delta t} \Big| \mathcal{H}_t\Big] \\ &= \lim_{\Delta t \to 0^+} \sum_{n=0}^{\infty} n \frac{(\lambda \Delta t)^n \exp(-\lambda \Delta t)}{n! \Delta t} = \lim_{\Delta t \to 0^+} \frac{\exp(-\lambda \Delta t)}{\Delta t} \Big(\sum_{n=0}^{\infty} n \frac{a^n}{n!} \Big) \Big|_{a = \lambda \Delta t} \\ &= \lim_{\Delta t \to 0^+} \frac{\exp(-\lambda \Delta t)}{\Delta t} \exp(\lambda \Delta t) \lambda \Delta t = \lambda, \end{aligned}$$

where we used the sum $\sum_{n=0}^{\infty} n a^n / n! = a \exp(a)$. □

2.5.3 Inhomogeneous Poisson Processes

As we can see from the intensity function, homogeneous Poisson processes are restrictive when we are trying to describe the complicated time-sequences in the real-world. One way to generalize the concept of homogeneous Poisson processes is to allow the intensity function to be an arbitrary function. The resulting process is called an inhomogeneous Poisson process [31]. We define the inhomogeneous Poisson process with inter-arrival times to compare this concept with the homogeneous Poisson process.

Definition 2.5.4. *Let $\mu : \mathbb{R}_{>0} \to \mathbb{R}_{\geq 0}$ be a non-negative function. An* inhomogeneous Poisson process *is a temporal point process with the inter-arrival random variables $\{Z_i\}$. The distribution of each random variable Z_i is given as follow:*

$$P(Z_i = z | X_{i-1} = x_{i-1}; \mu) = \mu(x_{i-1} + z) \exp\Big(- \int_{x_{i-1}}^{x_{i-1}+z} \mu(a) da \Big), \tag{2.26}$$

where $X_i = \sum_{j=1}^{i} Z_i$, $X_0 = 0$ and an observation of $\{X_i\}$ is $\{x_i\}$.

The cumulative distribution function for Z_i can be computed with integration.

$$P(Z_i \leq z | X_{i-1} = x_{i-1}; \mu) = 1 - \exp\Big(- \int_{x_{i-1}}^{x_{i-1}+z} \mu(a) da \Big). \tag{2.27}$$

Later we will prove that the intensity function of an inhomogeneous Poisson process is exactly $\mu(t)$. For the time being, we call the function $\mu(t)$ the intensity function. First we derive the data likelihood when observing the points $\{x_i\}_{i=1}^n$ from an inhomogeneous Poisson process on the observation window $(0, T]$.

Theorem 2.5.3 (Gallager [31])**.** *Let $\{Z_i\}$ be inter-arrival random variables for an inhomogeneous Poisson process with the intensity function $\mu(t)$. Let $X_i = \sum_{j=1}^{i} Z_j$. The probability for the event $\{0 < X_1 = x_1 < \cdots < X_n = x_n \leq T, x_{n+1} > T\}$ is*

$$P(0 < X_1 = x_1 < \cdots < X_n = x_n \leq T, x_{n+1} > T) = \exp\Big(-\int_0^T \mu(a) da\Big) \prod_{i=1}^{n} \mu(x_i). \tag{2.28}$$

Proof. Since in Equation (2.26), $X_i = Z_i + X_{i-1}$, we have

$$P(Z_i = z | X_{i-1} = x_{i-1}; \mu) = P(X_i = z + x_{i-1} | X_{i-1} = x_{i-1}; \mu).$$

The probability can be computed as follows.

$$\begin{aligned}
&P(0 < X_1 = x_1 < \cdots < X_n = x_n \leq T, x_{n+1} > T) \\
=&P(X_1 = x_1) P(X_{n+1} > T | X_n = x_n) \prod_{i=2}^{n} P(X_i = x_i | X_{i-1} = x_{i-1}) \\
=&P(Z_1 = x_1) P(Z_{n+1} > T - x_n | X_n = x_n) \prod_{i=2}^{n} P(Z_i = x_i - x_{i-1} | X_{i-1} = x_{i-1}) \\
=&\exp\Big(- \int_0^T \mu(a) da \Big) \prod_{i=1}^{n} \mu(x_i).
\end{aligned}$$

□

The distribution of $N(t)$ in a homogeneous Poisson process is a Poisson distribution. Similarly, the distribution of $N(t)$ is also a Poisson distribution and we can prove this with Theorem 2.5.3.

Corollary 2.5.3. *Let $\{Z_i\}$ be inter-arrival random variables for an inhomogeneous Poisson process with the intensity function $\mu(t)$. $X_i = \sum_{j=1}^{i} Z_j$. The distribution of $N(t) = \#\{X_i, X_i \in (0, t]\}, t \in \mathbb{R}_{>0}$ is a Poisson distribution.*

$$P(N(t) = n; \mu) = \frac{1}{n!} \Big(\int_0^t \mu(a) da \Big)^n \exp\Big(- \int_0^t \mu(a) da \Big), n \in \mathbb{N}^+. \tag{2.29}$$

Proof. We can compute the probability directly with integration.

$$\begin{aligned}
&P(N(t) = n; \mu) \\
&= \int_{0 < x_1 < \cdots < x_n \leq t} P(X_1 = x_1, \ldots, X_n = x_n, X_{n+1} > t) dx_1 \cdots dx_n \\
&\qquad\qquad \text{(The law of total probability.)} \\
&= \exp\Big(- \int_0^t \mu(a) da \Big) \int_{0 < x_1 < \cdots < x_n \leq t} \prod_{i=1}^{n} \mu(x_i) dx_1 \cdots dx_n \qquad \text{(Theorem 2.5.3)} \\
&= \frac{1}{n!} \Big(\int_0^t \mu(a) da \Big)^n \exp\Big(- \int_0^t \mu(a) da \Big).
\end{aligned}$$

In the last step, we use the integral result which can be obtained from a straightforward computation.

$$\int_{0<x_1<\cdots<x_{k+1}\leq t}\prod_{i=1}^{n}\mu(x_i)dx_1\cdots dx_n=\frac{1}{n!}\Big(\int_0^t\mu(a)da\Big)^n \tag{2.30}$$

□

Before calculating the intensity function in an inhomogeneous Poisson process, we first examine the distribution of the number of events in an arbitrary observation window $(s, s+t]$.

Corollary 2.5.4. *Let $\{Z_i\}$ be random variables with distributions described in Equation (2.26) and $X_i=\sum_{j=1}^{i}Z_j$. Let $N(t)=\#\{X_i, X_i\in(0,t]\}, t\in\mathbb{R}_{>0}$. The distribution of $N(t+s)-N(s)$ is also a Poisson distribution.*

$$P(N(t+s)-N(s)=n;\mu)=\frac{1}{n!}\Big(\int_s^{t+s}\mu(a)da\Big)^n\exp\Big(-\int_s^{t+s}\mu(a)da\Big), n\in\mathbb{N}^+.$$

Proof. Notice that Equation (2.26) is not an exponential distribution now. This implies that the memoryless property does not hold for an inhomogeneous Poisson process. However, we can still compute the probability.

$$\begin{aligned}P(Z_i>a+b|Z_i>a, X_{i-1}=x_{i-1};\mu)&=\frac{P(Z_i>a+b, Z_i>a|X_{i-1}=x_{i-1};\mu)}{P(Z_i>a|X_{i-1}=x_{i-1};\mu)}\\&=\exp\Big(-\int_a^{a+b}\mu(u)du\Big).\end{aligned} \tag{2.31}$$

We will examine the distribution of the first event. Let $N(s)=m$ and then the probability that next event X_{m+1} occurs is:

$$P(X_{m+1}>r+s|N(s)=m, X_{m+1}>s)=\exp\Big(-\int_s^{s+r}\mu(u)du\Big).$$

Notice that the distribution does not depend on the information when X_m occurs and what $N(s)$ is. Reusing Equation (2.26), we can obtain the distribution of $N(t+s)-N(s)$.

$$P(N(t+s)-N(s)=n;\mu)=P(s\leq Z_{m+1}<\cdots<Z_{m+n}\leq t+s, Z_{m+n+1}>t+s).$$

The remaining computation is similar to that in Corollary 2.5.3. □

Based on Corollary 2.5.4, we can obtain the intensity function of the inhomogeneous Poisson process from the definition.

Corollary 2.5.5. *The intensity function of an inhomogeneous Poisson process is $\lambda(t)=\mu(t)$.*

Proof. Using Corollary 2.5.4, the distribution of the number of events $N(t+\Delta t)-N(t)$ is a Poisson-distributed random variable and

$$\mathbb{E}[N(t+\Delta t)-N(t)]=\int_t^{t+\Delta t}\mu(a)da.$$

The intensity function can be computed from Equation (2.20).

$$\lambda(t)\triangleq\lim_{\Delta t\to 0^+}\mathbb{E}\Big[\frac{N(t+\Delta t)-N(t)}{\Delta t}\Big|\mathcal{H}_t\Big]=\lim_{\Delta t\to 0^+}\frac{1}{\Delta t}\int_t^{t+\Delta t}\mu(a)da=\mu(t).$$

□

2.5.4 Sampling Algorithms for Poisson Processes

The simulation of a homogeneous Poisson process can be obtained directly from Definition 2.5.3. The basic idea is to sample each random variable Z_i independently. The sampling algorithm [83] is given in Algorithm 4.

Algorithm 4: The sampling algorithm for a homogeneous Poisson process.

Input : The hyper-parameter λ and a time window $(0, T]$.
Output: A sequence of arrival times $\{x_i\}$.

1 Initialize $k = 1, x_0 = 0$.
2 **while** *True* **do**
3 Sample $w \sim \text{Uniform}[0, 1]$.
4 Set $z = -(\ln w)/\lambda$.
5 **if** $x_{k-1} + z < T$ **then**
6 Set $x_k = x_{k-1} + z$. Set $k = k + 1$.
7 **else**
8 Return $\{x_i\}_{i=1}^{k}$.
9 **end**
10 **end**

In Algorithm 4, the inverse sampling [82] is used in Steps 3 and 4 to sample an exponential-distributed random variable Z. The inverse sampling can be verified by the following equation:

$$P(Z \leq z) = P(-\ln(W) \leq \lambda z) = P(W \geq \exp(-\lambda z)) = 1 - \exp(-\lambda z).$$

For an inhomogeneous Poisson process, the thinning algorithm [55, 72] can be used to sample the arrival times $\{x_i\}$. The sampling algorithm is provided in Algorithm 5.

Algorithm 5: The thinning algorithm for sampling an inhomogeneous Poisson process.

Input : The intensity function $\lambda(t)$ and a time window $(0, T]$.
Output: A sequence of arrival times $\{x_k\}$.

1 Initialize $k = 1, t = 0$.
2 Compute $\hat{\lambda} = \max_{s \in [0,T]} \lambda(s)$.
3 **while** *True* **do**
4 Sample two random variables $W, V \sim \text{Uniform}[0, 1]$.
5 Set $z = -(\ln w)/\hat{\lambda}$.
6 **if** $t + z < T$ **then**
7 **if** $v\hat{\lambda} < \lambda(t + z)$ **then**
8 Set $x_k = t + z$. Set $k = k + 1$.
9 **else**
10 Reject z.
11 **end**
12 Set $t = t + z$.
13 **else**
14 Return $\{x_i\}_{i=1}^{k}$.
15 **end**
16 **end**

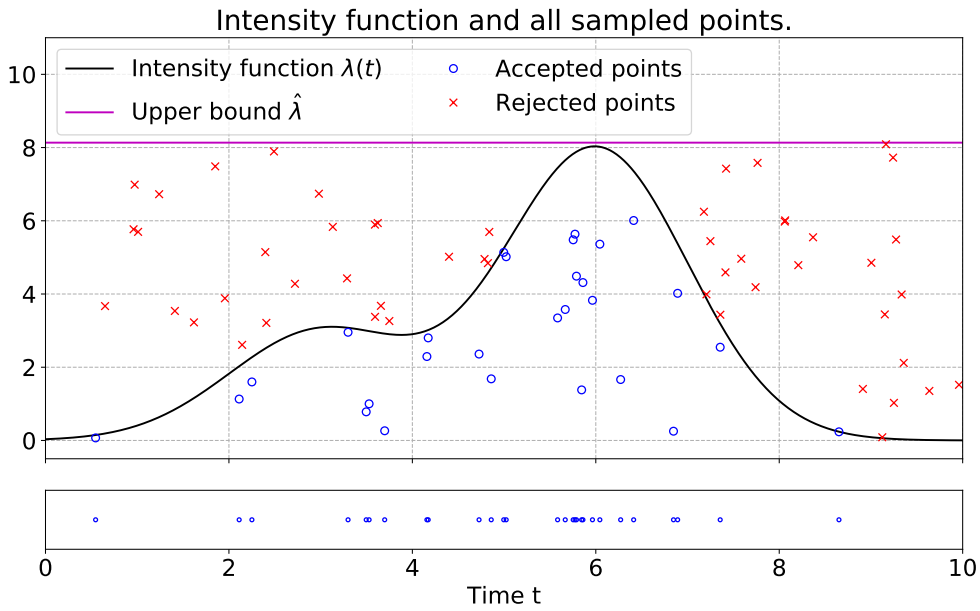

Figure 2.6: The illustration of the sampling process of the thinning algorithm. The intensity function is given in the top figure along with the accepted and rejected points, which are denoted with "$\circ$" and "$\times$" respectively. The vertical coordinate of each point corresponds to $v\hat{\lambda}$. Only points below the intensity function are accepted. The final sampled sequence is given in the bottom figure.

An illustration of one sampling algorithm is given in Figure 2.6. Each point in the figure is denoted with "$\circ$" or "$\times$". The vertical coordinate of each point corresponds to $v\hat{\lambda}$ in Algorithm 5. We can notice that the accepted points fall below the curve of the intensity function and the rejected points fall above the curve.

To prove that using Algorithm 5 we indeed sample an inhomogeneous Poisson process with the intensity function $\lambda(t)$, first we examine the distribution of the random variable X_1, which represents the arrival time of the first event.

Theorem 2.5.4 (Lewis and Shedler [55])**.** *The distribution of the arrival time of the first event is as follows:*

$$P(X_1 = x_1) = \lambda(x_1) \exp\Big(- \int_0^{x_1} \lambda(a) da \Big), \ x_1 \geq 0.$$

Proof. Before accepting the first event, there may be multiple rejections in Step 10. Let the rejected times be $\{r_i\}$ and $r_0 = 0$. The probability that the arrival

time of the first event is accepted after k rejections $\{r_i\}_{i=1}^k$ is as follows:

$$
\begin{aligned}
&P(X_1 = x_1, \{r_i\}_{i=1}^k \text{ is rejected}) \\
=&\int_{0<r_1<\cdots<r_k<x_1} \prod_{i=1}^{k} \Big[\hat{\lambda}\exp(-\hat{\lambda}(r_i - r_{i-1}))\Big(1 - \frac{\lambda(r_i)}{\hat{\lambda}}\Big)\Big] \qquad (r_0 = 0) \\
&\times \hat{\lambda}\exp(-\hat{\lambda}(x_1 - r_k))\frac{\lambda(x_1)}{\hat{\lambda}} dr_1 \cdots dr_k \\
=&\lambda(x_1)\exp(-\hat{\lambda}x_1)\int_{0<r_1<\cdots<r_k<x_1} \prod_{i=1}^{k} \Big[\hat{\lambda}\Big(1 - \frac{\lambda(r_i)}{\hat{\lambda}}\Big)\Big] dr_1 \cdots dr_k \\
=&\lambda(x_1)\exp(-\hat{\lambda}x_1)\frac{\Big(\int_0^{x_1} \hat{\lambda}\Big(1 - \frac{\lambda(a)}{\hat{\lambda}}\Big)da\Big)^k}{k!}. \qquad \text{(Equation (2.30))}
\end{aligned}
$$

The probability of $\{X_1 = x_1\}$ can be represented by the law of total probability.

$$
\begin{aligned}
P(X_1 = x_1) &= \sum_{k=0}^{\infty} P(X_1 = x_1, \{r_i\}_{i=1}^k \text{ is rejected}) \\
&= \lambda(x_1)\exp(-\hat{\lambda}x_1)\sum_{k=0}^{\infty}\frac{\Big(\int_0^{x_1} \hat{\lambda}\Big(1 - \frac{\lambda(a)}{\hat{\lambda}}\Big)da\Big)^k}{k!} \\
&= \lambda(x_1)\exp(-\hat{\lambda}x_1)\exp\Big(\int_0^{x_1} \hat{\lambda}\Big(1 - \frac{\lambda(a)}{\hat{\lambda}}\Big)da\Big) \\
&= \lambda(x_1)\exp\Big(-\int_0^{x_1} \lambda(a)da\Big).
\end{aligned}
$$

□

After accepting the first sample, the thinning algorithm restarts. We can conclude that the following events will obey the same pattern described in Equation (2.26) in a similar way. This implies that the sample we obtain from the thinning algorithm is truly a realization of the inhomogeneous Poisson process with the intensity function $\lambda(t)$.

2.6 The Intensity Estimation for Recurrent Event Data

In this section, we briefly review the previous studies on the estimation of the intensity function from the time-sequence data. The first assumption we make is that the time-sequence is drawn from an inhomogeneous Poisson process.

2.6.1 Estimation of the Mean Intensity Function

When having a data set of time-sequences from K subjects $\mathcal{D} = \{\boldsymbol{d}_k\}_{k=1}^K$ on $\mathcal{X}_k = [0, T]$, we can assign an independent intensity function to each of the subjects $\{\lambda_k(x)\}_{k=1}^K$ and estimate each intensity function separately. However, this is not efficient since the memory cost of this approach is $O(K)$. Moreover, the performance of the estimate for the time-sequence with very few arrival times will be very poor.

The most naive way to reduce the memory cost and share the statistical strength is to assume that all time-sequences are generated by the same inhomogeneous Poisson process with intensity $\lambda(x)$. Based on this assumption, the

logarithm of the data likelihood can be computed using Theorem 2.5.3.

$$\ln p(\mathcal{D}|\lambda(x)) = \ln \prod_{k=1}^{K} p(\boldsymbol{d}_k|\lambda(x)) \\ = \sum_{k=1}^{K}\sum_{j=1}^{N_k} \ln \lambda(x_j^{(k)}) + K\Big(- \int_0^T \lambda(x)dx\Big). \tag{2.32}$$

Based on the logarithm of the likelihood in Equation (2.32), we can prove the following theorem.

Theorem 2.6.1 (Cook and Lawless [15])**.** *When all observations are the same $\mathcal{X}_k = \mathcal{X}, \forall k = 1, \ldots, K$, the estimate $\hat{\lambda}(x)$ which we obtain by maximizing Equation* (2.32) *is an unbiased estimator of the mean of all intensity functions $\{\lambda_k(x)\}$.*

$$\mathbb{E}[\hat{\lambda}(x)] = \frac{1}{K}\sum_{k=1}^{K} \lambda_k(x). \tag{2.33}$$

Proof. Let the counting processes be $\{N_k(x)\}_{k=1}^K$. Correspondingly, we have the number of events in $[s, s+ds)$ which we denote as $\{dN_k(x)\}_{k=1}^K$ [15]. Using the definition of the intensity function, we have

$$\mathbb{E}[dN_k(x)] = \lambda_k(x)dx.$$

The logarithm of the likelihood can be rewritten as

$$\ln p(\mathcal{D}|\lambda(x)) = \sum_{k=1}^{K}\sum_{j=1}^{N_k} \ln \lambda(x_j^{(k)}) + K\Big(- \int_{\mathcal{X}} \lambda(x)dx\Big) \\ = \sum_{k=1}^{K} \int_{\mathcal{X}} \ln \lambda(x) dN_k(x) - K\int_{\mathcal{X}} \lambda(x)dx.$$

Taking the derivative with respect to $\lambda(x)$ and setting it to zero, we obtain that

$$\hat{\lambda}(x)dx = \frac{1}{K}\sum_{k=1}^{K} dN_k(x).$$

The expectation of the estimate $\hat{\lambda}$ found by maximizing this data likelihood is

$$\mathbb{E}[\hat{\lambda}(x)dx] = \mathbb{E}\Big(\frac{1}{K}\sum_{k=1}^{K} dN_k(x)\Big) = \frac{1}{K}\sum_{k=1}^{K} \lambda_k(x)dx.$$

□

To ease our notations, we review the estimation method of the intensity function assuming that we only have a single time sequence $K = 1$ in Sections 2.6.2 and 2.6.3. We can also use these methods to estimate the mean intensity function. However, only estimating the mean of the intensity functions fails to model the diversity among K time-sequences. We will review the studies on how to model the diversity among K time-sequences in Section 2.6.4.

2.6.2 Point Estimates of the Intensity Function

Previous studies on the point estimates for the intensity functions assume that the true intensity function is fixed before the sampling process. Here we introduce the kernel smoothing method [18] and the local likelihood method [103, 9, 43].

For the time-sequence in the recurrent event data, we first consider the situation where the number of subjects $K = 1$ and the data $\boldsymbol{d} = \{x_j \in \mathcal{X}\}_{j=1}^N$.

Kernel Smoothing Method

The estimate by the kernel smoothing method [18] is given by the following equation.

$$\hat{\lambda}(x) = \sum_{i=1}^{N} \kappa_h(x - x_j) = \frac{1}{h}\sum_{i=1}^{N} \kappa\Big(\frac{x - x_j}{h}\Big),\ h > 0, x \in \mathcal{X},$$

where h is a positive number termed the bandwidth in the transformed kernel function $\kappa_h(x)$. The kernel function $\kappa(\cdot)$ satisfies the following property:

$$\kappa(x) \geq 0,\ \int_{-\infty}^{\infty} \kappa(x)dx = 1,\ \int_{-\infty}^{\infty} x\kappa(x)dx = 0,\ \int_{-\infty}^{\infty} x^2\kappa(x)dx < \infty.$$

The bias of this estimate can be computed by the following equation [81].

$$\begin{aligned}
&\mathbb{E}[\hat{\lambda}(y)] - \lambda(y) = \mathbb{E}\Big[\sum_{i=1}^{N} \frac{1}{h}\kappa\Big(\frac{y - x_j}{h}\Big)\Big] - \lambda(y) \\
=\ & \mathbb{E}\Big[\frac{1}{h}\int_{-\infty}^{\infty} \kappa\Big(\frac{y - x}{h}\Big)dN(x)\Big] - \lambda(y) \qquad (N(x) \text{ is a counting process}) \\
=\ & \frac{1}{h}\int_{-\infty}^{\infty} \kappa\Big(\frac{y - x}{h}\Big)\lambda(x)dx - \lambda(y) = \int_{-\infty}^{\infty} \kappa(x)(\lambda(y - hx) - \lambda(y))dx \\
=\ & \int_{-\infty}^{\infty} \kappa(x)\Big(-\lambda'(y)hx + \frac{\lambda''(y)}{2}(hx)^2 + o(h^2)\Big)dx = o(h).
\end{aligned}$$

In the last equation, we use the Taylor expansion of $\lambda(x)$ at y.

$$\lambda(y - hx) = \lambda(y) - \lambda'(y)hx + \frac{\lambda''(y)}{2}(hx)^2 + o(h^2).$$

This indicates the bias of the estimate approximates 0 as the bandwidth $h \to 0$. Since the arrival times outside the window $\mathcal{X} = [0, T]$ are not observed, the end-correction is usually added to the kernel smoothing estimate [18].

$$\hat{\lambda}(x) = \frac{\sum_{i=1}^{N} \kappa_h(x - x_j)}{\int_0^T \kappa_h(x - t)dt}. \tag{2.34}$$

In Diggle [18], when using the uniform kernel function $\kappa(x) = 1/2, -1 \leq x \leq 1$, the bandwidth is chosen to minimize the expected minimum squared error (MSE) $\mathbb{E}[(\hat{\lambda}(x) - \lambda(x))^2]$. However, when a general kernel is used, MSE can not be easily computed. In the baseline experiment by Lloyd et al. [60], the bandwidth h is chosen by maximizing the leave-one-out training objective.

$$h^* = \arg\max_h \sum_{i=1}^{N} \ln \sum_{j \neq i=1}^{N} \frac{1}{h}\kappa\Big(\frac{x_i - x_j}{h}\Big).$$

Local Likelihood Method

In the local likelihood method [103], the objective function to be maximized is the local likelihood function $\mathcal{L}(\hat{\lambda}; x)$. $\hat{\lambda}$ is the estimate intensity function $\hat{\lambda}$.

$$\mathcal{L}(\hat{\lambda}; x) \stackrel{\Delta}{=} \sum_{i=1}^{N} \kappa_h(x_i - x) \ln \hat{\lambda}(x_i) - \int_0^T \kappa_h(t - x)\hat{\lambda}(t)dt. \tag{2.35}$$

To explain why maximizing Equation (2.35) is a good idea [43], note that as the number of observed events N grows, $\mathcal{L}(\hat{\lambda}; x)$ converges in probability

$$\begin{aligned}\mathcal{L}(\hat{\lambda}; x) &\to \int_0^T \kappa_h(t - x) \ln \hat{\lambda}(t)\lambda(t)dt - \int_0^T \kappa_h(t - x)\hat{\lambda}(t)dt \\ &= \int_0^T \kappa_h(t - x)\Big[\lambda(t) \ln \hat{\lambda}(t) - \hat{\lambda}(t)\Big]dt.\end{aligned}$$

This is equivalent to minimizing the Kullback-Leibler divergence

$$\int_0^T \kappa_h(t - x)\Big[\lambda(t) \ln \frac{\lambda(t)}{\hat{\lambda}(t)} - (\lambda(t) - \hat{\lambda}(t))\Big]dt,$$

where the kernel function is used to smooth the divergence.

The local likelihood method begins with the local polynomial approximation at x with parameters $\{\alpha_j\}_{j=0}^p$. The exponential transformation is used to ensure the non-negativity.

$$\ln \hat{\lambda}(s) \approx \sum_{j=0}^{p} \alpha_j (s - x)^j, \ |x - s| \leq h. \tag{2.36}$$

Inserting Equation (2.36) back to Equation (2.35), we can obtain the training objective $\mathcal{L}(\{\alpha_j\}_{j=0}^p; x)$.

$$\begin{aligned}\mathcal{L}(\{\alpha_j\}_{j=0}^p; x) \stackrel{\Delta}{=} &\sum_{i=1}^{N} \kappa_h(x_i - x) \sum_{j=0}^{p} \alpha_j (x_i - x)^j \\ &- \int_0^T \kappa_h(t - x) \exp\Big(\sum_{j=0}^{p} \alpha_j (s - x)^j\Big)dt.\end{aligned} \tag{2.37}$$

The optimal values $\{\alpha_j^*\}_{j=0}^p$ can be learned by maximizing this training objective. Based on Equation (2.36), the value the intensity function $\hat{\lambda}(s)$ at x is $\exp(\alpha_0^*)$. Taking the derivative of $\mathcal{L}(\{\alpha_j\}_{j=0}^p; x)$ with respect to α_0 and setting the derivative to be zero, we have

$$\hat{\lambda}(x) \equiv \exp(\alpha_0^*) = \frac{\sum_{i=1}^{N} \kappa_h(x_i - x)}{\int_0^T \kappa_h(t - x) \exp\Big(\sum_{j=1}^{p} \alpha_j^* (t - x)^j\Big)dt}.$$

When $p = 0$, the local likelihood estimate $\hat{\lambda}(x)$ recovers the kernel smoothing estimate with the end-correction in Equation (2.34). When $p \geq 1$, the local likelihood method is given in Algorithm 6. The bandwidth h in the local likelihood method can be found by the cross-validation.

Algorithm 6: The local likelihood method.

Input : The bandwidth h, the time window $(0, T]$, the time sequence $\boldsymbol{d}$ and the positions to be estimated $\boldsymbol{z} = \{z_i\}$.

Output: The estimation of the intensity function at the given positions $\{\hat{\lambda}(z_i)\}$.

```
Initialize k = 1, x_0 = 0.
for each z_i ∈ z do
    Construct the local likelihood L({α_j}_{j=0}^p; z_i) in Equation (2.37).
    Maximize L({α_j}_{j=0}^p; z_i) to obtain the optimal {α_j^*}_{j=0}^p.
    Set λ̂(z_i) = exp(α_0^*).
end
```

2.6.3 Variance Inference of the Intensity Function

From the Bayesian point of view, the underlying intensity function is no longer fixed before the sampling process. Instead it has a distribution. In this case, the inhomogeneous Poisson process is called a Cox process.

Definition 2.6.1. *A* Cox process *is an inhomogeneous Poisson process whose intensity function is drawn from a stochastic process.*

The stochastic process is usually chosen to be a Gaussian process, since one can sample an arbitrary function from a Gaussian process. However, as the function $f(x)$ sampled from a Gaussian process is not necessarily guaranteed to be non-negative, a transformation has to be applied to the $f(x)$. The transformation is chosen to be a sigmoid function in Adams et al. [2].

$$\lambda(x) = \frac{\lambda^*}{1 + \exp(-f(x))},\ f(x) \sim \mathcal{GP}(m(x), \kappa(x, x')),\ \lambda^* \geq 0,$$

where λ^* is a non-negative real number and it determines the upper bound of the intensity function. An MCMC algorithm is used to sample the function $f(x)$ based on observed time-sequence $\boldsymbol{d}$. In Lloyd et al. [60], the transformation is chosen to be a squared function $\lambda(x) = f^2(x)$ and the reason behind this choice is that this form admits a tractable and efficient variational inference framework. Next we introduce this variational inference framework based on the descriptions provided in Lloyd et al. [60].

Given the time-sequence data $\boldsymbol{d} = \{x_j \in \mathcal{X}\}_{j=1}^N$, we add a set of pseudo inputs $\bar{\boldsymbol{X}} = \{\bar{x}_m\}_{m=1}^M, M < N$ and their corresponding function values $\bar{\boldsymbol{f}} = \{f(\bar{x}_m)\}_{m=1}^M$ as in Section 2.4.2. The generative process for the time-sequence data $\boldsymbol{d}$ is given in Algorithm 7.

In Algorithm 7, $\kappa_{xx} \stackrel{\Delta}{=} \kappa(x, x)$,$\boldsymbol{\kappa}_{x\bar{\boldsymbol{X}}} \stackrel{\Delta}{=} \boldsymbol{\kappa}(x, \bar{\boldsymbol{X}})$ and we denote the inhomogeneous Poisson process sampling algorithm given in Algorithm 5 as IPP$(\cdot)$. The construction of the prior $p(f(x), \bar{\boldsymbol{f}})$ is different from the sparse Gaussian process representation in Section 2.4.2 since the the conditional probability is the same. When $m_0 = 0$, $f(x)$ may change its sign several times and it might restrict the optimization process [47].

The variational distribution is as follows:

$$q(f(x), \bar{\boldsymbol{f}}) = p(f(x)|\bar{\boldsymbol{f}})\mathcal{N}(\bar{\boldsymbol{f}}; \boldsymbol{\mu}, \boldsymbol{\Sigma}).$$

Algorithm 7: The generative process for the time-sequence data.

Input : The mean value $m_0 \geq 0$, the covariance function $\kappa(x, x')$ and a time window $(0, T]$

Output: The time-sequence data $\boldsymbol{d}$

1 Sample the vector $\bar{\boldsymbol{f}} \sim \mathcal{N}(\boldsymbol{m}_0, \boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}})$.

2 Compute the function $f(x)$.

$$f(x) \sim \mathcal{GP}(\boldsymbol{\kappa}_{x\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\bar{\boldsymbol{f}}, \kappa_{xx} - \boldsymbol{\kappa}_{x\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\kappa}_{x\bar{\boldsymbol{X}}}^{\top}). \tag{2.38}$$

3 Compute the intensity function $\lambda(x) = f^2(x)$.

4 Sample $\boldsymbol{d} \sim \mathrm{IPP}(\lambda(x))$ on the time window $(0, T]$.

The marginal distribution $q(f(x))$ is a Gaussian process $\mathcal{GP}(\bar{m}(x), \bar{\kappa}(x, x'))$.

$$\begin{aligned}
\bar{m}(x) &= \boldsymbol{\kappa}_{x\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\mu}, \\
\bar{\kappa}(x, x') &= \kappa_{xx'} - \boldsymbol{\kappa}_{x\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\kappa}_{x'\bar{\boldsymbol{X}}}^{\top} + \boldsymbol{\kappa}_{x\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\Sigma}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\kappa}_{x'\bar{\boldsymbol{X}}}^{\top}.
\end{aligned}$$

The training objective in the variational inference framework can be obtained from Equation (2.14), since the data likelihood $p(\boldsymbol{d}|f(x))$ is not conjugate to the Gaussian process prior.

$$\mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; \boldsymbol{\theta}) \triangleq \mathbb{E}_{q(f(x))}[\ln p(\boldsymbol{d}|f(x))] + \mathbb{E}_{q(\bar{\boldsymbol{f}})}\Big[\ln \frac{p(\bar{\boldsymbol{f}}; \bar{\boldsymbol{X}})}{q(\bar{\boldsymbol{f}})}\Big].$$

In $\mathcal{L}_{\mathrm{ELBO}}$, the second term is the same as Equation (2.18). Utilizing the data likelihood $p(\boldsymbol{d}|f(x))$ from Theorem 2.5.3, we give the computation for the first term.

$$\mathbb{E}_{q(f(x))}[\ln p(\boldsymbol{d}|f(x))] = \mathbb{E}_{q(f(x))}\Big[\sum_{i=1}^{N} \ln \lambda(x_i) - \int_0^T \lambda(x)dx\Big]. \tag{2.39}$$

Next we calculate the two expectations $\mathbb{E}_{q(f(x))} \int_0^T \lambda(x)dx$ and $\mathbb{E}_{q(f(x))} \ln \lambda(x_i)$ separately.

The Expectation $\mathbb{E}_{q(f(x))} \int_0^T \lambda(x)dx$

The expectation of the integral can be computed directly.

$$\begin{aligned}
&\mathbb{E}_{q(f(x))} \int_0^T \lambda(x)dx = \int_0^T \mathbb{E}_{q(f(x))}[f^2(x)]dx \\
&= \int_0^T \Big(\mathbb{E}^2_{q(f(x))}[f(x)] + \mathrm{Var}_{q(f(x))}[f(x)]\Big)dx = \int_0^T \Big(\bar{m}(x)^2 + \bar{\kappa}(x, x)\Big)dx \\
&= \int_0^T \Big(\kappa_{xx} - \boldsymbol{\kappa}_{x\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\kappa}_{x\bar{\boldsymbol{X}}}^{\top} + \boldsymbol{\kappa}_{x\bar{\boldsymbol{X}}}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}(\boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^{\top})\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\kappa}_{x\bar{\boldsymbol{X}}}\Big)dx \\
&= \int_0^T \Big(\kappa_{xx}dx - \mathrm{tr}(\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\Phi}) + \mathrm{tr}\Big(\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\Phi}\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}(\boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma})\Big)dx,
\end{aligned}$$

where $\boldsymbol{\Phi} \in \mathbb{R}_{>0}^{M\times M}$ and $\Phi_{ij} \triangleq \int_0^T \kappa(\bar{x}_i, z)\kappa(z, \bar{x}_j)dz$. When we assume the kernel function is the ARD kernel in Equation (2.10), the integral can be analytically

computed.

$$\begin{aligned}
&\int_0^T \kappa_{xx} dx = cT, \\
\Phi_{ij} &\triangleq \int_0^T \kappa(\bar{x}_i, z)\kappa(z, \bar{x}_j) dz \\
&= -\frac{c^2\sqrt{\pi b}}{2} \exp\Big(-\frac{(\bar{x}_i - \bar{x}_j)^2}{4b}\Big)\Big[\operatorname{erf}\Big(\frac{(\bar{x}_i + \bar{x}_j - 2T)}{2\sqrt{b}}\Big) - \operatorname{erf}\Big(\frac{\bar{x}_i + \bar{x}_j}{2\sqrt{b}}\Big)\Big].
\end{aligned}$$

The Expectation $\mathbb{E}_{q(f(x))} \ln \lambda(x_i)$

We explain the derivation in details since this part is not provided in Lloyd et al. [60]. The marginal distribution $q(f(x_i))$ is also a Gaussian distribution $\mathcal{N}(\hat{\mu}_i, \hat{\sigma}_i^2)$ with the mean and the variance given as follows:

$$\begin{aligned}
&\hat{\mu}_i = \boldsymbol{\kappa}_{x_i \bar{\boldsymbol{X}}}, \\
&\hat{\sigma}_i^2 = \kappa_{x_i x_i} - \boldsymbol{\kappa}_{x_i \bar{\boldsymbol{X}}} \boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1} \boldsymbol{\kappa}_{x_i \bar{\boldsymbol{X}}}^\top + \boldsymbol{\kappa}_{x_i \bar{\boldsymbol{X}}} \boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1} \boldsymbol{\Sigma} \boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1} \boldsymbol{\kappa}_{x_i \bar{\boldsymbol{X}}}^\top.
\end{aligned}$$

The expectation can be computed by the following integral.

$$\mathbb{E}_{q(f(x_i))} \ln \lambda(x) = \int_{-\infty}^{\infty} \mathcal{N}(x; \hat{\mu}_i, \hat{\sigma}_i^2) \ln x^2 dx. \tag{2.40}$$

We show through the following lemma that the expectation can be analytically computed.

Lemma 2.6.1. *Let a normal-distributed random variable* $Y \sim \mathcal{N}(\mu, \sigma^2)$ *and* $\varphi = (\mu/\sigma)^2$. *Then*

$$\mathbb{E}_Y[\ln Y^2] = \ln(2\sigma^2) + \sum_{j=0}^{\infty} \frac{(\varphi/2)^j \exp(-\varphi/2)}{j!} \psi(j + 1/2), \tag{2.41}$$

where $\psi(\cdot)$ *is the digamma function.*

Proof. Let $\tilde{Y} = Y/\sigma$, then the expectation can be calculated as

$$\begin{aligned}
\mathbb{E}_Y[\ln Y^2] &= \int_{-\infty}^{\infty} \ln y^2 \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(-\frac{(y-\mu)^2}{2\sigma^2}\Big) dy \\
&= \int_{-\infty}^{\infty} (\ln \tilde{y}^2 + \ln \sigma^2) \frac{\sigma}{\sqrt{2\pi}\sigma} \exp\Big(-\frac{(\tilde{y}\sigma - \mu)^2}{2\sigma^2}\Big) d\tilde{y} \qquad (Y = \bar{Y}\sigma) \\
&= \ln \sigma^2 + \int_{-\infty}^{\infty} \ln \tilde{y}^2 \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{(\tilde{y} - \mu/\sigma)^2}{2}\Big) d\tilde{y}. \qquad (2.42)
\end{aligned}$$

The second part has the form of $\mathbb{E}_{\bar{Y}}[\ln \bar{Y}^2]$, where $\bar{Y} \sim \mathcal{N}(\mu/\sigma, 1)$. Let $W = \bar{Y}^2$ and W follows a standard non-central chi-squared distribution with parameter $\varphi = (\mu/\sigma)^2$ [24]. The distribution of W is given as follows:

$$p(w) = \frac{e^{-\frac{w+\varphi}{2}}}{\sqrt{2w}} \sum_{j=0}^{\infty} \frac{(w\varphi/4)^j}{j!\Gamma(j + 1/2)}. \tag{2.43}$$

The expectation of $\ln W$ then is

$$\begin{aligned}\mathbb{E}_W[\ln W] &= \int_0^\infty \ln w \frac{e^{-\frac{w+\varphi}{2}}}{\sqrt{2w}} \sum_{j=0}^\infty \frac{(w\varphi/4)^j}{j!\Gamma(j+1/2)} dw \\ &= \sum_{j=0}^\infty \frac{(\varphi/4)^j e^{-\varphi/2}}{\sqrt{2} j!\Gamma(j+1/2)} \int_0^\infty e^{-w/2} w^{j-1/2} \ln w dw \\ &= \sum_{j=0}^\infty \frac{(\varphi/2)^j e^{-\varphi/2}}{j!} (\ln 2 + \psi(j+1/2)). \end{aligned} \tag{2.44}$$

The digamma function is defined as the integral [1].

$$\psi(z) \triangleq \frac{\Gamma'(z)}{\Gamma(z)} = \frac{1}{\Gamma(z)} \int_0^\infty e^{-x} x^{z-1} \ln x dx.$$

Substituting Equation (2.44) back yields the answer. □

To perform a practical computation, we use the property of a digamma function. Let γ be the Euler-Mascheroni constant and $\gamma \approx 0.57721$.

$$\psi\Big(i + \frac{1}{2}\Big) = -\gamma - 2\ln 2 + \sum_{i=1}^n \frac{2}{2i-1},\ i \in \mathbb{N}^+.$$

The expectation can be further transformed to the following form.

$$\mathbb{E}_Y[\ln Y^2] = -\gamma + \ln\Big(\frac{\sigma^2}{2}\Big) + \sum_{j=0}^\infty \frac{(\varphi/2)^j \exp(-\varphi/2)}{j!} \sum_{k=1}^j \frac{2}{2k-1}. \tag{2.45}$$

We can prove the following lemma to calculate the last complicated term.

Lemma 2.6.2. *Let $g(z)$ denote the sum*

$$g(z) \triangleq \exp(-z) \sum_{j=0}^\infty \frac{z^j}{j!} \sum_{k=1}^j \frac{1}{k-1/2}.$$

Let $M(a,b,z)$ denote the Kummer function of the first kind [1].

$$M(a,b,z) \triangleq \sum_{n=0}^\infty \frac{a^{(n)} z^n}{b^{(n)} n!},$$

where $a^{(0)} = 1, a^{(n)} = a(a+1)\cdots(a+n-1)$. Then

$$g(z) = -\frac{\partial M(a,b,-z)}{\partial a}\Big|_{a=0,b=0.5}, z \geq 0. \tag{2.46}$$

Proof. First we calculate the derivative of $g(z)$ with respect to z.

$$\begin{aligned} g'(z) &= \exp(-z) \sum_{j=0}^\infty \frac{z^j}{j!(j+1/2)} = 2\exp(-z) M(0.5, 1.5, z) \\ &= 2M(1, 1.5, -z) = 2\sum_{j=0}^\infty \frac{(-z)^j}{1.5^{(j)}}, \end{aligned}$$

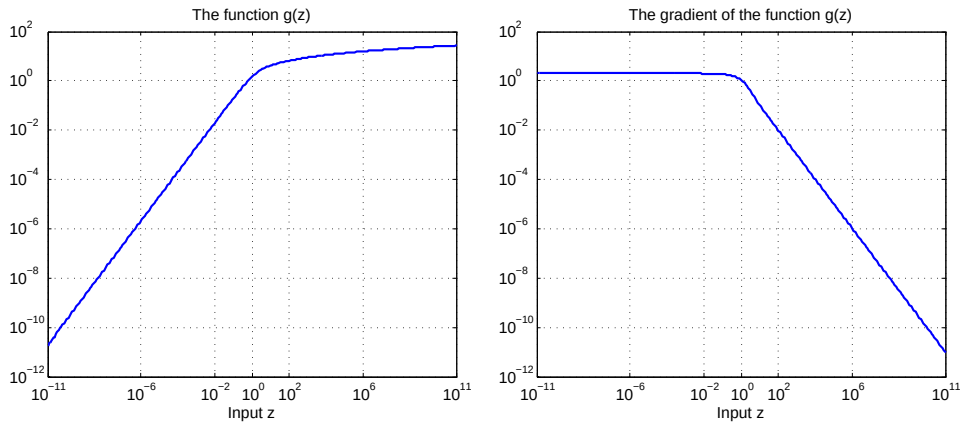

Figure 2.7: An illustration of the function $g(z)$ and $g'(z)$.

where we use the following property of the Kummer function $e^{-z}M(a,b,z) = M(b-a,b,-z)$. Integrating both the left-most and the right-most sides, we obtain that

$$g(z) = \int_0^z g'(s)ds + g(0) = -2z\sum_{j=0}^{\infty}\frac{(-z)^j}{(j+1)(1.5)_j}. \tag{2.47}$$

Using the following property from Ancarani and Gasaneo [3] for the derivative of the Kummer function of the first kind, we finish our proof.

$$\frac{\partial M(a,b,z)}{\partial a}\Big|_{a=0} = \frac{z}{b}\sum_{n=0}^{\infty}\frac{1^{(n)}1^{(n)}z^n}{2^{(n)}(1+b)^{(n)}n!} = \frac{z}{b}\sum_{n=0}^{\infty}\frac{z^n}{(n+1)(1+b)^{(n)}}.$$

□

Combining Equation (2.45) and Lemma 2.6.2, We obtain the following theorem for the computation of the expectation $\mathbb{E}_Y[\ln Y^2]$ of a normal-distributed random variable Y.

Theorem 2.6.2. *Let a normal-distributed random variable $Y \sim \mathcal{N}(\mu,\sigma^2)$ and $\varphi = (\mu/\sigma)^2$, then*

$$\mathbb{E}_Y[\ln Y^2] = -\gamma + \ln\left(\frac{\sigma^2}{2}\right) - \frac{\partial M(a,b,-\varphi/2)}{\partial a}\Big|_{a=0,b=0.5}, \tag{2.48}$$

where $M(a,b,z)$ is the Kummer function of the first kind.

Remark 1. *This theorem is not explicitly stated in Lloyd et al. [60]. We piece together the information from the previous studies [60, 56, 3].*

Since the operation of calculating $g(z)$ is very often, we use a multi-resolution lookup table to store the value of $g(z)$ and its derivative with respect to z. An illustration of the function $g(z)$ and $g'(z)$ is given in Figure 2.7.

$$\begin{aligned} g(z) &= -\frac{\partial M(a,b,-z)}{\partial a}\,|_{a=0,b=0.5} \\ &= -\lim_{\Delta\to 0^+}\frac{M(\Delta,0.5,-z)-M(0,0.5,-z)}{\Delta} = \lim_{\Delta\to 0^+}\frac{1-M(\Delta,0.5,-z)}{\Delta}, \\ g'(z) &= 2M(1,1.5,-z). \end{aligned}$$

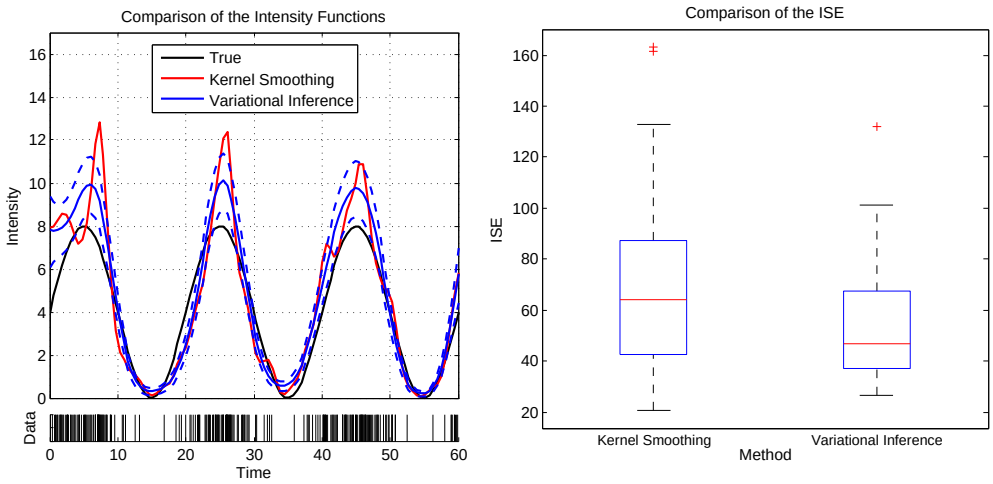

Figure 2.8: The illustration of the comparison the kernel smoothing and the variational inference method. (Left top) The inferred intensity function by the kernel smoothing, the variational inference and the true intensity function. (Left bottom) The data set used to perform the inference. (Right) The box plot of the MISE for the kernel smoothing and the variational inference method. We repeat the sampling and the inference process for 30 times.

For small z, its value $M(a, b, z)$ can be obtained from a R mathematic toolbox [104]. When $|z| \to \infty$, the computation is really slow and we use the following property [1] to approximate $M(a, b, z)$.

$$M(a, b, z) \approx \Gamma(b)\Big(\frac{e^z z^{a-b}}{\Gamma(a)} + \frac{(-z)^{-a}}{\Gamma(b-a)}\Big), |z| \to \infty.$$

Up to now, all terms in $\mathcal{L}_{\text{ELBO}}$ can be analytically computed and the parameters to be optimized are $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}, c, b\}$, where $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ are the parameters in the variational distribution and $\{c, b\}$ are the hyper-parameters of the ARD kernel function in Equation (2.10). We use the VB-EM framework in Algorithm 3 to optimize the parameters.

A comparison of the kernel smoothing estimate and the variational inference estimate are given in Figure 2.8. We calculate the integrated squared error (ISE). For the variational inference method, we use the mean function as $\hat{\lambda}(x)$.

$$\text{ISE}(\hat{\lambda}, \lambda) = \int_0^T (\lambda(x) - \hat{\lambda}(x))^2 dx.$$

Lloyd et al. [60] also show through synthetic experiments that the variational inference estimate can outperform the kernel smoothing estimate in terms of prediction accuracy.

2.6.4 Latent Poisson Process Allocation

In order to model the diversity among K subjects when the time-sequence data are in the form of recurrent events, Lloyd et al. [61] proposed the latent Poisson process allocation (LPPA) model. In LPPA, a set of L basis functions is used and the kth time-sequence is assumed to be generated by a Cox process with an intensity function $\lambda_k(x)$. The entire generative process is given by Algorithm 8.

Algorithm 8: The generative process for LPPA.

Input : The number of latent function L, the number of the time-sequences K, the mixture weights $\{\theta_{kl}\}$, the mean value m_0, the covariance functions in L Gaussian processes $\{\kappa_l\}$ and a time window $(0, T]$.

Output: The time-sequence data $\mathcal{D} = \{\boldsymbol{d}_k\}_{k=1}^K$.

for *each basis function* $l = 1, \ldots, L$ **do**
  Sample $f_l \sim \mathcal{GP}(m_0(x), \kappa_l(x, x'))$.
end
for *each subject* $k = 1, \ldots, K$ **do**
  Calculate the intensity function.

$$\lambda_k(x) = \sum_{l=1}^{L} \theta_{kl} f_l^2(x), \quad \theta_{kl} \geq 0. \tag{2.49}$$

  Sample $\boldsymbol{d}_k \sim \text{IPP}(\lambda_k(x))$ on the time window $(0, T]$.
end

In Algorithm 8, $f_l(t)$ is a function drawn from a Gaussian process prior, θ_{kl} is its weight, and L is the number of latent functions. To ensure the non-negativity of λ_k, f_l are squared and weights θ_{kl} are required to be non-negative.

Similar to the sparse Gaussian process in Section 2.4.2, a set of pseudo inputs $\bar{\boldsymbol{X}} = \{\bar{x}_m\}_{m=1}^M, M < N$ and their corresponding function values for each basis function $\bar{\boldsymbol{f}}_l = \{f_l(\bar{x}_m)\}_{m=1}^M,\ l = 1, \ldots, L$ are added. The joint likelihood is given as follows:

$$p(\mathcal{D}, \{f_l(x)\}_{l=1}^L, \{\bar{\boldsymbol{f}}_l\}_{l=1}^L) = \prod_{k=1}^{K} p(\boldsymbol{d}_k | \{f_l(x)\}_{l=1}^L; \theta_{kl}) \prod_{l=1}^{L} p(f_l(x) | \bar{\boldsymbol{f}}_l) p(\bar{\boldsymbol{f}}_l; \bar{\boldsymbol{X}}),$$

where the two distributions $p(\bar{\boldsymbol{f}}_l; \bar{\boldsymbol{X}})$ and $p(f_l(x)|\bar{\boldsymbol{f}}_l)$ are defined below:

$$\begin{aligned} p(\bar{\boldsymbol{f}}_l; \bar{\boldsymbol{X}}) &= \mathcal{N}(\bar{\boldsymbol{f}}_l; \boldsymbol{m}_0, \boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}),\ m_0 > 0, \\ f_l(x) &\sim \mathcal{GP}(\boldsymbol{\kappa}_{l,x\bar{\boldsymbol{X}}} \boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1} \bar{\boldsymbol{f}}, \kappa_{l,xx} - \boldsymbol{\kappa}_{l,x\bar{\boldsymbol{X}}} \boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1} \boldsymbol{\kappa}_{l,x\bar{\boldsymbol{X}}}^\top). \end{aligned}$$

In LPPA, the variational distribution $q(\{f_l(x)\}_{l=1}^L, \{\bar{\boldsymbol{f}}_l\}_{l=1}^L)$ is chosen as

$$\begin{aligned} q(\{f_l(x)\}_{l=1}^L, \{\bar{\boldsymbol{f}}_l\}_{l=1}^L) &= \prod_{l=1}^{L} p(f_l(x)|\bar{\boldsymbol{f}}_l) q(\bar{\boldsymbol{f}}_l) \\ &= \prod_{l=1}^{L} p(f_l(x)|\bar{\boldsymbol{f}}_l) \mathcal{N}(\bar{\boldsymbol{f}}_l; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l). \end{aligned}$$

The marginal distribution for $q(f_l(x))$ is a Gaussian process with the mean function $\bar{m}_l(x)$ and the covariance function $\bar{\kappa}_l(x, x'))$.

$$\begin{aligned} \bar{m}_l(x) &= \boldsymbol{\kappa}_{l,x\bar{\boldsymbol{X}}} \boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1} \boldsymbol{\mu}_l, \\ \bar{\kappa}_l(x, x') &= \kappa_{l,xx'} - \boldsymbol{\kappa}_{l,x\bar{\boldsymbol{X}}} \boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1} \boldsymbol{\kappa}_{l,x'\bar{\boldsymbol{X}}}^\top + \boldsymbol{\kappa}_{l,x\bar{\boldsymbol{X}}} \boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1} \boldsymbol{\Sigma} \boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1} \boldsymbol{\kappa}_{l,x'\bar{\boldsymbol{X}}}^\top. \end{aligned}$$

Given the joint data likelihood and the variational distribution, the ELBO

can be derived as follows:

$$\begin{aligned}
\ln p(\mathcal{D};\{\theta_{kl}\}) &= \ln \int p(\mathcal{D},\{f_l(x)\}_{l=1}^L,\{\bar{\boldsymbol{f}}_l\}_{l=1}^L;\{\theta_{kl}\})df_1\cdots df_L d\bar{\boldsymbol{f}}_1\cdots d\bar{\boldsymbol{f}}_L \\
&\geq \mathbb{E}_{q(\{f_l(x)\}_{l=1}^L,\{\bar{\boldsymbol{f}}_l\}_{l=1}^L)}\Big[\ln\frac{p(\mathcal{D},\{f_l(x)\}_{l=1}^L,\{\bar{\boldsymbol{f}}_l\}_{l=1}^L;\{\theta_{kl}\})}{q(\{f_l(x)\}_{l=1}^L,\{\bar{\boldsymbol{f}}_l\}_{l=1}^L)}\Big] \\
&= \prod_{k=1}^K \mathbb{E}_{q(\{f_l(x)\}_{l=1}^L)}\Big[\ln p(\boldsymbol{d}_k|\{f_l(x)\}_{l=1}^L;\theta_{kl})\Big] + \sum_{l=1}^L \mathbb{E}_{q(\bar{\boldsymbol{f}}_l)}\Big[\ln\frac{p(\bar{\boldsymbol{f}}_l;\bar{\boldsymbol{X}})}{q(\bar{\boldsymbol{f}}_l)}\Big]. \quad (2.50)
\end{aligned}$$

The second term is the sum of K negative Kullback-Leibler divergence between two Gaussian distributions, each of which can be computed similar as Equation (2.18).

$$\mathbb{E}_{q(\bar{\boldsymbol{f}}_l)}\Big[\ln\frac{p(\bar{\boldsymbol{f}}_l;\bar{\boldsymbol{X}})}{q(\bar{\boldsymbol{f}}_l)}\Big] = \frac{1}{2}\ln\frac{|\boldsymbol{\Sigma}_k|}{|\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}|}+\frac{M}{2}-\frac{1}{2}\mathrm{tr}\Big(\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}(\boldsymbol{\Sigma}_l+(\boldsymbol{\mu}_l-m_0)(\boldsymbol{\mu}_l-m_0)^\top\Big).$$

The first term, however, is not tractable. To derive a tractable lower bound of the intractable ELBO, we first prove the following lemma from Paisley [74]. In Lloyd et al. [61], a different derivation is used and the result is the same.

Lemma 2.6.3. *Let $\{X_k\}_{k=1}^K$ be a set of positive random variables, then*

$$\mathbb{E}\ln\Big(\sum_{k=1}^K X_k\Big) \geq \ln\Big(\sum_{k=1}^K \exp(\mathbb{E}\log X_k)\Big). \quad (2.51)$$

Proof. The function $\ln(\cdot)$ is concave. Using an auxiliary probability vector, $(p_1,\ldots,p_K)$, where $p_k>0$ and $\sum_{k=1}^K p_k = 1$, it follows from Jensen 's inequality that

$$\mathbb{E}\ln\Big(\sum_{k=1}^K X_k\Big) = \mathbb{E}\ln\Big(\sum_{k=1}^K p_k\frac{X_k}{p_k}\Big) \geq \sum_{k=1}^K p_k\mathbb{E}\ln\Big(\frac{X_k}{p_k}\Big) \quad (2.52)$$

Taking derivatives with respect to each element in $\{p_k\}$ and setting each derivative to zero, we have

$$p_k = \frac{\exp(\mathbb{E}\ln X_k)}{\sum_{v=1}^K \exp(\mathbb{E}\ln X_v)} \quad (2.53)$$

Inserting this back, we obtain the desired bound. □

Using this lemma, each term $\mathbb{E}_q\Big[\ln p(\boldsymbol{d}_k|\{f_l(x)\}_{l=1}^L;\theta_{kl})\Big]$ can now be lower bounded by the following theorem.

Theorem 2.6.3 (Lloyd et al. [61])**.** *A lower bound for the first term in Equation* (2.50) *is given by the following equation.*

$$\begin{aligned}
&\mathbb{E}_q\Big[\ln p(\boldsymbol{d}_k|\{f_l(x)\}_{l=1}^L;\theta_{kl})\Big] \\
&\quad\geq \sum_{j=1}^{N_k}\ln\sum_{l=1}^L\Big(\theta_{kl}+\exp\Big(\mathbb{E}_q[\ln f_l^2(x_j^{(k)})]\Big)\Big) - \int_0^T\sum_{l=1}^L\theta_{kl}\mathbb{E}_q[f_l^2(x)]dx. \quad (2.54)
\end{aligned}$$

Proof. Using the data likelihood in Theorem 2.5.3, we obtain the data likelihood for each term.

$$\begin{aligned}
\mathbb{E}_q\Big[\ln p(\boldsymbol{d}_k|\{f_l(x)\}_{l=1}^L;\theta_{kl})\Big] &= \sum_{j=1}^{N_k}\mathbb{E}_q[\ln\lambda_k(x_j^{(k)})] - \int_0^T \mathbb{E}_q\lambda_k(x)dx \\
&= \sum_{j=1}^{N_k}\mathbb{E}_q[\ln\Big(\sum_{l=1}^L \theta_{kl}f_l^2(x_j^{(k)})\Big)] - \int_0^T \mathbb{E}_q\lambda_k(x)dx \qquad \text{(Equation (2.49))} \\
&\geq \sum_{j=1}^{N_k}\ln\sum_{l=1}^L \exp\Big(\mathbb{E}_q[\ln\theta_{kl}f_l^2(x_j^{(k)})]\Big) - \int_0^T \mathbb{E}_q\lambda_k(x)dx \qquad \text{(Lemma 2.6.3)} \\
&= \sum_{j=1}^{N_k}\ln\sum_{l=1}^L \Big(\theta_{kl} + \exp\Big(\mathbb{E}_q[\ln f_l^2(x_j^{(k)})]\Big)\Big) - \int_0^T \mathbb{E}_q\sum_{l=1}^L\theta_{kl}f_l^2(x)dx.
\end{aligned}$$

□

In Equation (2.54), the two terms $\mathbb{E}_q[\ln f_l^2(x_j^{(k)})]$ and $\int_0^T \mathbb{E}_q[f_l^2(x)]dx$ can be computed analytically by the Kummer function of the first kind and the Φ matrix respectively in Section 2.6.3. Up to now, we derive a lower bound of the ELBO in Equation (2.50) and each term in the lower bound can now be analytically computed.

The parameters to be optimized are $\{\{\boldsymbol{\mu}_l\},\{\boldsymbol{\Sigma}_l\},\{\theta_{kl}\},\{c_l,b_l\}\}$, where $\{\boldsymbol{\mu}_l\}$, $\{\boldsymbol{\Sigma}_l\}$ are the parameters in the variational distribution, $\{\theta_{kl}\}$ are the mixture weights and $\{c_l,b_l\}$ are the hyper-parameters in the ARD kernel of Equation (2.10). In Lloyd et al. [61], the joint optimization on all parameters was used. However, we find it more helpful to use the VB-EM framework in Algorithm 3 since the learning rates for all parameters are not the same and this phenomenon is also reported in the on-line Gaussian process regression task [41]. Optimizing the parameters separately helps speed up the tuning process.

2.7 The Intensity Estimation for Panel Count Data

Traditional estimates for the panel count data are point estimates and the underlying assumption is that the intensity function is fixed before the sampling process. In this section, we briefly review the local expectation-maximization (LocalEM) method in Betensky et al. [9], Fan et al. [25].

For the ease of notations, we first consider the situation where we only have one subject $K = 1$ and the panel count data is $\boldsymbol{d} \triangleq \{(\mathcal{X}_i, m_i)\}_{i=1}^N$. The training objective of the LocalEM method can be derived from the local likelihood method [9]. If we have the exact time-stamp x_i, the local likelihood function is given in Equation (2.35). However, in the panel count data, the exact arrival time for each event x_i is not revealed and Betensky et al. [9] integrate the uncertainty in the arrival time x_i in the local likelihood function $\mathcal{L}(\hat{\lambda};x)$.

$$\begin{aligned}
\mathcal{L}(\hat{\lambda};x) &= \sum_{i=1}^N m_i\mathbb{E}_t\Big[\kappa_h(t-x)\ln\hat{\lambda}(t)\Big|t\in\mathcal{X}_i\Big] - \int_0^T \kappa_h(t-x)\hat{\lambda}(t)dt \\
&= \sum_{i=1}^N m_i\int_{\mathcal{X}_i}\kappa_h(t-x)\frac{\hat{\lambda}(t)}{\int_{\mathcal{X}_i}\hat{\lambda}(s)ds}\ln\hat{\lambda}(t)dt - \int_0^T \kappa_h(t-x)\hat{\lambda}(t)dt.
\end{aligned}$$

Using the local polynomial approximation in Equation (2.36), we have the approximation of $\mathcal{L}(\hat{\lambda}; x)$ and we denote it by $\mathcal{L}(\{\alpha_j\}_{j=0}^p; x)$.

$$\mathcal{L}(\{\alpha_j\}_{j=0}^p; x) \triangleq \sum_{i=1}^{N} m_i \mathbb{E}_t\Big[\kappa_h(t-x)\sum_{j=0}^{p}\alpha_j(t-x)^j\Big|t\in\mathcal{X}_i\Big] - \int_0^T \kappa_h(t-x)\exp\Big(\sum_{j=0}^{p}\alpha_j(s-x)^j\Big)dt. \tag{2.55}$$

Taking the derivative of $\mathcal{L}(\{\alpha_j\}_{j=0}^p; x)$ with respect to α_0 and setting it to zero, we obtain the relationship between the optimal values $\{\alpha_j^*\}_{j=0}^p$.

$$\hat{\lambda}(x) = \exp(\alpha_0^*) = \frac{\sum_{i=1}^N m_i \mathbb{E}_t\Big[\kappa_h(t-x)\Big|t\in\mathcal{X}_i\Big]}{\int_0^T \kappa_h(t-x)\exp\Big(\sum_{j=1}^p \alpha_j^*(s-x)^j\Big)dt}$$

Next we discuss the zero-order approximation ($p = 0$) and the high-order approximation ($p \geq 1$) separately.

2.7.1 Zero-Order Approximation

When the zero-order approximation is used, that is $p = 0$, the estimate $\hat{\lambda}$ is reduced to the following equation:

$$\hat{\lambda}(x) = \exp(\alpha_0^*) = \frac{\sum_{i=1}^N m_i \int_{\mathcal{X}_i}\kappa_h(t-x)\frac{\hat{\lambda}(t)}{\int_{\mathcal{X}_i}\hat{\lambda}(s)ds}dt}{\int_0^T \kappa_h(t-x)dt} \tag{2.56}$$

$\hat{\lambda}(x)$ appears on both sides of the equation and on the right side, the values $\hat{\lambda}(x), x \in \mathcal{X}$ are required. To reduce the computational complexity, Fan et al. [25] approximate the estimate of the intensity function $\hat{\lambda}(x), x \in \mathcal{X}$ by a piece-wise constant function.

$$\hat{\lambda}(x) \approx \frac{1}{||Q_j||}\int_{Q_j}\hat{\lambda}(u)du = \frac{\Lambda_j}{||Q_j||},\ x \in Q_j,$$

where $\Lambda_j \triangleq \int_{Q_j}\hat{\lambda}(u)du$ and $\{Q_j\}_{j=1}^J$ is a finer partition of the original window $\mathcal{X}$ from $\{\mathcal{X}_i\}_{i=1}^N$.

$$Q_k \bigcap Q_j = \emptyset, k \neq j, \qquad \bigcup_{\{j|Q_j\cap\mathcal{X}_i\neq\emptyset\}} Q_j = \mathcal{X}_i.$$

Let $I \in [0,1]^{N\times J}$ be an indicator matrix and

$$I_{ij} = \begin{cases} 1 & \text{if } \mathcal{X}_i \bigcap Q_j \neq \emptyset, \\ 0 & \text{if } \mathcal{X}_i \bigcap Q_j = \emptyset. \end{cases}$$

The zero-order estimate in Equation (2.56) can then be approximated with

Algorithm 9: The LocalEM estimate with the zero-order approximation.

Input : The bandwidth h, the time window $(0, T]$, the panel count data $\boldsymbol{d}$.

Output: The estimation of the intensity function $\hat{\lambda}(x)$.

Set $r = 0$.
Initialize $\{\Lambda_j^{(0)}\}$.
while $\{\Lambda_j^{(r)}\}_{j=1}^J$ *do not converge* **do**
    Update $\{\Lambda_l^{(r+1)}\}$ using Equation (2.58).
    $r = r + 1$.
end
Compute $\hat{\lambda}(x)$ using Equation (2.57).

this piecewise constant function.

$$\begin{aligned}\hat{\lambda}(x) &= \sum_{i=1}^N m_i \int_{\mathcal{X}_i} \frac{\kappa_h(t-x)}{\int_0^T \kappa_h(s-x)ds} \frac{\hat{\lambda}(t)}{\int_{\mathcal{X}_i} \hat{\lambda}(s)ds} dt \\ &= \sum_{i=1}^N m_i \sum_{j=1}^J \int_{Q_j} \frac{\kappa_h(t-x)}{\int_0^T \kappa_h(s-x)ds} \frac{\hat{\lambda}(t)}{\int_{\mathcal{X}_i} \hat{\lambda}(s)ds} dt \\ &\approx \sum_{i=1}^N m_i \sum_{j=1}^J \int_{Q_j} \frac{\kappa_h(t-x)}{\int_0^T \kappa_h(s-x)ds} \frac{I_{ij}\Lambda_j}{||Q_j|| \sum_{k=1}^J I_{ik}\Lambda_k} dt. \end{aligned} \tag{2.57}$$

To obtain the optimal values $\{\Lambda_j^*\}_{j=1}^J$, Fan et al. [25] first integrate the leftmost side and the rightmost sides of Equation (2.57) on Q_l.

$$\begin{aligned}\Lambda_l &= \int_{Q_l} \hat{\lambda}(x)dx \approx \int_{Q_l} \sum_{i=1}^N m_i \sum_{j=1}^J \int_{Q_j} \frac{\kappa_h(t-x)}{\int_0^T \kappa_h(s-x)ds} \frac{I_{ij}\Lambda_j}{||Q_j|| \sum_{k=1}^J I_{ik}\Lambda_k} dtdx \\ &= \sum_{j=1}^J \Big(\int_{Q_l} \int_{Q_j} \frac{\kappa_h(t-x)}{||Q_j|| \int_0^T \kappa_h(s-x)ds} dtdx \Big) \Big(\sum_{i=1}^N m_i \frac{I_{ij}\Lambda_j}{\sum_{k=1}^J I_{ik}\Lambda_k} \Big).\end{aligned}$$

Then the optimal values $\{\Lambda_j^*\}_{j=1}^J$ can then be found by iteratively updating $\{\Lambda_j^{(r)}\}_{j=1}^J, r \in \mathbb{N}^+$.

$$\Lambda_l^{(r+1)} \approx \sum_{j=1}^J \Big(\int_{Q_l} \int_{Q_j} \frac{\kappa_h(t-x)}{||Q_j|| \int_0^T \kappa_h(s-x)ds} dtdx \Big) \Big(\sum_{i=1}^N m_i \frac{I_{ij}\Lambda_j^{(r)}}{\sum_{k=1}^J I_{ik}\Lambda_k^{(r)}} \Big). \tag{2.58}$$

When using the zero-order approximation, the LocalEM method reduces to iteratively updating $\{\Lambda_j^{(r)}\}_{j=1}^J$ and the algorithm is given in Algorithm 9. We notice that if we use the zero-order approximation, we can obtain an estimate of the whole intensity function $\lambda(x)$.

The bottleneck of the computation complexity lies in the calculation of the integral $\int_{Q_l} \int_{Q_j} \kappa_h(t-x)dtdx$ for all $l, j = 1, \ldots, J$ and consequently the computation complexity scales as $O(J^2)$. Since $\{Q_j\}$ is a finer partition of $\mathcal{X}$, the computation complexity is $O(N^2A^2)$ with A and N indicating the finer partition level and the total number of intervals.

Algorithm 10: The LocalEM estimate with the high-order approximation.

Input : The bandwidth h, the time window $(0, T]$, the panel count data $\boldsymbol{d}$ and the positions to be estimated $\boldsymbol{z} = \{z_i\}$.

Output: The estimation of the intensity function at the given positions $\{\hat{\lambda}(z_i)\}$.

for *Each* $z_i \in \boldsymbol{z}$ **do**
  Initialize $\{\alpha_j^{(0)}\}_{j=0}^p, \{\Lambda_j^{(0)}\}_{j=1}^J$.
  Calculate $\mathcal{L}(\boldsymbol{\alpha}, \{\Lambda_j^{(0)}\}_{j=1}^J; z_i)$.
  Set $r = 1$.
  while $\mathcal{L}(\boldsymbol{\alpha}, \{\Lambda_j^{(0)}\}_{j=1}^J; z_i)$ *does not converge* **do**
    E-step: Update $\{\Lambda_l^{(r)}\}_{j=1}^J$ using Equation (2.60) till convergence.
    M-step: Update $\{\alpha_j^{(r)}\}_{j=0}^p$ by

$$\boldsymbol{\alpha}^{(r)} = \arg\max_{\boldsymbol{\alpha}} \mathcal{L}(\boldsymbol{\alpha}, \{\Lambda_j^{(r)}\}_{j=1}^J; z_i).$$

    $r = r + 1$.
  end
  Compute $\hat{\lambda}(z_i) = \exp(\alpha_0^{(r-1)})$.
end

2.7.2 High-Order Approximation

When the high-order approximation ($p \geq 1$) is used, the estimate can also be approximated with a piece-wise constant function.

$$\begin{aligned}\hat{\lambda}(x) = \exp(\alpha_0^*) &= \frac{\sum_{i=1}^N m_i \mathbb{E}_t\Big[\kappa_h(t-x)\Big| t \in \mathcal{X}_i\Big]}{\int_0^T \kappa_h(t-x)\exp\Big(\sum_{j=1}^p \alpha_j^*(s-x)^j\Big)dt} \\ &\approx \sum_{i=1}^N m_i \sum_{j=1}^J \int_{Q_j} \frac{\kappa_h(t-x)}{\Psi(x;\{\alpha_j^*\}_{j=1}^p)} \frac{I_{ij}\Lambda_j}{||Q_j||\sum_{k=1}^J I_{ik}\Lambda_k} dt, \end{aligned} \tag{2.59}$$

where $\Psi(x;\{\alpha_j\}_{j=1}^p)$ is defined as follows:

$$\Psi(x;\{\alpha_j\}_{j=1}^p) \triangleq \int_0^T \kappa_h(t-x)\exp\Big(\sum_{j=1}^p \alpha_j(s-x)^j\Big)dt.$$

This leads to the LocalEM algorithm to jointly update $\{\alpha_j\}_{j=0}^p$ and $\{\Lambda_j\}_{j=1}^J$. In the rth expectation step (E-step), the estimations of $\{\Lambda_j\}_{j=1}^J$ are obtained by the following Equation.

$$\Lambda_l^{(r)} \approx \sum_{j=1}^J \Big(\int_{Q_l}\int_{Q_j} \frac{\kappa_h(t-x)}{||Q_j||\Psi(x;\{\alpha_j^{(r-1)}\}_{j=1}^p)} dtdx\Big)\Big(\sum_{i=1}^N m_i \frac{I_{ij}\Lambda_j^{(r-1)}}{\sum_{k=1}^J I_{ik}\Lambda_k^{(r-1)}}\Big). \tag{2.60}$$

In the rth maximization step (M-step), the values $\{\alpha_j\}_{j=0}^p$ are obtained by

maximizing the local likelihood function.

$$\begin{aligned}\mathcal{L}(\{\alpha_j\}_{j=0}^p; x) &= \sum_{i=1}^N m_i \mathbb{E}_t\Big[\kappa_h(t-x)\sum_{j=0}^p \alpha_j (t-x)^j \Big| t \in \mathcal{X}_i\Big] \\ &\quad - \int_0^T \kappa_h(t-x)\exp\Big(\sum_{j=0}^p \alpha_j (s-x)^j\Big)dt \\ &\approx \sum_{i=1}^N m_i \sum_{j=1}^J \int_{Q_j} \kappa_h(t-x)\sum_{l=0}^p \alpha_l (t-x)^l \frac{I_{ij}\Lambda_j}{||Q_j||\sum_{k=1}^J I_{ik}\Lambda_k} dt \\ &\quad - \int_0^T \kappa_h(t-x)\exp\Big(\sum_{j=0}^p \alpha_j (s-x)^j\Big)dt. \end{aligned} \tag{2.61}$$

We denote the rightmost side of the equation as $\mathcal{L}(\boldsymbol{\alpha}, \{\Lambda_j\}_{j=1}^J; x)$.

Since the estimate $\hat{\lambda}(x)$ now depends on the optimal values $\{\{\alpha_j^*\}_{j=0}^p\}$ and the latter depends on the position of the evaluating position x, we can no longer obtain an estimate of the entire intensity function. The algorithm for jointly optimizing the $\{\alpha_j\}_{j=0}^p$ and $\{\Lambda_j\}_{j=1}^J$ is concluded in Algorithm 10.

Chapter 3

Panel Count Data: Variational Inference with Gaussian Processes

In this chapter, we present the first Bayesian inference framework for Gaussian process-modulated Poisson processes when the temporal data appear in the form of panel counts.

3.1 Introduction

Characteristics of panel count data. A common characteristic of the panel count data is that we only have the numbers of occurrences between subsequent observation times. In particular, the exact occurrence times of the events are unknown. Hence, panel counts are non-negative integers and they represent the number of occurrences of events within a fixed period. Classical examples often arise in the clinical trials [95] where patients are required to go back to the hospital after a certain treatment and only the numbers of symptoms between subsequent visits are recorded, such as the number of vomits or new tumors. Figure 3.1 gives an example of panel count data.

Objective of this study. The purpose of this chapter is to present the variational Bayesian inference on **G**aussian-**p**rocess-modulated **P**oisson **p**rocesses (GP3) that permits panel data observations.

There have been extensive studies on GP3 models and various inference algorithms are introduced for *recurrent event data* when timestamps of the events are fully observable, e.g., Monte Carlo sampling [19, 2], Laplace approximation [29] and variational inference [60]. Among these approaches, the variational inference method [60] provides a computationally efficient estimate of the intensity function and does not require a careful discretization of the underlying space. A brief review of the variational inference approach is provided in Section 2.6.3.

To the best of our knowledge, however, there has not been any study carried out on the variational inference of the GP3 model when the data come in the form of panel counts. Our ultimate goal is to infer the underlying intensity function in the panel count data.

Related statistical works. Based on the maximum likelihood criterion, several non-parametric estimators have been proposed to infer the underlying intensity function [91], e.g., a non-parametric maximum pseudo-likelihood estimator (NPMPLE) [100], a non-parametric maximum pseudo-likelihood estimator with gamma frailty (NPMPLGF) [105] and the local Expectation-Maximization (LocalEM) estimator [25]. Unlike NPMPLE and NPMPLGF, which only estimate

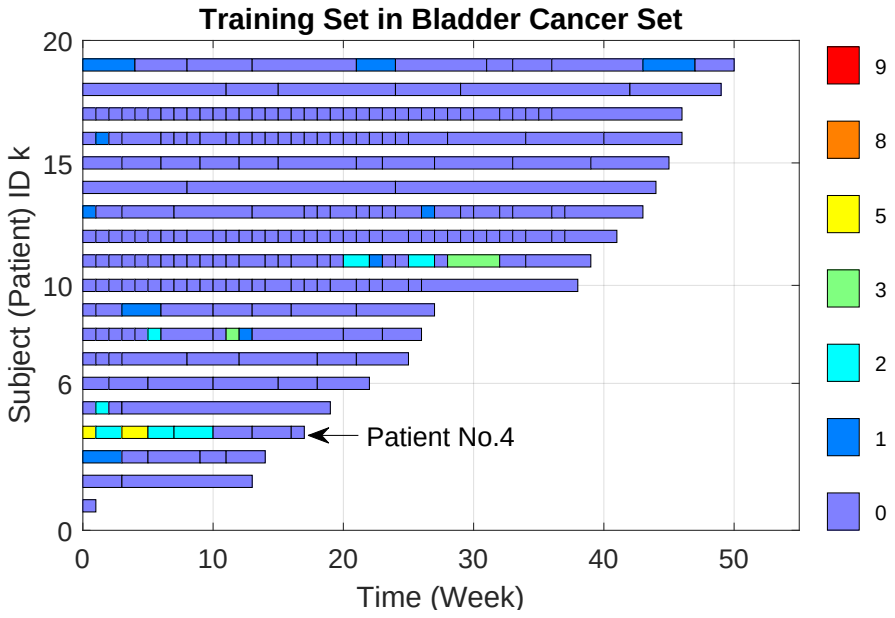

Figure 3.1: **Bladder Cancer Data Set**. This figure illustrates the panel count data from the patients. For the kth subject (or the kth patient), his/her observation window $\mathcal{X}^{(k)}$ is divided into disjoint intervals. The ith interval is denoted as $\mathcal{X}_i^{(k)}$. For example, patient No. 4 ($k = 4$) has an observation window which is divided into 8 disjoint intervals, i.e., $\bigcup_{i=1}^{8} \mathcal{X}_i^{(4)} = \mathcal{X}^{(4)}$ and $X_i \cap X_j = \emptyset$ for $i \neq j$. Patients may drop out from the study at any time and therefore their observation windows are different. An interval is shown by a rectangle. We use different colors to indicate the different numbers of new bladder tumors observed in this interval. Note that we only have access to *the number of events* in each interval.

the cumulative intensity function at a set of points, LocalEM provides a smooth estimate of the underlying intensity function due to the use of an exponential quadratic kernel [25]. A review of the LocalEM algorithm is given in Section 2.7.

Besides the computational cost in selecting the bandwidth of the exponential quadratic kernel, the estimators obtained by the LocalEM algorithm and other similar algorithms are point-estimates in the sense that the estimated intensity function is a point in the functional space. These point-estimates fail to capture the uncertainty in the data set. We show an example of the estimated intensity function by LocalEM in Figure 3.2. The uncertainty of the intensity function helps us understand the difficulty of the prediction at a given time.

3.2 Background

Before presenting the model, we first derive the likelihood of the panel count data and then briefly review the Gaussian process modulated Poisson processes. A more detailed review can be found in Section 2.6.3.

3.2.1 Likelihood of Panel Count Data

In the *recurrent event data*, one approach to modeling the events $\{x_j^{(k)} \in \mathcal{X}\}$ from each subject is to use the inhomogeneous Poisson processes (IPP) [52] and assume that there is a fixed underlying intensity function $\lambda(x) : \mathcal{X} \to \mathbb{R}^+$. Given

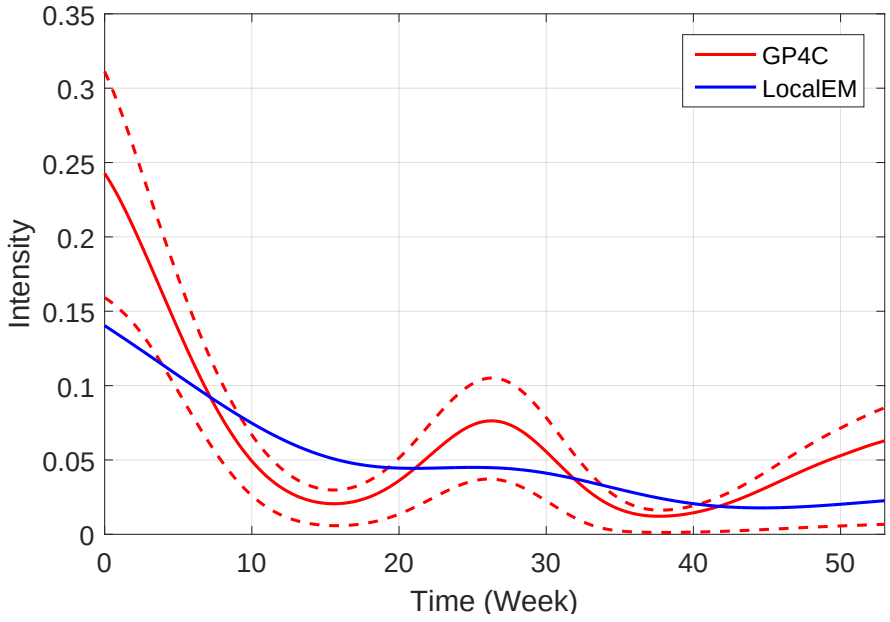

Figure 3.2: **Bladder Cancer Data Set**. Inferred intensity function by the LocalEM and GP4C methods. For GP4C, a 75% credible interval is given by dotted lines. Our estimator GP4C provides the additional uncertainty in the estimated intensity function compared with LocalEM. See Section 3.5 for details.

the intensity function $\lambda(x)$, the likelihood for the observed events is given by Theorem 2.5.3.

$$p(\{x_j^{(k)}\}|\lambda(x)) = \exp\Big(-\int_{\mathcal{X}} \lambda(x)dx\Big) \prod_j \lambda(x_j^{(k)}).$$

To derive the likelihood of the *panel count data* $\mathcal{D}$, we use two important properties of an IPP [52]. The first property is provided in Corollary 2.5.3. Given the intensity function $\lambda(x)$, the probability that we observe $m_i^{(k)}$ events in the interval $\mathcal{X}_i^{(k)}$ is given as follows:

$$p(m_i^{(k)}|\lambda(x); \mathcal{X}_i^{(k)}) = \frac{r_{ik}^{m_i^{(k)}}}{m_i^{(k)}!} \exp(-r_{ik}), \tag{3.1}$$

where $r_{ik} \triangleq \int_{\mathcal{X}_i^{(k)}} \lambda(x)dx$ is the rate parameter of the Poisson distribution. Hereafter, we omit the dependency on $\mathcal{X}_i^{(k)}$ for simplicity. However, the likelihood depends on the intervals. Even for the same sequence, after censored with different intervals, the likelihood of the sequence will vary. See Section 3.5.2 for a brief discussion.

The second property is that on two disjoint intervals $\mathcal{X}_i^{(k)}$ and $\mathcal{X}_j^{(k)}$, the numbers of events on these intervals are independent random variables.

$$p(m_j^{(k)}, m_i^{(k)}|\lambda(x)) = p(m_j^{(k)}|\lambda(x))p(m_i^{(k)}|\lambda(x)), \ \mathcal{X}_i^{(k)} \bigcap \mathcal{X}_j^{(k)} = \emptyset. \tag{3.2}$$

Based on these two properties, the likelihood of the panel count data $\mathcal{D}$ can be derived. We assume that all subjects share the same intensity function $\lambda(x)$. Using this assumption, we can obtain an estimation of the mean intensity function as is demonstrated in Theorem 2.6.1. Since K subjects are independent of each

other and for the kth subject, the N_k intervals $\{\mathcal{X}_i^{(k)}\}_{i=1}^{N_k}$ are disjoint, we obtain the following likelihood:

$$p(\mathcal{D}|\lambda(x)) = \prod_{k=1}^{K} p(\boldsymbol{d}_k|\lambda(x)) = \prod_{k=1}^{K}\prod_{i=1}^{N_k} p(m_i^{(k)}|\lambda(x)). \tag{3.3}$$

Several maximum likelihood estimators have been proposed on the basis of this likelihood or its variants, e.g., NPMPLE [100, 101], NPMPLGF [105] and the LocalEM estimator [25]. An estimate from LocalEM using the zero-order approximation[1] on the data set in Figure 3.1 is given in Figure 3.2. As we discussed, these estimators fail to model the uncertainty in the intensity function.

3.2.2 GP3 Model

In order to model the uncertainty of the intensity function $\lambda(x)$ via a kernel, the traditional approach is to use the Cox process [52]. A Cox process is defined via a stochastic intensity function $\lambda(x)$. The stochastic process to generate the intensity function is usually chosen to be a Gaussian process (GP) [2] and the model using a GP is called a GP3 model.

For the *recurrent event data*, GP3 models have been studied extensively [2, 38, 60]. The following model is an example of GP3 models [60],

$$\lambda(x) = f^2(x),\ f \sim \mathcal{GP}(g(x), \kappa(x, x')), \tag{3.4}$$

where $\mathcal{GP}(g(x), \kappa(x, x'))$ denotes the Gaussian process with mean function $g(x)$ and covariance function $\kappa(x, x')$. The function $f(x)$ drawn from a GP prior is squared to ensure the non-negativity of the intensity function. The GP3 model in Equation (3.4) admits a complete variational inference framework. Moreover, this intensity model can be enhanced with an independent variable for each subject or a mixture structure [61] to flexibly model the heterogeneity of the intensity functions across several subjects.

3.3 Variational Inference Framework

In this section, we explain the GP4C model. We derive a simple and tractable lower bound of the intractable evidence lower bound and then introduce a heuristic method to analyze the error of the lower bound.

3.3.1 Model

In order to retain the scalability and efficiency of the variational inference approach [60] and add the uncertainty on the intensity function when we only observe the panel count data, we use the GP3 model defined in Equation (3.4) as the underlying intensity model.

The joint distribution $p(\mathcal{D}, f)$ can be obtained by combining the likelihood model in Equation (3.3) and the intensity model in Equation (3.4).

$$p(\mathcal{D}, f) = \Big[\prod_{k=1}^{K} p(\boldsymbol{d}_k|\lambda(x))\Big] p(f; g, \kappa). \tag{3.5}$$

We call this model the **GP**-modulated **P**oisson **P**rocess model for **P**anel **C**ount data (GP4C).

[1] Note that we did not compare the result with the higher-order approximation since using the higher-order approximation, we can not obtain an estimate of the entire intensity function.

3.3.2 Variational Inference

We use the GP construction in Section 2.6.3 to reduce the computational complexity with the set of pseudo inputs $\bar{\boldsymbol{X}} = \{\bar{x}_m\}_{m=1}^M$ on $\mathcal{X}$ [60]. Let $\bar{\boldsymbol{f}} \triangleq [f(\bar{x}_1), \ldots, f(\bar{x}_M)]^\top$. The joint model with additional pseudo inputs is

$$p(\mathcal{D}, f, \bar{\boldsymbol{f}}) = p(\mathcal{D}|f)p(f|\bar{\boldsymbol{f}})p(\bar{\boldsymbol{f}}).$$

The variational distribution is defined as follows:

$$q(f, \bar{\boldsymbol{f}}) = p(f|\bar{\boldsymbol{f}})q(\bar{\boldsymbol{f}}), \tag{3.6}$$

where $q(\bar{\boldsymbol{f}}) = \mathcal{N}(\bar{\boldsymbol{f}}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathcal{N}(\bar{\boldsymbol{f}}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The evidence lower bound (ELBO) $\mathcal{L}$ can be obtained by using Jensen's inequality.

$$\begin{aligned}
\ln p(\mathcal{D}) &\geq \iint q(f, \bar{\boldsymbol{f}}) \ln \frac{p(\mathcal{D}, f, \bar{\boldsymbol{f}})}{q(f, \bar{\boldsymbol{f}})} df d\bar{\boldsymbol{f}} \\
&= \sum_{k=1}^{K} \sum_{i=1}^{N_k} \Big(m_i^{(k)} \mathbb{E}_q \Big[\ln \int_{\mathcal{X}_i^{(k)}} f^2(x) dx \Big] - \ln(m_i^{(k)}!) \Big) \\
&\quad - \sum_{k=1}^{K} \mathbb{E}_q \Big[\int_{\mathcal{X}^{(k)}} f^2(x) dx \Big] + \mathbb{E}_q \Big[\ln \frac{p(\bar{\boldsymbol{f}})}{q(\bar{\boldsymbol{f}})} \Big] \triangleq \mathcal{L}.
\end{aligned} \tag{3.7}$$

In ELBO, when assuming that the covariance function $\kappa(x, x')$ is the automatic relevance determination (ARD) function in Equation (2.10), the second term in the ELBO can be analytically calculated [60] and we copy the result from Section 2.6.3.

$$\begin{aligned}
\int_{\mathcal{X}^{(k)}} \mathbb{E}^2_{q(f(x))}[f(x)] dx &= \operatorname{tr}(\boldsymbol{K}^{-1}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} \boldsymbol{\Phi} \boldsymbol{K}^{-1}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} \boldsymbol{\mu}\boldsymbol{\mu}^\top), \\
\int_{\mathcal{X}^{(k)}} \operatorname{Var}_{q(f(x))}[f(x)] dx &= c|\mathcal{X}^{(k)}| - \operatorname{tr}(\boldsymbol{K}^{-1}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} \boldsymbol{\Phi}) + \operatorname{tr}(\boldsymbol{K}^{-1}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} \boldsymbol{\Phi} \boldsymbol{K}^{-1}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} \boldsymbol{\Sigma}), \\
\mathbb{E}_q \Big[\int_{\mathcal{X}^{(k)}} f^2(x) dx \Big] &= \int_{\mathcal{X}^{(k)}} \Big(\mathbb{E}^2_{q(f(x))}[f(x)] + \operatorname{Var}_{q(f(x))}[f(x)] \Big) dx \\
&= c|\mathcal{X}^{(k)}| - \operatorname{tr}(\boldsymbol{K}^{-1}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} \boldsymbol{\Phi}) + \operatorname{tr}(\boldsymbol{K}^{-1}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} \boldsymbol{\Phi} \boldsymbol{K}^{-1}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}} (\boldsymbol{\mu}\boldsymbol{\mu}^\top + \boldsymbol{\Sigma})),
\end{aligned} \tag{3.8}$$

where $\boldsymbol{\Phi}$ is an $R \times R$ matrix related to the pseudo inputs with its (i, j)-th entry equal to $\int_{\mathcal{X}^{(k)}} \kappa(\bar{x}_i, x)\kappa(x, \bar{x}_j) dx$ and $\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}$ is the covariance matrix computed at the pseudo inputs. However, the ELBO $\mathcal{L}$ is still intractable, since we can not analytically compute the expected integral $\mathbb{E}_q \Big[\ln \int_{\mathcal{X}_i^{(k)}} f^2(x) dx \Big]$ in the first term.

3.3.3 A Tractable Lower Bound

We tackle the intractable expectation by deriving a tractable lower bound. In Lemma 2.6.1, we have shown that if $Y \sim \mathcal{N}(\mu, \sigma^2)$ and $\varphi = (\mu/\sigma)^2$, then

$$\mathbb{E}_Y[\ln Y^2] = \ln(2\sigma^2) + \sum_{j=0}^{\infty} \frac{(\varphi/2)^j \exp(-\varphi/2)}{j!} \psi(j + 1/2). \tag{3.9}$$

Let

$$g_m(y) \triangleq \sum_{j=0}^{\infty} \frac{y^j \exp(-y)}{j!} \psi(j + m). \tag{3.10}$$

We have $\mathbb{E}_Y[\ln Y^2] = \ln(2\sigma^2) + g_{0.5}(\varphi/2)$. The function $g_m(y)$, where y is a positive real number and m is a positive integer, has been studied in the analysis of mobile and wireless communication systems [69]. For $m = 1/2$, $g_{0.5}(\varphi/2)$ can be computed using the Kummer function of the first kind in Section 2.6.3, which is stored in a precomputed multi-resolution look-up table.

$$g_{0.5}(\varphi/2) = -\frac{\partial M(a, 1/2, -\varphi/2)}{\partial a}\Big|_{a=0} - 2\ln 2 - \gamma, \tag{3.11}$$

where γ is Euler's constant and $\gamma \approx 0.5772$. However, to the best of our knowledge, it is still not clear how to calculate the integral of the function $g_{0.5}(\varphi/2)$ when the parameter φ comes from a GP. To derive a tractable lower bound of the intractable expectation, we introduce the following theorem to give a lower bound of the function $g_m(y)$ and the proof is deferred to Section 3.6.

Theorem 3.3.1. *Let $Y \sim \mathcal{N}(\mu, \sigma^2)$ and ξ be a constant which does not depend on μ and σ.*

$$\mathbb{E}_Y[\ln Y^2] \geq \ln(\mu^2 + b\sigma^2) + \xi, \ \forall b \in [0, 1]. \tag{3.12}$$

Based on Theorem 3.3.1, we propose the following corollary which introduces a lower bound for the intractable expectation in the ELBO.

Corollary 3.3.1. *Let f be a GP as defined in Equation (3.4). For $b \in [0, 1]$, the following bound holds:*

$$\mathbb{E}_q\Big[\ln \int_{\mathcal{X}_i^{(k)}} f^2(x)dx\Big] \geq \ln\Big(\int_{\mathcal{X}_i^{(k)}} \Big(\mathbb{E}_q^2 f(x) + b\mathrm{Var}_q f(x)\Big)dx\Big) + \xi, \tag{3.13}$$

where the variational distribution q is given in Equation (3.6).

Proof. We first use Jensen's inequality on the logarithm function and then interchange the order of integration and expectation.

$$\mathbb{E}_q\Big[\ln \int_{\mathcal{X}_i^{(k)}} f^2(x)dx\Big] = \mathbb{E}_q\Big[\ln \int_{\mathcal{X}_i^{(k)}} \tilde{p}(x)\frac{f^2(x)}{\tilde{p}(x)}dx\Big] \geq \int_{\mathcal{X}_i^{(k)}} \tilde{p}(x)\mathbb{E}_q\Big[\ln \frac{f^2(x)}{\tilde{p}(x)}\Big]dx, \tag{3.14}$$

where $\tilde{p}(x)$ is a probability distribution on $\mathcal{X}_i^{(k)}$. Furthermore, maximizing this lower bound with respect to $\tilde{p}(x)$ yields the optimal distribution:

$$\tilde{p}_{\mathrm{opt}}(x) \propto \exp\Big(\mathbb{E}_q \ln f^2(x)\Big). \tag{3.15}$$

We remark that this result is analogous to that of the discrete version presented in Paisley [74]. Substituting Equation (3.15) into the right-hand side of Equation (3.14) yields

$$\begin{aligned}
\mathbb{E}_q\Big[\ln \int_{\mathcal{X}_i^{(k)}} f^2(x)dx\Big] &\geq \ln\Big(\int_{\mathcal{X}_i^{(k)}} e^{\mathbb{E}_q \ln f^2(x)}dx\Big) \\
&\geq \ln\Big(\int_{\mathcal{X}_i^{(k)}} e^{\ln(\mathbb{E}_q^2 f(x) + b\mathrm{Var}_q f(x)) + \xi}dx\Big) \quad \text{(Theorem 3.3.1)} \\
&= \ln\Big(\int_{\mathcal{X}_i^{(k)}} \Big(\mathbb{E}_q^2 f(x) + b\mathrm{Var}_q f(x)\Big)dx\Big) + \xi, \qquad (3.16)
\end{aligned}$$

where we have invoked Theorem 3.3.1 in the penultimate line whilst defining $y := f(x)$. □

It should be emphasized that we are making no further assumptions on the dimensionality of x in the proof of Corollary 3.3.1. Hence we may augment the dimensionality of x in Corollary 3.3.1 such that it can also be applied to problems in spatial point processes. In summary, the ELBO in Equation (3.7) inherits an analytical bound. We present the following:

Theorem 3.3.2. *A tractable lower bound of the ELBO $\mathcal{L}$ in the GP4C model is given as follows:*

$$\begin{aligned}\mathcal{L} \geq \tilde{\mathcal{L}} \triangleq & -\sum_{k=1}^{K} \mathbb{E}_q\Big[\int_{\mathcal{X}^{(k)}} f^2(x)dx\Big] + \mathbb{E}_q\Big[\ln \frac{p(\bar{\boldsymbol{f}})}{q(\bar{\boldsymbol{f}})}\Big] \\ & + \sum_{k=1}^{K}\sum_{i=1}^{N_k} m_i^{(k)} \ln\Big(\int_{\mathcal{X}_i^{(k)}} \Big(\mathbb{E}_q^2 f(x) + b\mathrm{Var}_q f(x)\Big)dx\Big) \\ & - \sum_{k=1}^{K}\sum_{i=1}^{N_k} \Big(-m_i^{(k)}\xi + \ln(m_i^{(k)}!)\Big). \end{aligned} \tag{3.17}$$

Proof. The theorem can be obtained by applying Corollary 3.3.1 on the ELBO $\mathcal{L}$ in Equation (3.7). □

The derivations of $\mathbb{E}_q^2 f(x)$ and $\mathrm{Var}_q f(x)$ in the second line of Equation (3.17) follow similar derivations in Equation (3.8). The third line in $\tilde{\mathcal{L}}$ is a constant and thus can be omitted when maximizing the lower bound. In the lower bound, $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ and $\boldsymbol{\theta} = \{c, b\}$ are the variational parameters and hyper-parameters in the covariance function of a GP, respectively. We use the variational Expectation-Maximization (vEM) algorithm [17] in Algorithm 3 to update the variational parameters and the hyper-parameters iteratively on the modified ELBO $\tilde{\mathcal{L}}$.

3.3.4 The Value of Parameter b

A natural question is, how do we select the parameter b in Corollary 3.3.1? Recall that two inequalities were used in the proof. It is cumbersome to evaluate Inequality (3.14) since it is an integral over $\mathcal{X}_i^{(k)}$. We first examine the influence of Theorem 3.3.1.

Let the difference between the lower bound and the true value in Theorem 3.3.1 be $h(\varphi; b)$.

$$\begin{aligned} h(\varphi; b) &\triangleq \ln(\mu^2 + b\sigma^2) + \xi - \mathbb{E}_Y[\ln Y^2] \\ &= \ln(\mu^2 + b\sigma^2) + \xi - \Big(\ln(2\sigma^2) - \frac{\partial M(a, 1/2, -\varphi/2)}{\partial a}\Big|_{a=0} - 2\ln 2 - \gamma\Big) \\ &= \ln(\varphi^2 + b) + \xi' + \frac{\partial M(a, 1/2, -\varphi/2)}{\partial a}\Big|_{a=0}, \end{aligned}$$

where $\xi' = \xi + \ln 2 + \gamma$. Let $\varphi(x) \triangleq \mathbb{E}_q^2[f(x)]/\mathrm{Var}_q[f(x)]$. In Equation (3.16) where we invoke Theorem 3.3.1, the difference f_{error} between the two sides of the inequality is

$$\begin{aligned} f_{\text{error}} \triangleq & \ln\Big(\int_{\mathcal{X}_i^{(k)}} e^{\mathbb{E}_q \ln f^2(x)} dx\Big) - \ln\Big(\int_{\mathcal{X}_i^{(k)}} e^{\ln(\mathbb{E}_q^2 f(x) + b\mathrm{Var}_q f(x)) + \xi} dx\Big) \\ = & \ln\Big(\int_{\mathcal{X}_i^{(k)}} e^{\mathbb{E}_q \ln f^2(x)} dx\Big) - \ln\Big(\int_{\mathcal{X}_i^{(k)}} e^{\mathbb{E}_q \ln f^2(x)} e^{-h(\varphi(x); b)} dx\Big). \end{aligned}$$

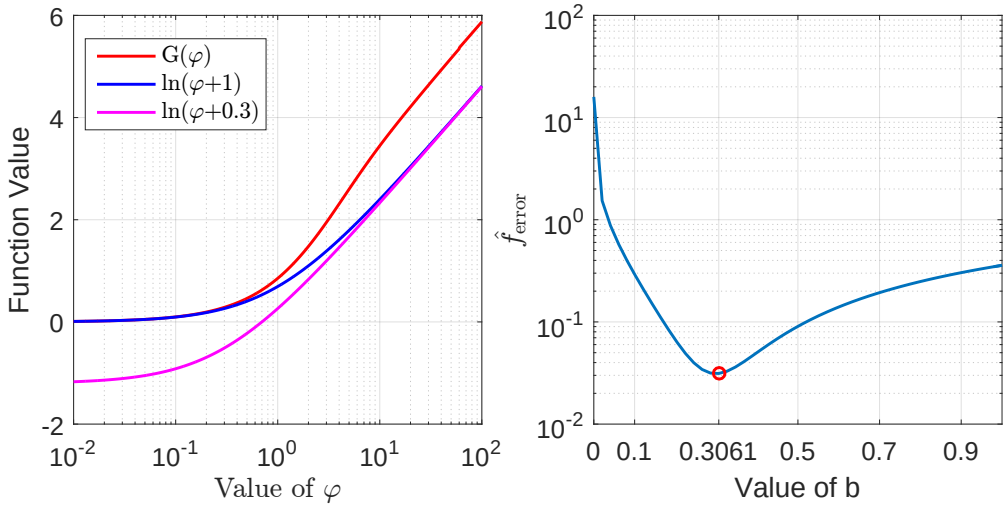

Figure 3.3: **Influences of b in Lemma 2**. (Left) The true value of $G(\varphi) = g_{0.5}(\varphi/2) + 2\ln 2 + \gamma$ by a look-up table and two simple lower bounds $\ln(\varphi+1)$ and $\ln(\varphi+0.3)$. The curve $\ln(\varphi+0.3)$ correlates with the curve of the true value better. (Right). The heuristic error $\hat{f}_{\text{error}}$ when varying the choices of b and the best b is shown with a red circle.

Since the two functions $e^{\mathbb{E}_q \ln f^2(x)}$ and $e^{-h(\varphi(x);b)}$ are both non-negative and continuous for $x \in \mathcal{X}_i^{(k)}$, by the mean value theorem there exists $x_c \in \mathcal{X}_i^{(k)}$ such that

$$\ln\Big(\int_{\mathcal{X}_i^{(k)}} e^{\mathbb{E}_q \ln f^2(x)} e^{-h(\varphi(x);b)} dx\Big) = \ln\Big(\int_{\mathcal{X}_i^{(k)}} e^{\mathbb{E}_q \ln f^2(x)} dx\Big) - h(\varphi(x_c);b).$$

Consequently, the error is $f_{\text{error}} = h(\varphi(x_c);b)$, $x_c \in \mathcal{X}_i^{(k)}$. However, it is difficult to obtain the value of x_c or an upper bound on the error f_{error}.

In Paisley et al. [75], a more correlated lower bound of the ELBO serves as a better control variate in reducing the variance of a stochastic gradient. The correlation can be measured by the variance of the error. Inspired by this study, we use a heuristic approach to examine different choices of b. More specifically, for each choice of b, we calculate the following heuristic error.

$$\hat{f}_{\text{error}} \triangleq \frac{1}{|\Phi|}\sum_{\varphi\in\Phi}(h(\varphi;b) - \bar{h})^2, \ \bar{h} = \frac{1}{|\Phi|}\sum_{\varphi\in\Phi} h(\varphi;b).$$

The set Φ consists of 5000 logarithmically spaced points between 10^{-6} and 10^6. We calculate $\hat{f}_{\text{error}}$ on a vector of 50 evenly spaced choices of b between 0 and 1. The result is shown in Figure 3.3. We see that the optimal choice of b is 0.3061 and this indicates that the choice of b might not be close to 0 or 1. In the actual situation, this optimal value of b depends on the range of φ in the data and the influence of Inequality (3.14), we evaluate several choices of b on synthetic data sets in Section 3.5.

3.3.5 Computational Complexity

Let each interval in temporal point processes be $\mathcal{X}_i^{(k)} = [x_{ai}^{(k)}, x_{bi}^{(k)}]$ with two end points $x_{ai}^{(k)}$ and $x_{bi}^{(k)}$. Two intervals are different if at least one end point is different. We denote the number of different intervals in the data set as N

and the number of pseudo inputs as M. For each interval, the computation complexity of GP4C is $\mathcal{O}(M^3)$ which is determined by the inverse of the matrix calculation when evaluating $\mathrm{Var}_q f(x)$ in Equation (3.17). The computational complexity during one iteration of the vEM algorithm is $\mathcal{O}(NM^3)$ since in our implementation, we calculate the integral of all N different intervals.

We analyze the computational complexity of the LocalEM [25] algorithm for comparison. In LocalEM, $\{x_{ai}^{(k)}\}$ and $\{x_{bi}^{(k)}\}$ are first merged into a single ordered set X where duplicated values are removed. We denote the size of the merged set X as $\bar{N}$ and generally $\bar{N} \leq N$. Then the Gaussian quadratic rule with $\bar{M}$ points is used to calculate the integral of the intensity function between subsequent values in the set X and the computational complexity during one iteration is $\mathcal{O}(\bar{N}^2\bar{M}^2)$.

If the size of the merged set $\bar{N}$ is significantly smaller than N, LocalEM may be computationally more efficient than GP4C. However, if $\bar{N} \approx N$, LocalEM may suffer from the term $\bar{N}^2$ in the computational complexity. We provide additional experiments on the influence of the number $\bar{N}$ in Section 3.5.

3.4 GP4C With Individual Weight

In this section, we briefly discuss how to model the diversity from multiple time-sequences. Then we will show how to use the VB-EM framework in Algorithm 3 to optimize the additional parameters.

3.4.1 Model

It is practical to assume that the kth subject has an individual weight parameter υ_k multiplied to the basic intensity function, because in traditional panel count data sets, each subject is a patient whose personal information, such as age, is not the same and the count data from each patient may vary greatly. Such a modification is called the unobservable independent random effects in Cook and Lawless [15]. In the simplest case, we consider the following model for the underlying intensity function:

$$\lambda_k(x) = \upsilon_k f^2(x),\ f \sim \mathcal{GP}(g(x), \kappa(x, x')),\ k = 1, \ldots, K, \tag{3.18}$$

where $\upsilon_k \in \mathbb{R}_{>0}$ is a deterministic and positive real number. The likelihood is as follows.

$$p(\mathcal{D}, f) = \Big[\prod_{k=1}^{K} p(\boldsymbol{d}_k | \lambda(x); \upsilon_k) \Big] p(f; g, \kappa). \tag{3.19}$$

We call this model the **GP4C** with individual **W**eight (GP4CW) model. We can further generalize this model by assuming that the intensity function of the kth subject is a linear combination of basis intensity functions as in LPPA [61] and the mixture weights are also deterministic.

3.4.2 Variational Inference

The derivation of the lower bound in the GP4CW model is almost the same as the procedure in GP4C. Similarly to Theorem 3.3.2, we can obtain the following lower bound for the GP4CW model and the proof is omitted.

Theorem 3.4.1. *A tractable lower bound of the ELBO $\mathcal{L}$ in the GP4CW model*

is given as follows:

$$\begin{aligned}\mathcal{L} \geq \tilde{\mathcal{L}} \triangleq & -\sum_{k=1}^{K} \upsilon_k \mathbb{E}_q\Big[\int_{\mathcal{X}^{(k)}} f^2(x)dx\Big] + \mathbb{E}_q\Big[\ln \frac{p(\bar{\boldsymbol{f}})}{q(\bar{\boldsymbol{f}})}\Big] \\ & + \sum_{k=1}^{K}\sum_{i=1}^{N_k} m_i^{(k)} \ln\Big(\int_{\mathcal{X}_i^{(k)}} \Big(\mathbb{E}_q^2 f(x) + b\mathrm{Var}_q f(x)\Big)dx\Big) + \sum_{k=1}^{K}\sum_{i=1}^{N_k} m_i^{(k)} \ln \upsilon_k \\ & - \sum_{k=1}^{K}\sum_{i=1}^{N_k}\Big(-m_i^{(k)}\xi + \ln(m_i^{(k)}!)\Big). \end{aligned} \tag{3.20}$$

Note that we have an additional set of hyper-parameters $\{\upsilon_k\}_{k=1}^K$. A point estimate of υ_k can be found by taking the derivative of $\tilde{\mathcal{L}}$ and setting it to zero. The result is given as follows:

$$\upsilon_k = \max\Big\{\epsilon, \frac{\sum_{i=1}^{N_k} m_i^{(k)}}{\int_{\mathcal{X}^{(k)}} \mathbb{E}_q[f^2(x)]dx}\Big\}, \tag{3.21}$$

where $\epsilon = 10^{-6}$ is a small number to guarantee the positiveness of υ_k. The learning algorithm is given in Algorithm 11.

Algorithm 11: The learning algorithm for GP4CW.

Input : The training data set $\mathcal{D} = \{\boldsymbol{d}_k\}_{k=1}^K, K \in \mathbb{N}_+$.
Output: An estimation of the parameters $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ in the variational distribution $q(\bar{\boldsymbol{f}})$ and the hyper-parameters $\boldsymbol{\theta} = \{c, b, \boldsymbol{\upsilon}\}$.

```
Initialize t = 0, L̃^(t) = Inf.
Initialize the parameters μ^(t), Σ^(t) and the hyper-parameters θ^(t).
while True do
    t = t + 1.
    Update μ^(t), Σ^(t) to increase L̃(μ, Σ; θ^(t−1)).
    Update c^(t), b^(t) to increase L̃(μ^(t), Σ^(t); c, b, υ^(t−1)).
    for each k = 1, ..., K do
        Evaluate υ_k^(t) in Equation (3.21).
    end
    Calculate the current L̃^(t) = L̃(μ^(t), Σ^(t); θ^(t)).
    if |L̃^(t) − L̃^(t−1)| < 10^−6 |L̃^(t)| then
        Break.
    end
end
```

3.5 Experiment

We evaluate our proposed GP4C model and compare it with the benchmark methods on both synthetic and real-world data sets. The algorithms are programmed in Matlab R2015b and run on an Intel Xeon E5-2667 CPU with a memory of 64GB. Our code is available at github.com/Dinghy/GP4C.

3.5.1 Experiment Settings

For each data set $\mathcal{D}$, we randomly partition the subjects into training and testing sets, which we denote as $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$, respectively. We repeat each setting

for $S = 40$ times. In the sth trial, the training and testing sets are denoted as $\mathcal{D}_{\text{train}}^{(s)}$ and $\mathcal{D}_{\text{test}}^{(s)}$.

Benchmark

Two benchmark algorithms are used.

a) **GP3** [60]. This benchmark reflects the performance that can be obtained if we obtain the recurrent event data set where we have the exact timestamps.

b) **LocalEM** [25]. Both LocalEM and GP4C are nonparametric estimators based on the maximum likelihood criterion. To fairly compare the computation time, we implemented the LocalEM algorithm in MATLAB based on the R code provided in Fan et al. [25]. This method produces a smooth estimate of the intensity function due to the use of an exponential quadratic kernel. We use a 5-fold cross-validation on the training set to select the bandwidth of the exponential quadratic kernel.

Evaluation Metric

We evaluate the performance of the algorithms in terms of three metrics.

a) The integrated squared error (ISE). In synthetic data sets, we have the ground truth of the intensity function λ_{true} and the integrated squared error can be calculated using our estimated intensity function $\lambda_{\text{est}}^{(s)}$ during the sth trial. To measure the bias of each estimator, we calculate the mean of the integrated squared error as follows:

$$\text{ISE}(s) \triangleq \int_{\mathcal{X}} (\lambda_{\text{est}}^{(s)}(x) - \lambda_{\text{true}}(x))^2 dx. \tag{3.22}$$

For GP4C, to measure its bias, we omit the variance of the estimator and use the expectation of the intensity function $\mathbb{E}_{q^{(s)}}[f^2(x)]$ as $\lambda_{\text{est}}^{(s)}(x)$.

b) Test log likelihood $\mathcal{L}_{\text{test}}$. During the sth trial, the logarithm of the test likelihood can be written as follows:

$$\mathcal{L}_{\text{test}}(s) \triangleq \ln \int p(\mathcal{D}_{\text{test}}^{(s)}|f) p(f|\mathcal{D}_{\text{train}}^{(s)}) df. \tag{3.23}$$

For LocalEM, since this estimator provides a point-estimate and we directly use the estimated function $f^{(s)}$ to calculate $\mathcal{L}_{\text{test}}(s)$. For GP4C and GP3, we need to sample the function $f^{(s)}$ from the variational distribution. Recall that during the sth trial, the test likelihood is

$$\begin{aligned}
\mathcal{L}_{\text{test}}(s) &\triangleq \ln \int p(\mathcal{D}_{\text{test}}^{(s)}|f) p(f|\mathcal{D}_{\text{train}}^{(s)}) df \\
&\approx \ln \frac{1}{U} \sum_{u=1}^{U} p(\mathcal{D}_{\text{test}}^{(s)}|f^{(s,u)}) \qquad (3.24) \\
&= \ln \sum_{u=1}^{U} \exp\Big(\ln p(\mathcal{D}_{\text{test}}^{(s)}|f^{(s,u)}) \Big) - \ln U \\
&= \ln \sum_{u=1}^{U} \exp\Big(\sum_{k=1}^{K_{\text{test}}} \sum_{i=1}^{N_k} \Big(m_i^{(k)} \ln r_{ik}^{(s,u)} - \ln(m_i^{(k)}!) \Big) \\
&\quad - \sum_{k=1}^{K_{\text{test}}} \int_{\mathcal{X}^{(k)}} \Big(f^{(s,u)}(x) \Big)^2 dx \Big) - \ln U. \qquad (3.25)
\end{aligned}$$

In the above derivation, we use

$$f^{(s,u)} \sim \mathcal{N}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\Sigma}^{(s)}),$$
$$r_{ik}^{(s,u)} = \int_{\mathcal{X}_i^{(k)}} \left(f^{(s,u)}(x)\right)^2 dx.$$

We can calculate the test likelihood for each subject similarly. In Equation (3.24), we draw $U = 50$ samples of the function $f^{(s,u)}$ from the variational distribution $q^{(s)}(f)$ on a vector of 3001 evenly-spaced points on $\mathcal{X}$ and we approximate points at an arbitrary position on $\mathcal{X}$ with the linear interpolation. The log-exp-sum trick is used to calculate the $\mathcal{L}_{\text{test}}(s)$. We calculate all integrals in $p(\mathcal{D}_{\text{test}}^{(s)}|f)$ using Simpson's rule with 501 evenly-spaced points.

In Equation (3.25), the term $\sum_k \sum_i \ln(m_i^{(k)}!)$ can be extracted out and treated as a constant.

c) Computation time T. We record the training time measured in seconds for each setting. For GP3 and GP4C, we record the computation time of the training process. For LocalEM, it includes the time of 5-fold cross-validation on the training set to select the bandwidth of the exponential quadratic kernel and the time of a training process over the whole training set.

Optimization Settings

For GP3 and GP4C, following Lian et al. [57], we use the re-parametrization trick $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^\top$ by Cholesky decomposition and add positivity constraints to the diagonal elements in $\boldsymbol{L}$. Due to this constraint on $\boldsymbol{L}$, we use the limited-memory projected quasi-Newton algorithm [85] to optimize the variational parameters $\{\boldsymbol{\mu}, \boldsymbol{L}\}$. We add a jitter term $\epsilon \boldsymbol{I}$ where $\epsilon = 10^{-6}$ to the covariance matrix $\boldsymbol{K}_{\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}$ to avoid numerical instability [96].

3.5.2 Synthetic Data Sets

We test three synthetic data sets which we denote as the Synthetic A, B and C data sets, respectively.

On the Synthetic A data set, the intensity function is a square wave function $h_1(x)$ as follows. See Figure 3.4 for an illustration of $h_1(x)$.

$$h_1(x) = \begin{cases} 7 & \text{if } \operatorname{mod}\left(\left[\frac{x}{10}\right], 2\right) = 0, \\ 2 & \text{otherwise.} \end{cases}$$

On the Synthetic B and C data sets, the underlying intensity functions are drawn according to Equation (3.4). We first draw a function from a GP on a vector of 3001 evenly-spaced points in $\mathcal{X} = [0, T]$, where $T = 60$. We approximate the value of the function at an arbitrary position with linear interpolation. The function is then squared to guarantee the positiveness of the intensity function. See Figure 3.6 for an illustration.

During the sth trial, we first generate a *recurrent event data set* with 100 subjects on the same observation window $\mathcal{X}^{(k)} = \mathcal{X}$. Then we generate the corresponding *panel count data set* $\mathcal{D}^{(s)}$ by censoring each subject with 10 intervals. We generate the censored intervals by a draw from a Dirichlet distribution $\boldsymbol{w}^{(k)} \sim \text{Dir}(\boldsymbol{\theta})$ and $\boldsymbol{\theta}$ is a 10-dimensional vector with all elements equal to 1. The ith

Table 3.1: **Synthetic data sets**. Mean and standard deviation of statistics about different choices of b over 40 runs. GP3 uses the *recurrent event data* while LocalEM and GP4C use the *panel count data*. For GP4C, $b = 0.3$ and $b = 0$ perform better than $b = 1$ in terms of ISE and $\mathcal{L}_{\text{test}}$.

Method	**ISE**	$\mathcal{L}_{\text{test}}$	$T[s]$
(Synthetic A)			
GP3	29.5±1.0	-1366.5±17.4	16±4
GP4C(1)	41.8±6.2	-3236.9±542.3	25±5
GP4C(0)	40.8±3.3	-1378.1±16.9	19±4
GP4C(0.3)	40.2±3.2	-1377.8±17.5	20±3
LocalEM	44.6±3.1	-1383.5±17.0	33±2
(Synthetic B)			
GP3	0.5±0.2	-783.1±20.7	8±1
GP4C(1)	1.9±2.1	-1005.8±81.5	55±44
GP4C(0)	2.7±0.8	-794.5±20.1	17±3
GP4C(0.3)	2.4±0.7	-794.2±20.2	17±4
LocalEM	3.5±0.7	-800.3±19.6	33±2
(Synthetic C)			
GP3	1.2±0.4	-864.1±14.9	8±3
GP4C(1)	2.3±1.5	-1194.6±100.5	52±53
GP4C(0)	2.1±0.6	-871.2±15.9	17±2
GP4C(0.3)	2.0±0.7	-872.0±15.7	18±3
LocalEM	5.2±1.1	-882.7±16.5	34±2

interval of the kth subject can be computed as $\mathcal{X}_i^{(k)} = [\sum_{j=1}^{i-1} w_j^{(k)} T, \sum_{j=1}^{i} w_j^{(k)} T]$. We randomly partition $\mathcal{D}^{(s)}$ into two parts, where 50 subjects are used for training and 50 for testing.

Different choices of the hyper-parameter b

On all three synthetic data sets, we test three different choices of b in $\{0, 0.3, 1\}$. We choose the number of pseudo inputs to be 30. We calculate the ISE and $\mathcal{L}_{\text{test}}$ and the results are provided in Table 3.1. We see that $b = 0, 0.3$ generally outperform $b = 1$ on these simple synthetic data sets. However, the difference between $b = 0$ and $b = 0.3$ is not significant. The reason is that Inequality (3.14) and the range of φ on $\mathcal{X}$ are also relevant to the actual performance of different b, as we discussed in Section 3.3.4.

To investigate the reason behind the bad performance of $\mathcal{L}_{\text{test}}$ when $b = 1$, we plot the best result in terms of ISE during 40 trials in Figure 3.4. We see that GP4C ($b = 1$) over-estimates the variance of the intensity function and the over-estimated variance leads to the poor performance in $\mathcal{L}_{\text{test}}$. We fix $b = 0.3$ during the remaining experiments for simplicity.

Number of the pseudo inputs

We vary the number of pseudo inputs in GP3 and GP4C since this number determines the accuracy of approximation in a sparse GP. We expect that for

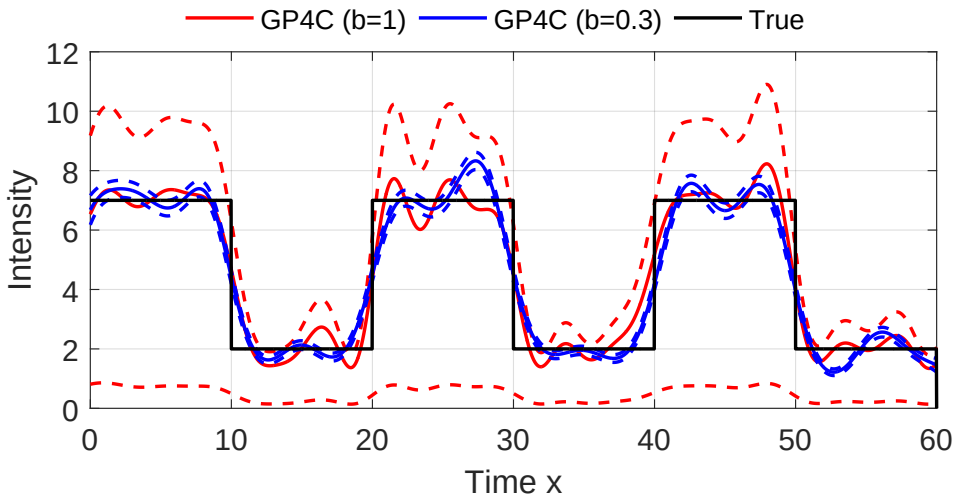

Figure 3.4: **Synthetic A Data Set**. The estimated intensity functions from GP4C ($b = 1$) and GP4C ($b = 0.3$) are shown with 75% credible intervals. True intensity function $h_1(x)$ is given for comparison. We see that GP4C ($b = 1$) over-estimates the variance of the intensity function.

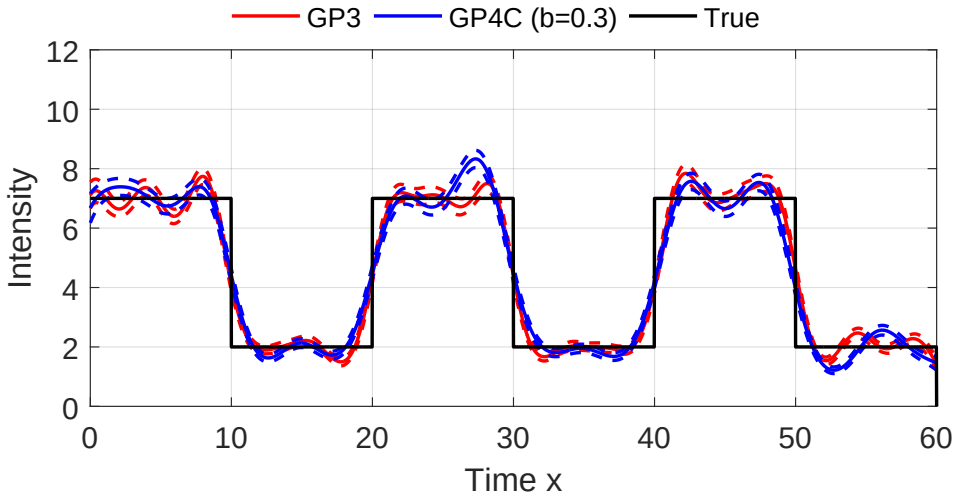

Figure 3.5: **Synthetic A Data Set**. The estimated intensity functions from GP3 and GP4C ($b = 0.3$) are shown with 75% credible intervals. True intensity function $h_1(x)$ is given for comparison. We see that the variance of the GP3 and GP4 ($b = 0.3$) are comparable.

GP-based methods the test likelihood will be relatively stable when increasing the number of pseudo inputs according to previous studies on sparse GPs [96].

The result for the Synthetic A data set is given in Figures 3.7. In Figure 3.7, we see that for GP3 and GP4C, ISE and $\mathcal{L}_{\text{test}}$ stay relatively stable with the increase of the number of pseudo inputs. The computation time of GP3 and GP4C will grow with the increase of the number of pseudo inputs.

In both Table 3.1 and Figure 3.7, we see that GP4C outperforms LocalEM on these three datasets. However, we also notice that there is still a gap between GP3 and GP4C in terms of $\mathcal{L}_{\text{test}}$ and ISE in Table 3.1. Two reasons may account for this fact. The first one is that the data are provided in the form of panel counts rather than exact timestamps. The second reason is that we use a lower

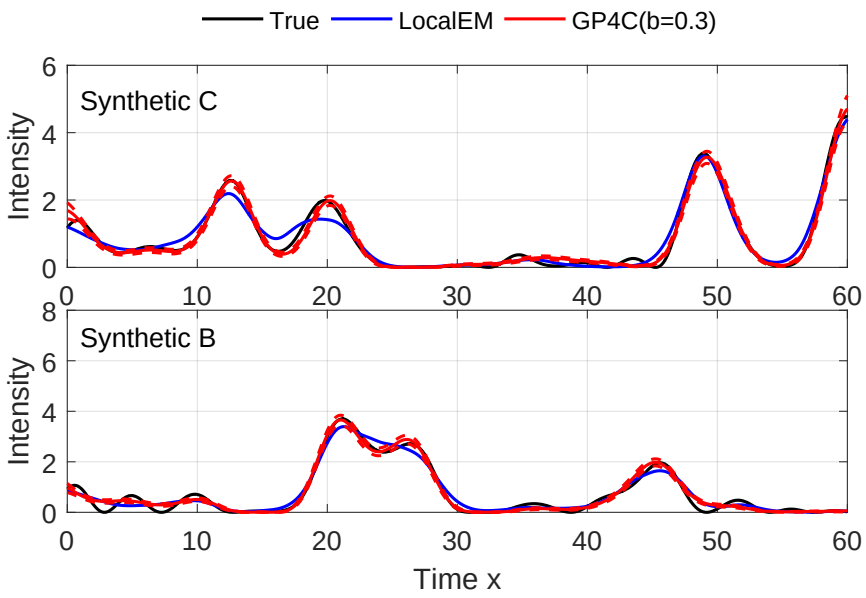

Figure 3.6: **Synthetic B & C Data Sets**. An illustration of the underlying intensity functions and inferred intensity functions by the LocalEM and GP4C methods. The underlying intensity function is drawn from a Gaussian process. For GP4C, a 75% credible interval is given by dotted lines.

bound of the true ELBO to perform the variational inference, which may lead to a bias. This bias can be alleviated with the stochastic variational inference [75], where our lower bound can serve as a control variate. We leave this as a future study.

The Dependence of the Likelihood on the Number of Intervals

The likelihood of the panel count data for the kth subject depends on the disjoint intervals $\{\mathcal{X}_i^{(k)}\}_{i=1}^{N_k}$, where $\bigcup \mathcal{X}_i^{(k)} = \mathcal{X}^{(k)}$. One phenomenon is that as the number of disjoint intervals N_k increases, the likelihood tends to decrease. This is because as we use finer disjoint intervals, we are less uncertain about the position of the time-stamps.

We conduct an experiment to show this phenomenon. First we draw a time-sequence from the intensity function $\lambda(t) = 5$ on $\mathcal{X} = [0, 60]$ and then censor the time-sequence using N disjoint intervals. We vary the number of disjoint intervals N and calculate the likelihood of the generated panel count data set. The result is given in Figure 3.10. We see that the logarithm of the likelihood decreases with the increase of the number of intervals.

The Dependence of the Computation Complexity on the Size $\bar{N}$

The computational complexity of LocalEM during one iteration is $\mathcal{O}(\bar{N}^2\bar{M}^2)$ while for GP4C it is $\mathcal{O}(NM^3)$, where N and $\bar{N}$ denote the number of different intervals in the data set and the size of the merged set X. We conduct an experiment to show the influence of the size $\bar{N}$.

We generate $U = 70$ subjects from the same intensity function $\lambda(t) = h_1(t)$, which is the same as Synthetic A data set. We generate the corresponding panel count data set by censoring each subject with 10 intervals. Then we vary the number of $\bar{N}$ by rounding each end point to the next smaller integer with the

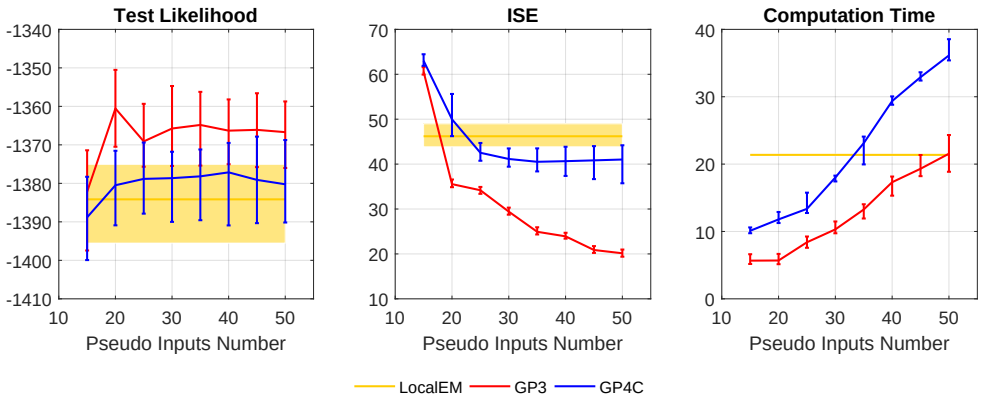

Figure 3.7: **Synthetic Data Set**. Comparison of performance of GP3, GP4C and LocalEM in terms of $\mathcal{L}_{\text{test}}$, ISE and T when varying the number of pseudo inputs for sparse GPs. For the test likelihood, ISE and the computation time, the median, the 0.25 and 0.75 quantiles of the statistics in 40 experiments are shown with error bars or shaded area. For GP3 and GP4C, ISE and $\mathcal{L}_{\text{test}}$ stay relatively stable with the increase of the number of pseudo inputs.

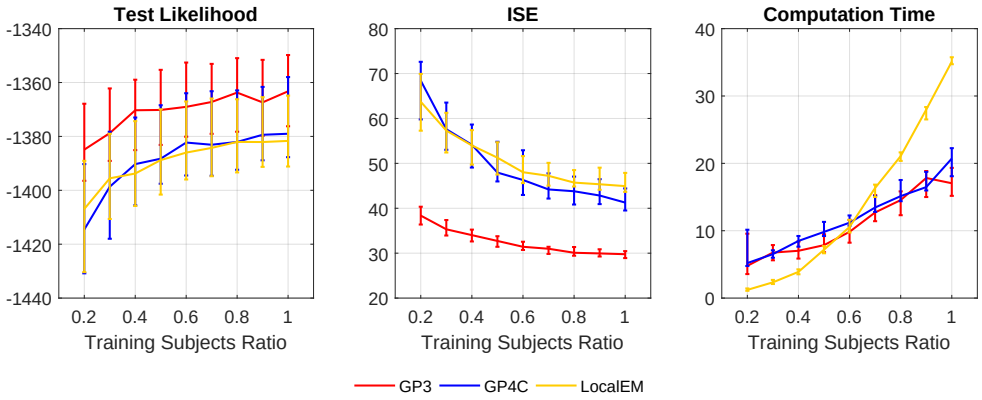

Figure 3.8: **Synthetic Data Set**. Comparison of performance of GP3, GP4C and LocalEM in terms of $\mathcal{L}_{\text{test}}$, ISE and T when varying the ratio of training subjects and the test set is the same. For ISE and the computation time, the 0.25 and 0.75 quantiles of the statistics in 40 experiments are shown with error bars. All methods benefit from the increase of the number of training subjects. The computation time of GP3 and GP4C grow linearly with the increase of the number of training subjects.

probability p_0. As p_0 get larger, more end points are rounded and the value of $\bar{N}$ decreases. The experiment result is given in Figure 3.9.

From the left plot of Figure 3.9, we see that the number $\bar{N}$ decreases linearly with the probability p_0. In the right plot, we notice that the computational time of LocalEM increases much faster when we have fewer duplicated points. We can conclude that when the number of duplicates is large, LocalEM is less efficient than GP4C.

Ratio of the Training Objects

We vary the number of training subjects by adjusting the ratio relative to full training subjects. We expect all methods will benefit from the increase of the

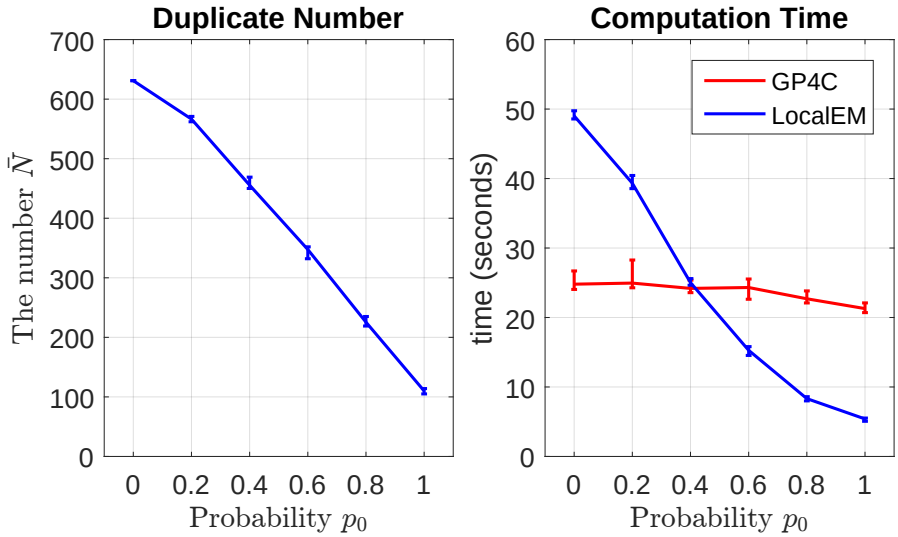

Figure 3.9: **Synthetic A Data Set**. Comparison of the computation time of GP4C and LocalEM algorithms when varying the number of duplicated points in the panel count data set. LocalEM algorithm achieves a worse computation time as the probability p_0 gets smaller.

training subjects.

The result for the Synthetic A data set is given in Figure 3.8. We see that all three methods benefit from the increase of the number of training subjects. The computation time of GP3 and GP4C grow linearly with the increase of the number of training subjects but LocalEM grows more rapidly.

3.5.3 Real World Data Sets

Sun and Zhao [91] provided three panel count data sets. Some statistics can be found in Table 3.2. A brief description about the these data sets can be found below.

a) **Nausea data set**. This data set contains the visiting times from 113 patients during 52 weeks. The panel count data were obtained by recording the reported count of vomits from each patient between two subsequent visits. Patients were divided into two groups, which are the treatment group (65 patients) and the placebo group (48 patients). We denote the two groups as the Nausea A (Na-A) and B (Na-B) sets.

b) **Bladder cancer data set**. This data set arises from a bladder cancer study conducted by the Veterans Administration Cooperative Urological Research Group. It records the counts of new tumors that occurred between subsequent visits from 85 patients during 53 weeks, who were divided into the placebo group (47 patients) and the treatment group (38 patients). We denote the two groups as the Bladder A (Bl-A) and B (Bl-B) sets.

c) **Skin cancer data set**. This data set was recorded during a skin cancer experiment conducted by the University of Wisconsin Comprehensive Cancer Center and the numbers of new skin cancers of two different types between two subsequent visits from 290 patients were recorded during five years. The visiting time was recorded in the form of days since the first visit and we divided the days by 30. Patients were divided into treatment

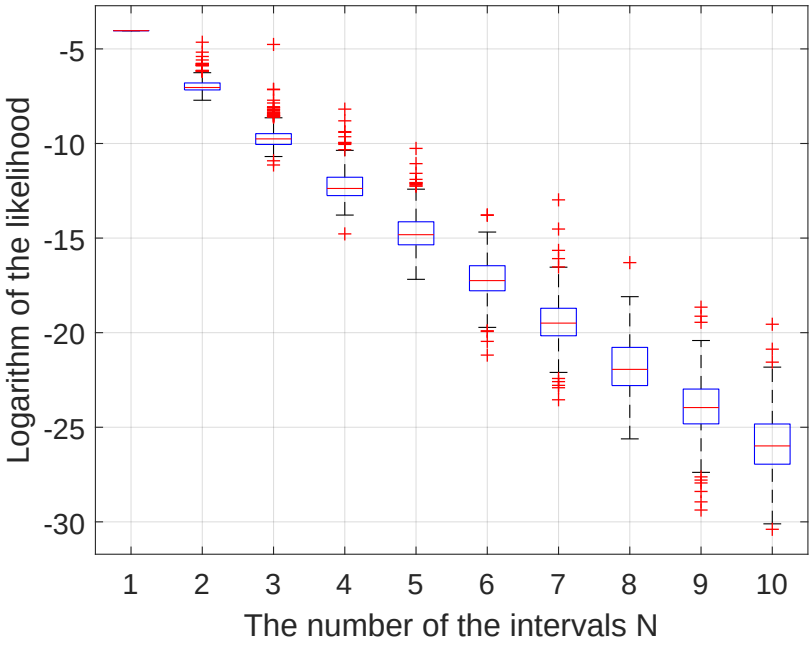

Figure 3.10: The logarithm of the likelihood of the same time-sequence when varying the number of disjoint intervals. As more disjoint intervals are used, the logarithm of the likelihood decreases. Even for the same number of disjoint intervals, the logarithm of the likelihood has a large variance.

Table 3.2: Statistics about the three data sets, where K, $\mathcal{X}$, $\bar{N}$ and N denote the number of subjects in each data set, the underlying continuous space, the number of different end points and the number of different intervals $\mathcal{X}_i^{(k)}$, respectively.

Data Set	$\mathcal{X}$	K	$\bar{N}$	N
Na-A	$[0, 55]$	65	45	109
Na-B	$[0, 55]$	48	38	84
Bl-A	$[0, 53]$	38	52	176
Bl-B	$[0, 53]$	47	52	201
Sk-A & Sk-B	$[0, 61.57]$	143	751	816
Sk-C & Sk-D	$[0, 62.63]$	147	808	887

and placebo groups. We denote the four groups from two types of cancer as the Skin A (Sk-A), Skin B (Sk-B), Skin C (Sk-C) and Skin D (Sk-D) sets.

Experiment Results Using GP4C

We use 18 pseudo inputs for all real world experiments. In each trial, we randomly split each data set into two parts, which are $\mathcal{D}_{\text{train}}^{(s)}$ (50%) and $\mathcal{D}_{\text{test}}^{(s)}$ (50%). On these three data sets, since the original data are in the form of panel counts, GP3 is not tested. We compare GP4C with LocalEM in terms of $\mathcal{L}_{\text{test}}$ and the computation time T.

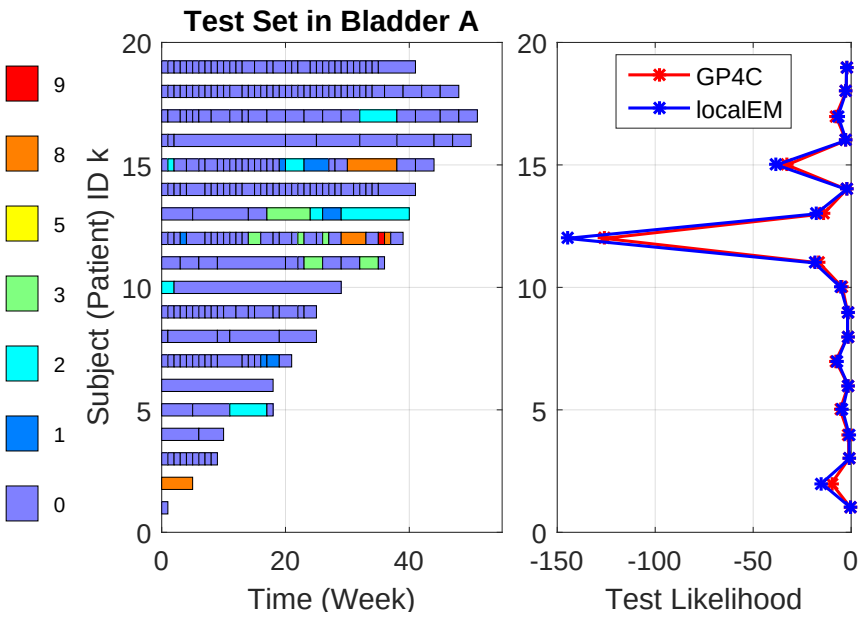

Figure 3.11: **Bladder A Data Set**. An illustration of the panel count data in the test set (Left) and the test likelihood from GP4C and LocalEM of each subject (Right). GP4C mainly outperforms LocalEM on two subjects whose numbers of newly-occurred cancers are large (No. 12 and 15).

The results are given in Table 3.3. The standard deviation of the likelihood is large since the likelihood depends on the censored intervals of the subjects, which vary greatly in different train/test split. We conduct an experiment to reduce the standard deviation in the end of this section. In Table 3.3, LocalEM performs better on the Nausea and Bladder data sets in terms of the computation time T. GP4C outperforms LocalEM in terms of test likelihood $\mathcal{L}_{\text{test}}$ in all data sets.

To see the difference between GP4C and LocalEM, we show the result of inferred intensities by two algorithms during one trial on the Bladder A data set in Figure 3.2. We see that GP4C provides the additional uncertainty which helps improve $\mathcal{L}_{\text{test}}$ compared with LocalEM. Since the Bladder A set is small, we plot the panel count data in the training set in Figure 3.1. The test set and the test likelihood of all its subjects are given in Figure 3.11. From the test likelihood of each subject, we see that GP4C outperforms LocalEM on two subjects whose counts of newly-occurred tumors are large (No. 12 and No. 15). The count 8 never occurs in the training set and a point-estimate will fail to model this uncertainty while a GP-modulated method will take the uncertainty into consideration.

Another observation about this data set is that there is a heterogeneity across all subjects. The traditional approach to modeling heterogeneity is to add an additional variable on the intensity function for each subject [15]. We have briefly discussed how to add this change to GP4C in Section 3.4.

Experiment Results Using GP4CW

On the three real world data sets, we implement the GP4CW model and the experiment settings are the same as GP4C. The test likelihood $\mathcal{L}_{\text{test}}$ and the computation time T are given in Table 3.4. We also plot the test likelihood of each subject and the inferred intensity function from GP4CW on the Bladder A data set in Figures 3.12 and 3.13, respectively. We can notice that GP4CW provides

Table 3.3: Mean and standard deviation of the test likelihood ($\mathcal{L}_{\text{test}}$) and the computation time T measured in seconds on the three panel count data sets over 40 runs. LocalEM performs better on the Nausea and Bladder data sets in terms of computation time. In all data sets, GP4C performs better on the test likelihood and outperforms LocalEM on computation time in the Skin data sets.

Data Set	**METHOD**	$\mathcal{L}_{\text{test}}$	$T[s]$
Na-A	LocalEM	-492.1±306.1	1±0
	GP4C	-484.9±201.8	10±10
Na-B	LocalEM	-473.2±212.2	1±0
	GP4C	-411.0±184.3	10±7
Bl-A	LocalEM	-201.8±46.9	1±0
	GP4C	-182.2±47.3	25±9
Bl-B	LocalEM	-313.1±54.2	1±0
	GP4C	-310.4±54.9	26±21
Sk-A	LocalEM	-259.1±27.3	39±3
	GP4C	-258.7±26.7	33±6
Sk-B	LocalEM	-198.1±47.1	39±3
	GP4C	-191.2±42.5	24±4
Sk-C	LocalEM	-358.0±35.8	47±4
	GP4C	-355.7±36.0	21±12
Sk-D	LocalEM	-200.9±31.9	46±3
	GP4C	-198.9±30.6	27±4

more accurate test likelihood on the patient No. 12 and No. 15. However, since GP4CW utilizes the unobservable independent random effects assumption, GP4CW can not provide an estimate of the mean intensity function as GP4C.

Experiment Results on Reducing the Standard Deviation on the Real World Data Set

In each trial, we randomly split the whole data set into two halves $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$, one for training and the other for testing. However, as is discussed in Figure 3.10, if the subjects in $\mathcal{D}_{\text{test}}$ do not share the same time window $\mathcal{X}_k$ and the same set of disjoint censoring intervals, the test likelihood $\mathcal{L}_{\text{test}}^{(1)}$ will vary greatly from subject to subject.

To reduce the large standard deviation caused by different random splits, we perform another round of training for each split, we train on $\mathcal{D}_{\text{test}}$ and calculate the test likelihood on the $\mathcal{D}_{\text{train}}$. The test likelihood is denoted as $\mathcal{L}_{\text{test}}^{(2)}$. This can be viewed as adding an additional reverse split. The final test likelihood is $\mathcal{L}_{\text{test}}^{(1)} + \mathcal{L}_{\text{test}}^{(2)}$. The result is given in the fourth column of Table 3.4.

We see that the variances of the test likelihood in the fourth column are reduced comparing to results in the third column.

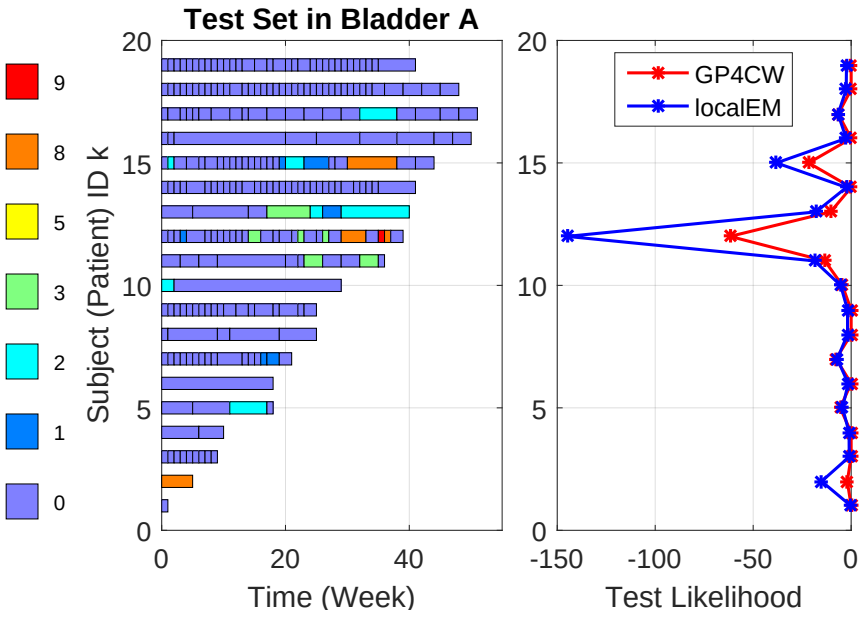

Figure 3.12: **Bladder A Data Set**. An illustration of the panel count data in the test set (Left) and the test likelihood from GP4CW and LocalEM of each subject (Right). GP4CW mainly outperforms LocalEM on two subjects whose numbers of newly-occurred cancers are large (No. 12 and 15).

3.6 Proof of Theorem 3.3.1

In this section, we will prove Theorem 3.3.1. Let us recall that

$$g_m(x) = \sum_{j=0}^{\infty} \frac{x^j \exp(-x)}{j!} \psi(j+m). \tag{3.26}$$

The derivative of $g_m(x)$ with respect to x is

$$\begin{aligned} g'_m(x) &= \sum_{j=0}^{\infty} \frac{(jx^{j-1} - x^j)\exp(-x)}{j!} \psi(j+m) \\ &= \sum_{j=0}^{\infty} \frac{x^j \exp(-x)}{j!} (\psi(j+m+1) - \psi(j+m)) \\ &= \sum_{j=0}^{\infty} \frac{x^j \exp(-x)}{j!} \frac{1}{j+m}, \end{aligned} \tag{3.27}$$

where we use the property of the digamma function [1].

$$\psi(z+1) - \psi(z) = \frac{1}{z}, z > 0.$$

To prove Theorem 3.3.1, we first present one important lemma.

Lemma 3.6.1. *For $m = \frac{1}{2}$, the following inequality holds.*

$$g'_m(x) \geq \begin{cases} \frac{1}{x+m} & \text{if } x \geq 1.5, \\ \frac{1}{x+m} + g'_m(1.5) - \frac{1}{m} & \text{if } 0 \leq x \leq 1.5. \end{cases}$$

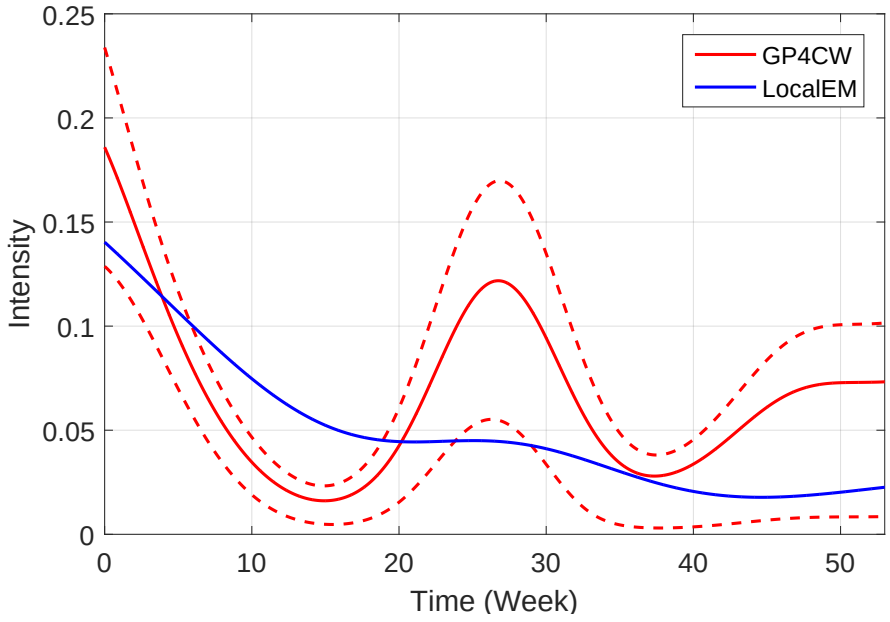

Figure 3.13: **Bladder Cancer Data Set**. Inferred intensity function by the LocalEM and GP4CW methods. For GP4CW, a 75% credible interval is given by dotted lines.

Proof. Let $h(x) = \frac{1}{x+0.5}$. We first examine the gradient of $g'_{0.5}(x)$ and $h(x)$.

$$\begin{aligned} g''_{0.5}(x) &= \sum_{j=0}^{\infty} \frac{x^j \exp(-x)}{j!} \Big(\frac{1}{j+0.5+1} - \frac{1}{j+0.5} \Big) \\ &= -\sum_{j=0}^{\infty} \frac{x^j \exp(-x)}{j!} \frac{1}{(j+0.5+1)(j+0.5)} < 0, \\ h'(x) &= -\frac{1}{(x+0.5)^2} < 0. \end{aligned}$$

Therefore $g'_{0.5}(x)$ and $h(x)$ are strictly decreasing. Next we examine the crossing points of $g'_{0.5}(x)$ and $h(x)$. One crossing point of $g'_{0.5}(x)$ and $h(x)$ is $x = 0$, since

$$g'_{0.5}(0) = \frac{1}{0.5} = h(0).$$

Using a property of the Kummer function of the first kind [1], we have

$$\begin{aligned} g'_{0.5}(x) = 2M(1, 1.5, -z) &= \int_0^1 \frac{\exp(-ux)}{\sqrt{1-u}} du \\ &\geq \int_0^1 \exp(-ux) du = \frac{1-e^{-x}}{x}. \end{aligned}$$

Since $e^x > 1 + 2x,\ x \geq 1.5$, we can show

$$g'_{0.5}(x) - h(x) \geq \frac{1-e^{-x}}{x} - h(x) = \frac{0.5e^{-x}(-2x + e^x - 1)}{x(x+0.5)} > 0,\ x \geq 1.5.$$

For $x \in [0, 1.5]$, we have

$$\begin{aligned} g'_{0.5}(x) - h(x) &\geq \min_{x \in [0,1.5]} \Big[g'_{0.5}(x) - h(x) \Big] \geq \min_{x \in [0,1.5]} g'_{0.5}(x) - \max_{x \in [0,1.5]} h(x) \\ &= g'_{0.5}(1.5) - \frac{1}{0.5}. \end{aligned}$$

Table 3.4: Mean and standard deviations of the test likelihood ($\mathcal{L}_{\text{test}}$) and the computation time (T) on the three panel count data sets for GP4C, GP4CW and LocalEM over 40 runs. GP4CW outperforms GP4C and LocalEM in terms of the test likelihood. Mean and standard deviations of the test likelihood ($\mathcal{L}^{(1)}_{\text{test}} + \mathcal{L}^{(2)}_{\text{test}}$) after performing another round of training to reduce the variance caused by random split are provided in the fourth column.

Data Set	**METHOD**	$\mathcal{L}_{\text{test}}$	$\mathcal{L}^{(1)}_{\text{test}} + \mathcal{L}^{(2)}_{\text{test}}$	$T[s]$
Na-A	LocalEM	-492.1±306.1	-1272.7±288.6	1±0
	GP4C	-484.9±201.8	-1205.5±157.1	10±10
	GP4CW	**-179.2±81.3**	**-417.8±72.5**	8±9
Na-B	LocalEM	-473.2±212.2	-957.2±116.1	1±0
	GP4C	-411.0±184.3	-844.0±85.5	10±7
	GP4CW	**-152.7±60.6**	**-307.1±34.2**	16±13
Bl-A	LocalEM	-201.8±46.9	-421.4±44.3	1±0
	GP4C	-182.2±47.3	-378.6±17.9	25±9
	GP4CW	**-95.5±29.0**	**-206.3±5.2**	29±12
Bl-B	LocalEM	-313.1±54.2	-684.7±54.4	1±0
	GP4C	-310.4±54.9	-664.1±19.2	26±21
	GP4CW	**-212.4±50.1**	**-461.4±19.3**	36±23
Sk-A	LocalEM	-259.1±27.3	-519.8±6.2	39±3
	GP4C	-258.7±26.7	-519.1±2.7	33±6
	GP4CW	**-183.0±21.6**	**-366.4±1.5**	35±8
Sk-B	LocalEM	-198.1±47.1	-392.5±28.5	39±3
	GP4C	-191.2±42.5	-375.8±4.2	24±4
	GP4CW	**-105.7±19.7**	**-210.9±2.8**	27±5
Sk-C	LocalEM	-358.0±35.8	-733.6±11.4	47±4
	GP4C	-355.7±36.0	-728.4±6.3	21±12
	GP4CW	**-243.6±26.9**	**-498.2±2.3**	19±11
Sk-D	LocalEM	-200.9±31.9	-404.2±14.8	46±3
	GP4C	-198.9±30.6	-400.2±6.8	27±4
	GP4CW	**-118.9±14.3**	**-241.5±1.9**	31±4

□

Remark 2. *Lemma 3.6.1 is similar to the results in Moser [69]. Moser [69] proved that the following inequality holds when $m \in \mathbb{N}_+$.*

$$g'_m(x) \geq \frac{1}{x+m},\ x \geq 0. \tag{3.28}$$

However, we discover that a claim used in their proof is wrong. Namely, Moser [69] claimed that "two strictly convex and strictly decreasing functions can intersect at most twice". A counter-example is shown below.

$$f_1(x) = x^2 - \sin(x),\ f_2(x) = x^2 - \frac{\sin(2x)}{4},\ x \leq -\frac{1}{2}.$$

We can verify that

$$f_1'(x) = 2x - \cos(x) < 0,\ \ f_2'(x) = 2x - \frac{\cos(2x)}{2} < 0$$
$$f_1''(x) = 2 + \sin(x) > 0,\ \ f_2''(x) = 2 + \sin(2x) > 0.$$

Therefore, $f_1(x)$ and $f_2(x)$ are strictly decreasing and convex. However, the two curves intersect at $x = -\pi k, k \in \mathbb{N}_+$. We conjecture that the conclusion in Equation (3.28) *is still correct.*

Based on Lemma 3.6.1, we can show the following corollary.

Corollary 3.6.1. *The following inequality holds:*

$$g_m(x) \geq \ln(x+m) + \xi, m = \frac{1}{2},$$

where $\xi = \frac{3}{2}\Big(g_m'(1.5) - \frac{1}{m}\Big) - \ln(m) + g_m(0)$.

Proof. In Lemma 3.6.1,

$$g_m'(x) \geq \begin{cases} \frac{1}{x+m} & \text{if } x \geq 1.5, \\ \frac{1}{x+m} + g_m'(1.5) - \frac{1}{m} & \text{if } 0 \leq x \leq 1.5. \end{cases}$$

Integrating $g_m'(x)$ from 0 to ∞ yields

$$\begin{aligned} \int_0^\infty g_m'(x)dx &= \int_0^{1.5} g_m'(x)dx + \int_{1.5}^\infty g_m'(x)dx \\ &\geq \int_0^{1.5} \Big(g_m'(1.5) - \frac{1}{m}\Big)dx + \int_0^\infty h(x)dx \\ &= \frac{3}{2}\Big(g_m'(1.5) - \frac{1}{m}\Big) + \ln(x+m) - \ln(m). \end{aligned}$$

Therefore we have

$$g_m(x) \geq \ln(x+m) + \frac{3}{2}\Big(g_m'(1.5) - \frac{1}{m}\Big) - \ln(m) + g_m(0) = \ln(x+m) + \xi. \quad (3.29)$$

□

Finally, we can prove Theorem 3.3.1. Invoking Corollary 3.6.1, it is obvious that the inequality holds true for $b = 1$,

$$\begin{aligned} \mathbb{E}_Y[\ln Y^2] = \ln(2\sigma^2) + g_{0.5}\Big(\frac{\mu^2}{2\sigma^2}\Big) &\geq \ln(2\sigma^2) + \ln\Big(\frac{\mu^2}{2\sigma^2} + \frac{1}{2}\Big) + \xi \\ &= \ln(\mu^2 + \sigma^2) + \xi. \end{aligned} \quad (3.30)$$

This implies that the inequality holds true for all values of $b \in [0, 1]$.

Chapter 4

Recurrent Event Data: Bayesian Nonparametric Poisson Process Allocation

In this chapter, we study how to model the diversity among multiple time-sequences. As the beginning, the time-sequence appears in the form of recurrent events. We present the Bayesian nonparametric Poisson process allocation (BaNPPA), a latent-function model for time-sequences, which automatically infers the number of latent functions.

4.1 Introduction

When modeling a collection of time-sequences, a key idea is to cluster the data into groups while allowing the groups to remain linked so as to share statistical strengths among them [94]. Several models have been proposed on the basis of this simple idea, e.g., the convolution process [38], nonnegative matrix factorization (NMF) [66], and latent Poisson process allocation (LPPA) [61]. These models employ latent factors to share statistical strengths and combine these functions to model the correlations within and among time-sequences.

Among these models, LPPA is a powerful approach because it uses latent functions obtained from a Gaussian process (GP). Such continuous latent functions are able to flexibly model complex structures in the data, and do not require a careful discretization such as that used in NMF.

However, a limitation of LPPA is that the number of latent functions needs to be set beforehand. If the chosen number is much larger than the actual number of latent functions required to explain the data, LPPA will still use all the latent functions. There is no mechanism in LPPA to prevent this "spread" of allocation, which creates a problem when our goal is to understand the reasons behind the events observed in the data. For example, this might make it difficult to explain the retweet patterns in Twitter where a sudden avalanche of retweets is quite common [32]. For such cases, LPPA will simply use all its latent functions to explain these spiky patterns.

In theory, the above problem can be solved by using Bayesian nonparametric (BNP) methods [42] which can automatically determine the number of relevant latent functions. However, as we show in this chapter, a direct application of existing BNP methods to LPPA is challenging. An obvious issue is that such an application typically requires the use of Markov Chain Monte Carlo (MCMC) algorithms which are slow to converge for large data sets. A more essential and technically intricate issue is that a naive application of BNP methods to LPPA suffers from an unidentifiability issue because the GP-modulated latent functions are not normalized. Unidentifiability is bad news when our focus is to understand the reasons behind the events.

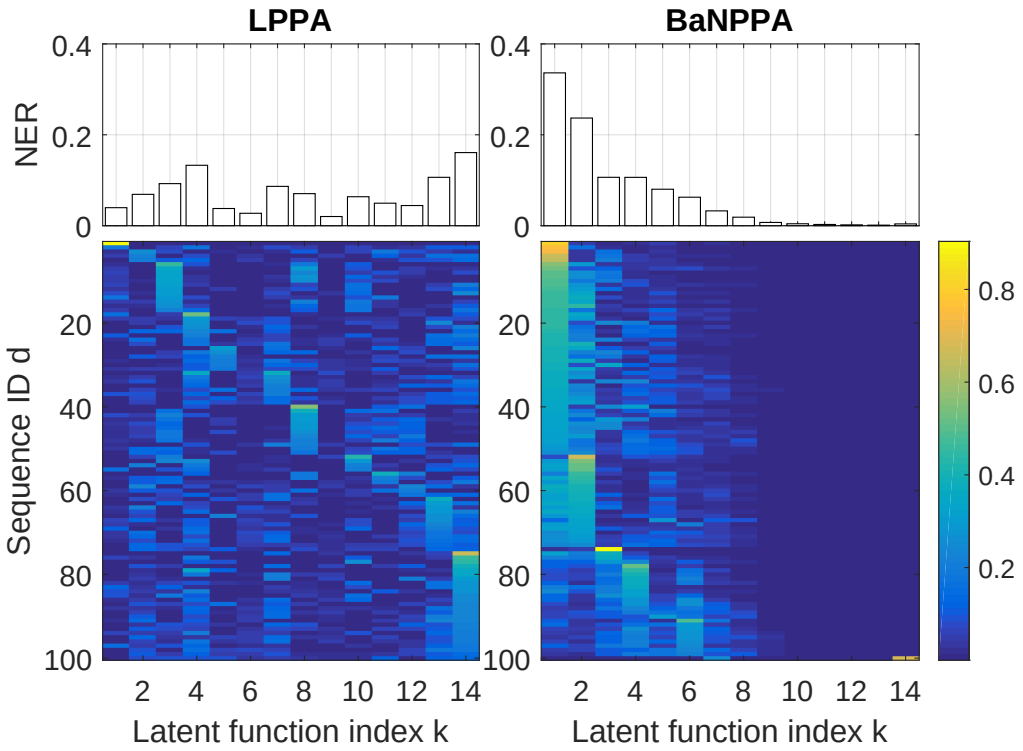

Figure 4.1: This figure illustrates that, even when a large number of latent functions are provided, BaNPPA automatically selects only a few to explain the data, while LPPA uses them all. The bottom plots show the weights of the latent functions for the Microblog dataset, where we see that BaNPPA assigns zero weights to many latent functions, while LPPA assigns every latent function to at least a few time-sequences. The top plots show a score which measures the average responsibility of the latent functions. See Section 4.5 for details.

In this chapter, we propose a new model to solve these problems. Our model, which we call the *Bayesian nonparametric Poisson process allocation* (BaNPPA) model, enables automatic inference of the number of latent functions while retaining the accuracy, interpretability, and scalability of LPPA. Unlike hierarchical models [94] which promote sharing through a common base measure, latent functions in our model are shared across all time-sequences due to the size-biased ordering which promotes sharing by penalizing latent functions that belong to higher indices [34, 76]. The size-biased ordering restricts the use of all latent functions. Figure 4.1 illustrates this on a real data set.

We propose a computationally efficient variational inference algorithm for BaNPPA and solve the unidentifiability issue by adding a constraint within the inference algorithm to regulate the volume of each latent function. Overall, we present a scalable and accurate Bayesian nonparameteric approach for time-sequence modeling. Figure 4.2 shows an example of the results obtained with BaNPPA on a real-world dataset.

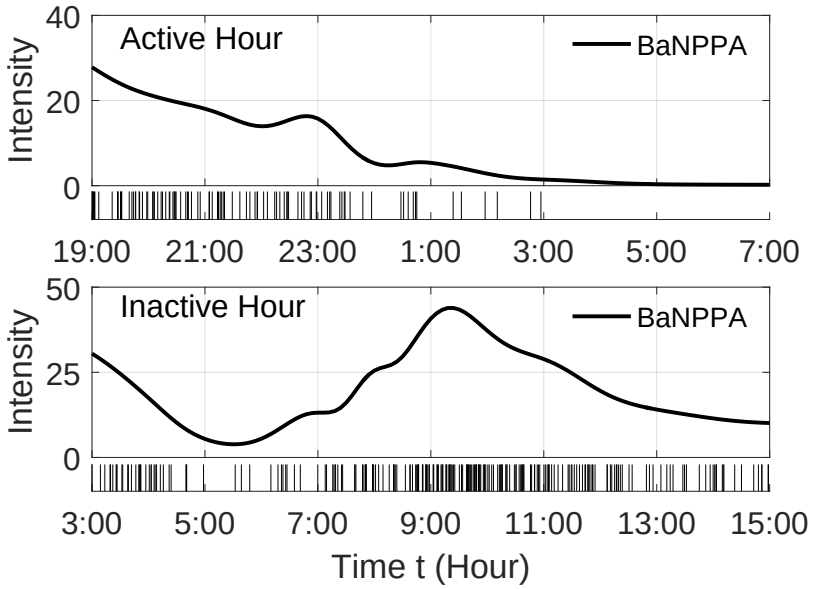

Figure 4.2: Illustrations of intensity functions obtained with BaNPPA on the Microblog dataset. Each plot shows a time-sequence (with small bars at the bottom) and the corresponding estimated intensity function (with solid lines). The top and bottom plots are for tweets posted during active and inactive hours of the day, respectively.

4.2 Time-Sequence Modeling and Its Challenges

Our goal is to develop a flexible model for time-sequences. We consider time-sequence that contains a set of time-stamps which record the occurrence of events. In this chapter, we study the recurrent event data and the data set $\mathcal{D} = \{\boldsymbol{d}_k\}_{k=1}^K$. Each subject will generate a sequence of events in an observation window $\mathcal{X}^{(k)} \subset \mathbb{R}$. We assume that all observation windows are the same $\mathcal{X}^{(k)} = \mathcal{X}$.

In the recurrent event data, the time-stamp of each event is a scalar and is fully observable. The time-sequence data from the kth subject can be represented as follows:

$$\boldsymbol{d}_k \triangleq \left\{x_j^{(k)} \in \mathcal{X}\right\}_{j=1}^{N_k}. \tag{4.1}$$

A common approach to model such time-sequences is to use the temporal Cox process [2, 60] which uses a stochastic intensity function $\lambda(t) : \mathbb{R}^+ \to \mathbb{R}^+$ to model the arrival times [52]. Given the intensity function $\lambda(t)$ and the observation window $\mathcal{X}$, the likelihood of the sequence $\boldsymbol{d}_k$ is given by Theorem 2.5.3.

$$\mathrm{P}(\mathbf{d}_k|\lambda_k(x)) = \exp\Big(-\int_{\mathcal{X}} \lambda_k(s)ds\Big) \prod_{j=1}^{N_k} \lambda_k(x_j^{(k)}). \tag{4.2}$$

In LPPA, to model multiple time-sequences, the kth time-sequence is assumed to be generated by a temporal Cox process with an intensity function $\lambda_k(x)$ which is modeled as follows:

$$\lambda_k(x) = \sum_{l=1}^{L} \theta_{kl} f_l^2(t), \quad \theta_{kl} \geq 0, \tag{4.3}$$

where $f_l(x)$ is a function drawn from a GP prior, θ_{kl} is its weight, and L is the number of basis latent functions. To ensure the non-negativity of $\lambda_k(x)$, $f_l(x)$ are squared and weights θ_{kl} are required to be non-negative. A more detailed review of the LPPA model is given in Section 2.6.4.

LPPA is a powerful approach which also enables scalable inference. Due to the GP prior, LPPA is capable of generating intensity functions with complex shapes. Scalable inference is made possible by using a set of pseudo inputs [96]. The overall computational complexity is $O(LNM^2)$, where N is the total number of events in $\mathcal{D}$ and M is the number of pseudo inputs.

One issue with LPPA is that L needs to be set beforehand. This not only increases the computation cost, but also creates a serious interpretability issue which is undesirable when our goal is to understand the reasons behind the data. Specifically, when the number of latent functions is much larger than what it needs to be, LPPA uses all of them, making it difficult to interpret the results. We give empirical evidence in support of this claim and correct this behavior by using a BNP method.

Unfortunately, a direct application of the existing BNP methods increases the computation cost and limits the flexibility of the model. The problem lies in the strict requirement that the latent functions needs to be a *normalized density function*, i.e., a function with a volume[1] equal to 1. For example, previous studies, such as Kottas [53], Ihler and Smyth [45], model the intensity functions with the following Dirichlet process mixture model,

$$\lambda_k(x) = s_k \sum_{l=1}^{\infty} \theta_{kl} \tilde{f}(t; \psi_l), \tag{4.4}$$

where $\tilde{f}$ are normalized density functions with parameters ψ_l and the weights θ_{kl} are non-negative and sum to one $\sum_{l=1}^{\infty} \theta_{kl} = 1$ ($s_k > 0$ is the rate parameter that models the number of events $N(\mathcal{X})$). Since the function $\tilde{f}$ needs to be normalized, the choices are limited to well-known density function which may not be very flexible to model complex time-sequences, e.g., Kottas [53] used the beta distribution and Ihler and Smyth [45] used the truncated Gaussian distribution.

In addition, such models require MCMC sampling algorithms which usually converge slowly on large data sets. To the best of our knowledge, it is still unclear how to build a nonparametric prior for such normalized density functions while enabling scalable inference, e.g., via variational methods.

We propose a nonparameteric model, called the Bayesian nonparameteric Poisson process allocation (BaNPPA), which avoids the need to explicitly specify the number of latent functions while retaining the flexibility and scalability of the LPPA model. Our method combines the models shown in Equation (4.3) and (4.4). We show that this direct combination has an unidentifiability issue, and we fix the issue within a variational-inference algorithm. Our approach therefore combines the strengths of the LPPA and BNP models while keeping their best features.

4.3 Bayesian Nonparametric Poisson Process Allocation

As discussed earlier, we need to set the number of latent functions beforehand for LPPA. We fix this issue by proposing a new model called BaNPPA that combines the non-parametric model of Equation (4.4) with the LPPA model shown in Equation (2.49).

[1]The volume of a function $f(t), t \in \mathcal{X}$ is defined as the integral $\int_{\mathcal{X}} f(t)dt$.

Specifically, we let $\tilde{f}$ in Equation (4.4) to be equal to $f_l^2(t)$, as follows:

$$\lambda_k(x) = s_k \sum_{l=1}^{\infty} \theta_{kl} f_l^2(x), \text{ where } s_k, \theta_{kl} > 0, \sum_{l=1}^{\infty} \theta_{kl} = 1. \tag{4.5}$$

Similar to LPPA, we draw functions $f_l(x)$ from a Gaussian process. We draw the weights θ_{kl} using a stick-breaking process, and use a Gamma distribution prior for the scalar rate parameter s_k. The final generative model of BaNPPA is shown in Algorithm 12.

Algorithm 12: The generative process for BaNPPA.

Input : The number of the time-sequences K, the hyper-parameters in the gamma distribution $\{a_0, b_0\}$, the stick-breaking process prior α, the mean value m_0, the covariance functions in L Gaussian processes $\{\kappa_l\}$ and a time window $\mathcal{X} = (0, T]$.

Output: The time-sequence data $\mathcal{D} = \{\boldsymbol{d}_k\}_{k=1}^K$.

for *each basis function* $l = 1, \ldots, \infty$ **do**
  Sample $f_l \sim \mathcal{GP}(m_0(x), \kappa_l(x, x'))$.
end
for *each subject* $k = 1, \ldots, K$ **do**
  Sample $\theta'_{kl} \sim \text{Beta}(1, \alpha)$.
  Calculate the mixture weight $\theta_{kl} = \theta'_{kl} \prod_{j=1}^{l-1}(1 - \theta'_{kj})$.
  Sample $s_k \sim \text{Gamma}(a_0, b_0)$.
  Calculate the intensity function.

$$\lambda_k(x) = s_k \sum_{l=1}^{\infty} \theta_{kl} f_l^2(t). \tag{4.6}$$

  Sample $\boldsymbol{d}_k \sim \text{IPP}(\lambda_k(x))$ on the time window $(0, T]$.
end

IPP$(\cdot)$ is the thinning algorithm for an inhomogeneous Poisson process and is given in Algorithm 5. In the model, we denote a beta distribution with shape parameters a and b by Beta(a, b) and a gamma distribution with shape parameter a and rate parameter b by Gamma(a, b).

The above model automatically determines the number of latent functions due to the size-biased ordering [76] obtained by using the stick-breaking process.

Both the latent functions $\{f_l^2(x)\}$ and the weights $\{\theta_{kl}\}$ use the same set of indices $l = 1, \ldots, \infty$. This implies that when generating the kth time-sequence, the latent function at a lower index l is more likely to be assigned a larger weight θ_{kl}. This encourages the model to use some latent functions more than the others.

Unfortunately, the above model is unidentifiable. This is because, unlike the nonparametric model of Equation (4.4), the latent functions $\{f_l^2\}$ are unnormalized, and therefore many combinations of s_k, $\{\theta_{kl}\}$ and $\{f_l\}$ might give us the same model. For example, the following transformation gives the same intensity function for any $\epsilon_k > 0$:

$$s_k \bar{\epsilon}_k, \left\{ \frac{\theta_{kl} \epsilon_l}{\bar{\epsilon}_k} \right\}, \left\{ \frac{f_l}{\sqrt{\epsilon_l}} \right\},$$

where $\bar{\epsilon}_k := \sum_{l=1}^{\infty} \theta_{kl} \epsilon_l$. We can check this by substituting the triplet in Equation (4.5). Since the volume of each f_l is not regulated, we can move the "mass" around between the components of the model.

This type of unidentifiability is problematic when our goal is to understand the reasons behind the patterns in the data. In our experiments, we observe that this leads to a shrinkage of the latent functions which affects interpretability as well as the quality of the estimated hyperparameters. In Section 4.4.2, we propose a way to fix this issue by adding a constraint on the volume of the latent function.

There is also another common identifiability problem in such mixture models. Lloyd et al. [61] claimed that LPPA is unidentifiable and non-unique since there may be multiple decompositions that are well supported by the data. In BaNPPA, due to the ordering constraints imposed by size-biased ordering, this unidentifiability issue is reduced.

We also need to guarantee that the expected intensity function at any time $\mathbb{E}[\lambda_k(x)]$ is finite. This can be achieved by fixing the GP hyperparameters. For example, assuming an automatic relevance determination (ARD) covariance functions in Equation (2.10), we can fix the hyperparameters g and c_l, which ensures that the mean and variance of each latent function f_l are finite. In that case, the value of $\mathbb{E}[\lambda_k(x)]$ is bounded due to the following relation:

$$\begin{aligned}\mathbb{E}[\lambda_k(x)] &= \mathbb{E}\left[s_k \sum_{l=1}^{\infty} \theta_{kl} f_l^2(x)\right] \leq \mathbb{E}[s_k] \max_l \mathbb{E}[f_l^2(x)] \\ &= \frac{a_0}{b_0} \max_l \Big(\mathbb{E}^2[f_l(t)] + \mathrm{Var}[f_l(t)]\Big). \end{aligned} \tag{4.7}$$

4.4 Inference

In this section, we first describe the general variational inference framework and provide a solution to the identifiability issue in Section 4.4.2. A derivation of the evidence lower bound (ELBO) and its derivatives are provided in Section 4.6.

4.4.1 Variational Inference

Denote $\boldsymbol{s} \triangleq \{s_k\}$, $\Theta \triangleq \{\theta'_{kl}\}$ and $\boldsymbol{f} \triangleq \{f_l\}$. Let $\boldsymbol{H}$ be the set of hyperparameters of the GP covariance function. The joint distribution of BaNPPA can be expressed as

$$\begin{aligned} p(\mathcal{D}, \boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{f}) = &\prod_{k=1}^{K} p(\boldsymbol{d}_k | \boldsymbol{f}, \boldsymbol{\theta'_k}, s_k) \prod_{k=1}^{K} \prod_{l=1}^{\infty} p(\theta'_{kl}; \alpha) \\ &\times \prod_{k=1}^{K} p(s_k; a_0, b_0) \prod_{l=1}^{\infty} p(f_l; \boldsymbol{m}_0, \boldsymbol{H}). \end{aligned}$$

We approximate the posterior distribution over Θ and $\boldsymbol{f}$, while computing a point estimate of $\boldsymbol{s}$. We follow Blei et al. [11] to truncate the number of latent functions to L which we select to be larger than the expected number of latent functions used by the data.

For the GP part, we use the same set of pseudo inputs $\{\bar{x}_m\}_{m=1}^{M}$, $M < N$ for each f_k to reduce the number of variational parameters [61]. Denote $\bar{\boldsymbol{f}}_l$ to be the vector $[f_l(\bar{x}_1), \ldots, f_l(\bar{x}_M)]^\top$, $\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}$ to be a covariance matrix whose i,j'th entry is equal to $\kappa_l(\bar{x}_i, \bar{x}_j)$, and $\boldsymbol{m}_0 \in \mathbb{R}^M$ to be a vector all of whose elements are equal to m_0. We choose the following forms for the variational distributions

of θ'_{kl} and $\bar{\boldsymbol{f}}_l$:

$$q(\theta'_{kl}) = \begin{cases} \text{Beta}(\tau_{kl,0}, \tau_{kl,1}) & \text{if } l < L, \\ \delta_1 & \text{if } l = 1. \end{cases}$$
$$q(\bar{\boldsymbol{f}}_l) = \mathcal{N}(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l),\ l = 1, \ldots, L.$$
$$q(s_k) = \delta_{\eta_k},\ k = 1, \ldots, K.$$

δ_x is a Dirac measure in Definition 2.3.1, and $\boldsymbol{\mu}_l$ and $\boldsymbol{\Sigma}_l$ are the mean and covariance of a Gaussian distribution. Following Lian et al. [57], we use the re-parametrization $\boldsymbol{\Sigma}_l = \boldsymbol{L}_l \boldsymbol{L}_l^\top$ by Cholesky decomposition and add positivity constraints to the diagonal elements in L_k during the optimization procedure.

Using the approximation of Titsias [96] and a mean-field assumption over Θ, we can use the following final variational distribution:

$$q(\boldsymbol{f}, \{\bar{\boldsymbol{f}}_l\}_{l=1}^L, \Theta) \triangleq \prod_{l=1}^{L} p(f_l|\bar{\boldsymbol{f}}_l) q(\bar{\boldsymbol{f}}_l) \prod_{k=1}^{K} \prod_{l=1}^{L} q(\theta'_{kl}) \prod_{k=1}^{K} q(s_k).$$

Denoting $\boldsymbol{\tau} \triangleq \{(\tau_{kl,0}, \tau_{kl,1})\}$ and $\boldsymbol{L} \triangleq \{L_l\}$, we get the following set of variational parameters and hyperparameters to be optimized:

$$\Phi = \{\boldsymbol{\tau}, \{\boldsymbol{\mu}\}_{l=1}^L, \{\boldsymbol{L}_l\}_{l=1}^L, \boldsymbol{H}, a_0, b_0, \alpha, \boldsymbol{s}\}.$$

4.4.2 An Alleviation Solution to the Identifiability Problem

So far, the framework seems very traditional. However, as we mentioned in Section 4.3, this model has an additional identifiability problem which might make interpretability difficult. In this section, we propose a solution to alleviate this issue.

A straightforward option is to directly impose a constraint on the volume of the latent functions $\int_{\mathcal{X}} f_l^2(x)dx$, where f_l is drawn from the posterior process $p(f_l|\mathcal{D})$. However this is intractable. In order to obtain a tractable constraint, we could instead impose a constraint on the following expectation:

$$\iint_{\mathcal{X}} p(f_l|\mathcal{D}) f_l^2(s) ds df_l = A,\ l = 1, \ldots, L, \tag{4.8}$$

where A is a positive constant. Within the variational inference framework, we use the variational distribution $q(f_l)$ to approximate the posterior $p(f_l|\mathcal{D})$, and add the following constraint to each latent function:

$$\iint_{\mathcal{X}} q(f_l) f_l^2(s) ds df_l = A, \quad l = 1, \ldots, L. \tag{4.9}$$

The above constraint can be easily computed unlike the volume constraint on the function f_l. In our experiments, we set $A = N/K$ where N is the total number of events in the data and K is the number of time-sequences in $\mathcal{D}$.

4.4.3 Optimization with Equality Constraints

Given the equality constraints in Equation (4.9), the optimization process can be formulated as follows, where we denote the ELBO as $\mathcal{L}_1(\Phi)$:

$$\max_{\Phi}\ \mathcal{L}_1(\Phi) \quad \text{s.t. } h_l(\Phi) = 0,\ l = 1, \ldots, L, \tag{4.10}$$
$$h_l(\Phi) = \int_{\mathcal{X}} \mathbb{E}_q[f_l^2(s)] ds - A.$$

Problem (4.10) is an optimization problem with equality constraints and we use the augmented Lagrangian method [8] to transform Problem (4.10) into a series of related optimization problems indexed by i:

$$\max_{\Phi} \ \mathcal{L}_1(\Phi) - \sum_{l=1}^{L} \Big(w_{il} h_l(\Phi) + \frac{1}{2} v_{il} h_l^2(\Phi) \Big), \tag{4.11}$$

where $\{w_{il}\}$ is a bounded sequence and $\{v_{il}\}$ is a non-negative monotonically-increasing sequence with respect to i. We denote this objective $\mathcal{L}_{\boldsymbol{v_i}}(\Phi, \boldsymbol{w_i})$. For each optimization problem in Equation (4.11), $\mathcal{L}_{\boldsymbol{v_i}}(\Phi, \boldsymbol{w_i})$ is still upper bounded.

Theorem 4.4.1. *Each optimization problem is upper bounded.*

$$\mathcal{L}_{\boldsymbol{v_i}}(\Phi, \boldsymbol{w_i}) \leq \ln p(\mathcal{D}) + \sum_{l=1}^{L} \frac{w_{il}^2}{2 v_{il}}, \quad i \in \mathbb{N}^+.$$

Proof. $\mathcal{L}_1(q)$ can be easily bounded by variational inference framework

$$\mathcal{L}_1(q) \leq \ln p(\mathcal{D})$$

Let $h_{il} = \int_{\mathcal{X}} \mathbb{E}_q[f_l^2(s)] ds - A$, and then we have

$$\begin{aligned} &\sum_{l=1}^{L} w_{il} \Big(\int_{\mathcal{X}} \mathbb{E}_q[f_l^2(s)] ds - A \Big) + \sum_{l=1}^{L} \frac{v_{il}}{2} \Big(\int_{\mathcal{X}} \mathbb{E}_q[f_l^2(s)] ds - A \Big)^2 \\ =& \sum_{l=1}^{L} (w_{il} h_{il} + \frac{v_{il}}{2} h_{il}^2) \geq \sum_{l=1}^{L} \frac{w_{il}^2}{2 v_{il}} \end{aligned}$$

Combining these two parts finishes the proof. □

Thus if we use coordinate ascent with respect to Φ, the algorithm is guaranteed to arrive at a local maximum. To set $\boldsymbol{v}_{il}$ and $\boldsymbol{w}_{il}$, we follow the suggestions from Bertsekas [8], and set $\boldsymbol{v}_{i+1,l} = 4\boldsymbol{v}_{il}$ and $\boldsymbol{w}_{i+1,l} = \boldsymbol{w}_{il} + \boldsymbol{v}_{il} \boldsymbol{h}_l(\Phi_i)$. We initialize $v_{1l} = 4, w_{1l} = 1, \forall l$.

4.4.4 Computational Complexity

Optimization problems shown in Equation (4.11) are not significantly more expensive than the original optimization problem. Although in Equation (4.11), we have to optimize additional parameters, the bottleneck is still the matrix-matrix multiplication in the evaluation of $q(f_l)$. For one iteration of the training procedure, the computational complexity is $\mathcal{O}(LNM^2)$, which is the same as LPPA.

One might expect that the total computational complexity of our algorithm is worse than LPPA because we have to solve a sequence of problems. We find that "warm starts" are very effective in improving the convergence of our algorithm [8]. Namely, we reuse the final value Φ_{i-1} of the previous optimization as the starting value for the i'th round and terminate the training process when the relative change in the likelihood is small. In our experiments, we observed that the convergence of BaNPPA is rather fast and comparable to LPPA.

Table 4.1: Data sets used for the experiments. Here, D is the number of time-sequences, N_{train} and N_{test} are the total number of events in the training and test set respectively, and $\mathcal{X}$ is the time window.

Data set	D	N_{train}	N_{test}	$\mathcal{X}$
Synthetic A	200	6,304	6,010	[0,60]
Synthetic B	250	8,074	8,110	[0,80]
Microblog	500	44,628	44,352	[3,15]
Citation	600	106,113	106,340	[0,20]

4.5 Experiments

In this section, We evaluate our proposed BaNPPA model and compare it with the benchmark methods on both synthetic and real-world data sets. The algorithms are programmed in Matlab R2015b and run on an Intel Xeon E5-2667 CPU with a memory of 64GB. The code to reproduce our experiments can be found at github.com/Dinghy/BaNPPA.

4.5.1 Experiment Settings

First we show the settings for all experiments in the following experiments.

Benchmark

We compare our proposed BaNPPA model with two other models.

1. **LPPA**. We re-implement the LPPA model based on the descriptions in Lloyd et al. [61].
2. **BaNPPA-NC**. To measure the effect of adding the constraint shown in Equation (4.9), we also compare to a variant of BaNPPA which does not contain any constraints. We call it BaNPPA with No Constraints, i.e., BaNPPA-NC.

Data Sets

We test the three methods on two synthetic and two real-world data sets. Table 4.1 summarizes the overall statistics and we give detailed information below.

- **Synthetic A**. We sample 200 time-sequences from a mixture of 4 latent functions $\tilde{f}(x;\psi_l)$ shown in the top plot of Figure 4.3. The intensity function for the kth time-sequence is generated as follows:

$$
\begin{aligned}
s_k &\sim \text{Gamma}(2,3),\\
\theta_k &\sim \text{Dir}(1.2,1,0.8,0.6),\\
\tilde{f}(x;\psi_l) &\propto \exp\Big(-\frac{(x-15+10l)^2}{10}\Big)+\exp\Big(-\frac{(x-55+10l)^2}{10}\Big),\\
\lambda_k(x) &= s_k\sum_{k=1}^{4}\theta_{kl}\tilde{f}(x;\psi_l),\ x\in[0,60].
\end{aligned}
$$

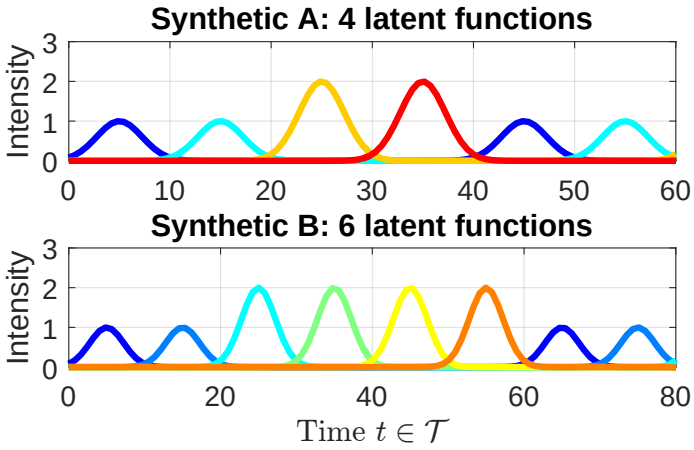

Figure 4.3: Latent functions used to create synthetic data set A and B are shown in the top and bottom plots, respectively. In both the data sets, there are two latent functions with two modes while the rest have only one mode. Different colors indicate different latent functions.

Here $\mathrm{Dir}(\cdot)$ denotes the Dirichlet distribution. Each $\tilde{f}(t;\psi_l)$ is either a Gaussian distribution or a mixture of two Gaussian distributions normalized by its integral.

- **Synthetic B**. This data set is similar to Synthetic A but there are 6 latent functions shown in the bottom plot of Figure 4.3. The intensity function for the kth time-sequence is generated as follows:

$$
\begin{aligned}
s_k &\sim \mathrm{Gamma}(2,3),\\
\theta_k &\sim \mathrm{Dir}(1.2,1,0.8,0.6,0.5,0.5),\\
\tilde{f}(x;\psi_l) &\propto \exp\Big(-\frac{(x-15+10l)^2}{10}\Big)+\exp\Big(-\frac{(x-75+10l)^2}{10}\Big),\\
\lambda_k(x) &= s_k\sum_{l=1}^{6}\tilde{f}(x;\psi_l),\ \ x\in[0,60].
\end{aligned}
$$

Each $\tilde{f}(t;\psi_l)$ is either a Gaussian distribution or a mixture of two Gaussian distributions normalized by its integral.

- **Microblog data set**. This data set contains 500 tweets and all retweets of each tweet from 7 tweet-posters on Sina micro-blog platform obtained through the official API[2]. Two examples are shown in Figure 4.2. Through time-sequence modeling, we can try to understand the retweet patterns. For example, one reason could be that the tweets posted at an inactive hour (late at night) will regain the attention from the followers several hours later next morning [21, 32]. BaNPPA could help us understand such reasons as illustrated in Figure 4.2.

- **Citation data set**. This data set contains the Microsoft academic graph until February 5th, 2016 obtained from the KDDcup 2016 [3]. The original

[2]http://open.weibo.com/wiki/Oauth/en
[3]https://kddcup2016.azurewebsites.net/

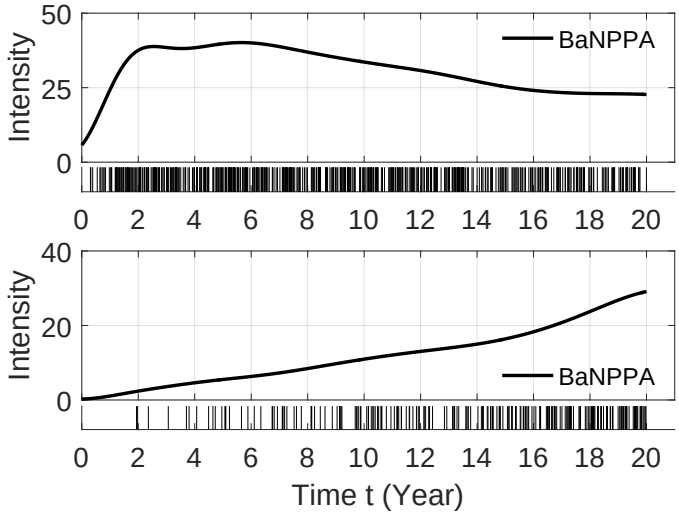

Figure 4.4: **Citation data set**. Top: A paper which slowly gets citation and becomes popular many years later. Bottom: A paper which quickly gets citation after being published. Smooth lines are the mean intensity function inferred from LPPA and BaNPPA. Small bars is the time of each citation. The x-axis indicates the time in year after publication.

data set contains 126,909,021 papers and we use a subset of it. Time-sequence modeling can be used to understand the patterns of citations, e.g., some papers quickly get citations while some others get it slowly.

Evaluation Metrics

We evaluate the methods using the test likelihood and visualize the result of the mixture weights by NER and UNER.

- **Test Likelihood**. To measure the predictive performance, we follow Lloyd et al. [60] and use the following approximation to the test likelihood which we denote by

$$\mathcal{L}_{\text{test}}(\mathcal{D}_{\text{test}}, \Theta, \boldsymbol{s}) \triangleq \sum_{k=1}^{K} \sum_{n=1}^{N_{\text{test}}^k} \ln \left(s_k \sum_{l=1}^{L} \theta_{kl} \exp \left(\mathbb{E}_q(\ln f_l^2(x_n^{(k)})) \right) \right) - \sum_{k=1}^{K} s_k \sum_{l=1}^{L} \theta_{kl} \int_{\mathcal{X}} \mathbb{E}_q[f_l^2(s)] ds. \tag{4.12}$$

  This is a lower bound to the test likelihood $\ln p(\mathcal{D}_{\text{test}}|\mathcal{D}_{\text{train}})$ and a higher value means a better approximation of the test likelihood. For LPPA, the allocation parameters θ_{kl} are the point-estimated weights and the rate parameter $s_k = 1$. For BaNPPA and BaNPPA-NC, we report averaged value over $q(\boldsymbol{\theta}_k)$. We can compute a similar approximation $\mathcal{L}_{\text{train}}$ on the training data. A detailed derivation can be found in Section 4.6.3.

- **NER and UNER**. To visualize the responsibility of each latent function in the model, we first define the normalized allocation matrix $\hat{\Theta} \in \mathbb{R}_+^{D \times K}$

whose (d, k)'th entry is equal to,

$$\hat{\theta}_{kl} = \frac{\mathbb{E}_q[\theta_{kl} \int_{\mathcal{X}} f_l^2(s)ds]}{\sum_{m=1}^{L} \mathbb{E}_q[\theta_{km} \int_{\mathcal{X}} f_m^2(s)ds]}. \tag{4.13}$$

The normalization in the above matrix tries to remove the unidentifiability introduced due to the unconstrained volume of the latent functions in LPPA and BaNPPA-NC. Based on $\hat{\Theta}$, we can compute a normalized score that can measure the responsibility of each latent function. We define the normalized expected responsibility (NER)

$$\hat{v}_l = \frac{1}{K} \sum_{k=1}^{K} \hat{\theta}_{kl},\ l = 1, \ldots, L.$$

A larger NER indicates that the corresponding latent function is more often occupied by the model. Another measure is the unnormalized expected responsibility (UNER)

$$v_l = \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}_q[\theta_{kl}],\ l = 1, \ldots, L,$$

which omits the contribution of the volume of $f_l^2(x)$.

Optimization Settings

Our goal is to measure the improvements obtained with the automatic inference of L using BaNPPA. To do so, we fix L to 14 for BaNPPA and BaNPPA-NC, and compare them to LPPA with a range of values for L. We expect BaNPPA to give a comparable performance to the best setting of L in LPPA.

We use a random initialization for the allocation matrix Θ and $\boldsymbol{\tau}$. We use 18, 24, 30 and 30 pseudo inputs for the four data sets, respectively. We follow the common practice and add a jitter term εI to the covariance matrix $\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}$ to avoid numerical instability [4]. For hyper-parameters a_0 and b_0 in the gamma distribution, we use the counts of events $\{N_{\text{train}}^k\}_{k=1}^K$ to initialize (a_0, b_0).

To maintain the positivity constraints on $\boldsymbol{L}$ and $\boldsymbol{\tau}$, we use the limited-memory projected quasi-Newton algorithm [85]. For BaNPPA, we stop the training process when the relative change between $\mathcal{L}_{\boldsymbol{v_i}}(\Phi_i, \boldsymbol{w_i})$ and $\mathcal{L}_{\boldsymbol{v_{i+1}}}(\Phi_{i+1}, \boldsymbol{w_{i+1}})$ is less than 10^{-3}. For other methods, we terminate the training process when the relative change in ELBO is less than 10^{-3}.

4.5.2 Experiment Results

We conduct experiments for two different methods of setting the hyper-parameter α. In the first method, we learn α within a VB-EM framework (initialize $\alpha = 1$, see details in Section 4.6). In the second method, we do not learn α rather fix it to one of the value in the set $\{1.1, 2, 4, 6, 8\}$. In both methods, all experiments were repeated five times.

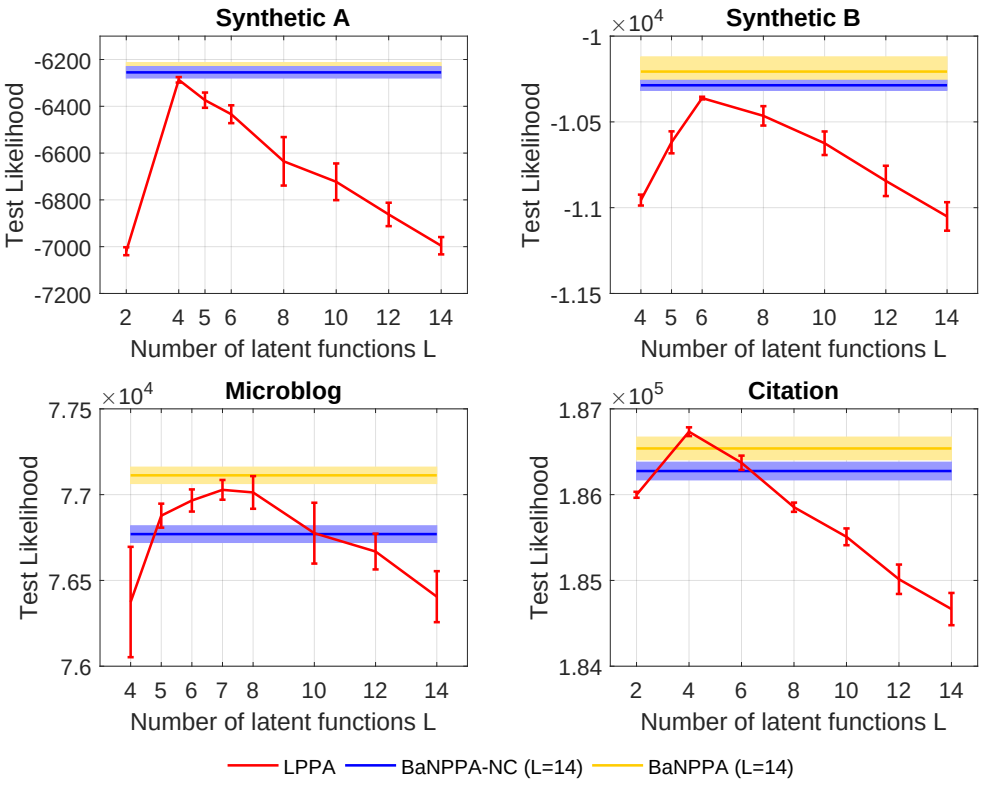

Figure 4.5: BaNPPA gives the best test-likelihoods (higher is better) and performs comparably to the best setting of L for LPPA. For BaNPPA and BaNPPA-NC, we use a fixed value of $L = 14$. Error bars and shaded areas show the 95% confidence intervals.

Results When Optimizing the Hyper-parameter α

Figure 4.5 shows the comparison of the test-likelihoods when optimizing α. The test likelihood of LPPA drops when increasing the number of latent functions L. As desired, BaNPPA achieves comparable results to the best setting of L in LPPA. BaNPPA-NC also performs well but slightly worse than BaNPPA.

The comparison of the train likelihood $\mathcal{L}_{\text{train}}$ when we optimize the hyper-parameter α is given in Figure 4.6. We can notice that for LPPA, the train likelihood keeps increasing when we increase L. This is also a sign of over-fitting.

The comparison of the final optimized α is given in Figure 4.7. We notice that for BaNPPA model, it can achieve a relatively smaller α. However, for BaNPPA-NC model, since there are no regulations on the volume of the intensity functions, the optimized α learned from the mixture weights θ_{kl} might be inaccurate.

We plot the change of the training likelihood in one trial in Figure 4.8. For the total computational complexity, both BaNPPA-NC and BaNPPA take more computation time but are still comparable to LPPA. Two reasons account for this fact. One is that there are more parameters to be optimized in BaNPPA and BaNPPA-NC and the other is that BaNPPA potentially has an infinite number of problems to be solved. In Figure 4.8, we can notice that the training likelihood for BaNPPA and the training likelihood for BaNPPA-NC stabilize rather quickly.

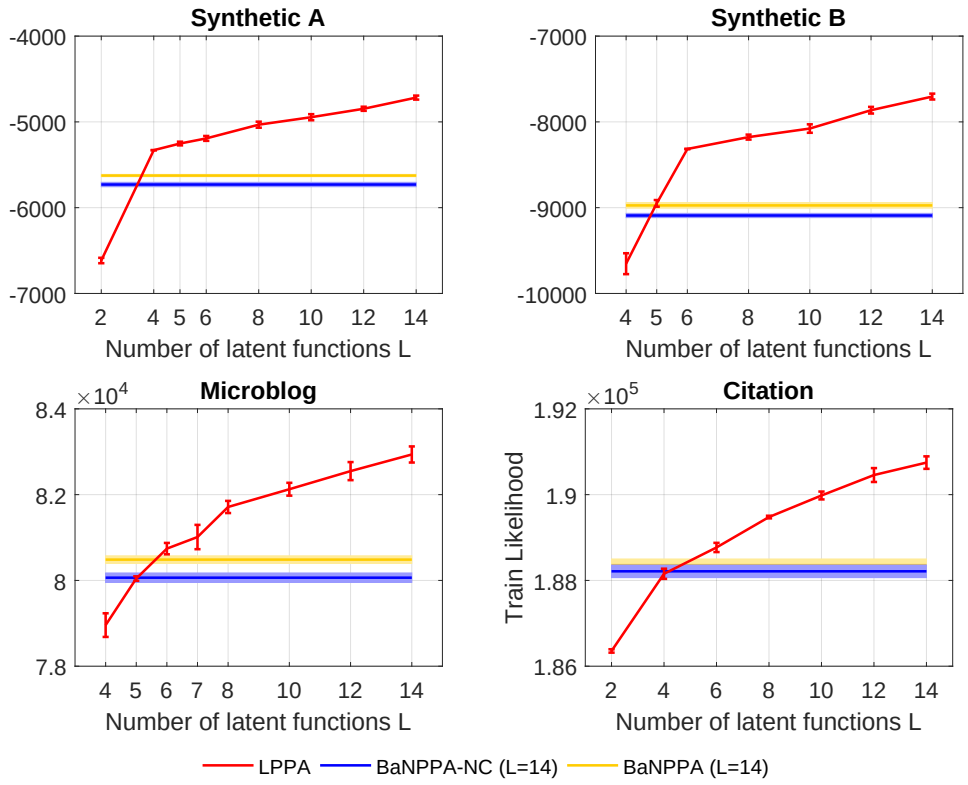

Figure 4.6: The comparison of the train likelihood for three algorithms. For LPPA, we change the number of latent functions L. For BaNPPA and BaNPPA-NC, we fix $L = 14$ and optimize the hyper-parameter α using the VB-EM framework. Error bars and shaded area represent the 95% confidence intervals.

Results When Fixing the Hyper-parameter α

Figure 4.9 shows the comparison of the test-likelihoods when α is fixed. In Figure 4.9, when increasing α, the performance of BaNPPA stays relatively stable and comparable to the best setting of LPPA. The performance of BaNPPA-NC however degrades with increasing α for all data sets. This shows that the volume constraint in BaNPPA improves the performance.

Visualization of the Allocation Matrix

For the Synthetic A data set, we further plot the NER scores (averaged over the five trials for $L = 14$) in the top plot in Figure 4.10. We see that, under LPPA, all latent functions have nonzero NER, while for BaNPPA only a small number of latent functions have high NER score.

In the bottom plot in Figure 4.10, we show the top four latent functions sorted according to the NER scores. For these plots, we used the best runs shown in Figure 4.5. We see that LPPA does not recover the true latent functions, while BaNPPA gives very similar results to the truth.

To visualize the responsibilities further, we plot the NER score and the normalized allocation matrix $\hat{\Theta}$ for the Mircoblog dataset in Figure 4.1. We show results for LPPA and BaNPPA. We choose runs that obtained the best test-likelihood in Figure 4.5, and visualize 100 time-sequences sampled randomly.

We see that as expected LPPA uses all latent functions to explain the data,

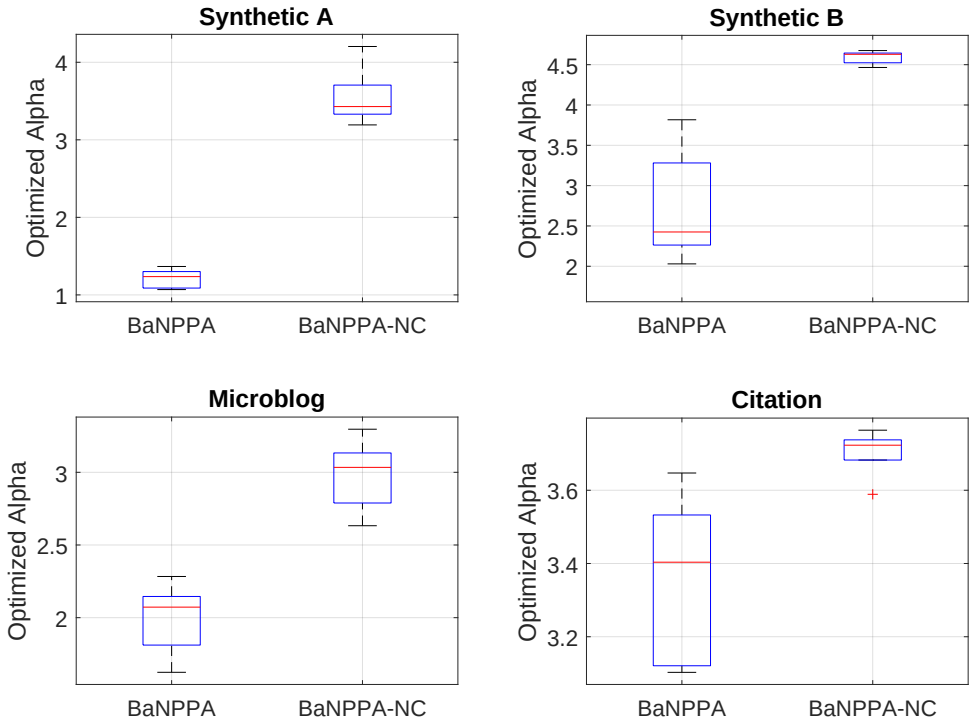

Figure 4.7: The comparison of the optimized α for four data sets (L=14) when optimizing the hyper-parameter α. BaNPPA achieves a smaller value of α comparing to BaNPPA-NC.

while BaNPPA assigns almost zero weights to latent functions with higher indices. This further confirms that, even when a large number of latent functions are given, BaNPPA automatically selects only a few to explain the data, while LPPA might overfit.

Finally, we further explore the impact of the volume constraint of Equation (4.9) in BaNPPA. We compare BaNPPA and BaNPPA-NC on the Synthetic B data set in Figure 4.11. We use results for $L = 14$ and $\alpha = 8$.

In the top plot, we see that BaNPPA and BaNPPA-NC both give similar UNER scores, yet as shown in the bottom plot, BaNPPA-NC does not recover the true latent functions. This result can be explained by looking at the expected volume $\mathbb{E}_q[\int_{\mathcal{X}} f_l^2(x)dx]$ shown in the middle plot. For BaNPPA, the volumes of all latent functions are equal, while, for BaNPPA-NC, the latent functions with higher UNER scores are assigned higher volume which eventually also get higher weights. This imbalance in the weights for some functions makes the results of BaNPPA-NC and BaNPPA different from each other. This result clearly shows that the volume constraint in BaNPPA plays an important role to recover the true latent functions which is important for interpretability.

Synthetic Data Sets with a Relatively Large L

Overall, BaNPPA-NC performs similarly to BaNPPA when the latent structure is simple but becomes less favorable when the structure gets complicated. To further demonstrate this, we add three more synthetic data set with a larger L. The details of the three data sets are given as follows:

1. **Synthetic C**. We sample 200 time-sequences from a mixture of 6 latent

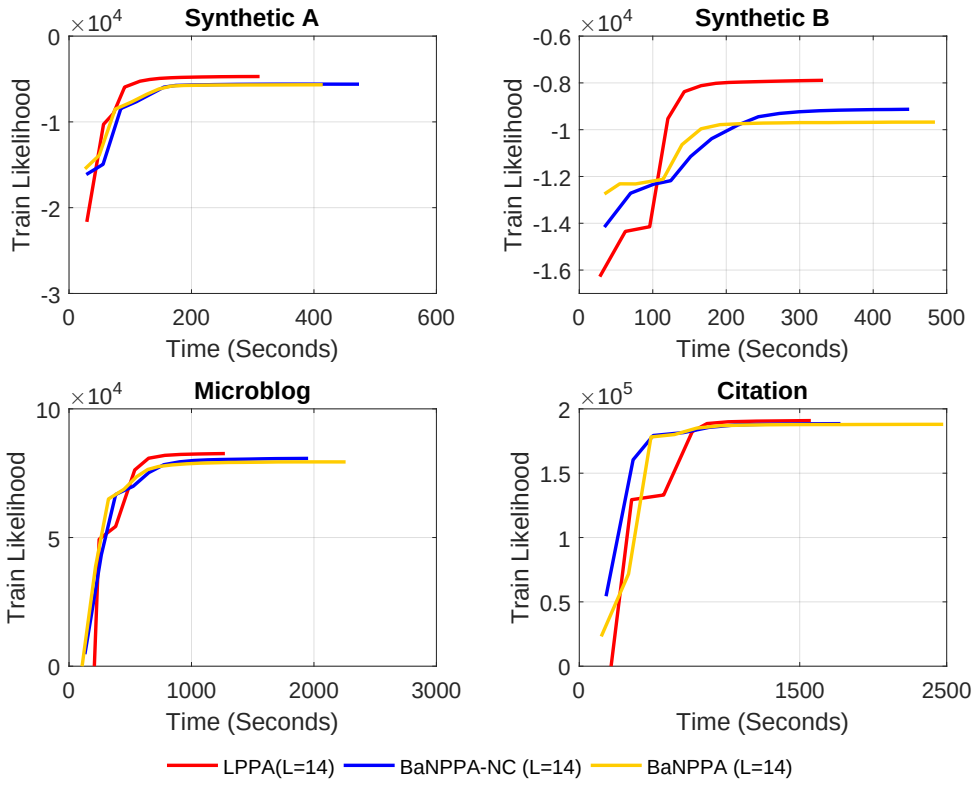

Figure 4.8: The comparison of the training likelihood versus time for four data sets (L=14) when optimizing the hyper-parameter α. The result of one trial is shown.

functions $\tilde{f}(x;\psi_l)$. The intensity function for the kth time-sequence is generated as follows:

$$\begin{aligned} s_k &\sim \text{Gamma}(2,3), \\ \boldsymbol{\theta}_k &\sim \text{Dir}(0.8,0.4,0.2,0.2,0.2,0.2), \\ \tilde{f}(x;\psi_l) &= \exp(-(x-15+10l)^2/10),\ l=1,\ldots,6, \\ \lambda_k(x) &= s_k \sum_{l=1}^{6} \theta_{kl} \tilde{f}(x;\psi_l),\ x \in [0,60]. \end{aligned}$$

2. **Synthetic D**. We sample 200 time-sequences from a mixture of 8 latent functions $\tilde{f}(x;\psi_l)$. The intensity function for the kth time-sequence is generated as follows:

$$\begin{aligned} s_k &\sim \text{Gamma}(2,3), \\ \boldsymbol{\theta}_d &\sim \text{Dir}(0.8,0.4,0.4,0.2,0.2,0.2,0.1,0.1), \\ \tilde{f}(x;\psi_l) &= \exp(-(x-15+10l)^2/10),\ l=1,\ldots,8, \\ \lambda_k(x) &= s_k \sum_{l=1}^{8} \theta_{kl} \tilde{f}(x;\psi_l),\ x \in [0,60]. \end{aligned}$$

3. **Synthetic E**.We sample 200 time-sequences from a mixture of 10 latent functions $\tilde{f}(x;\psi_l)$. The intensity function for the kth time-sequence is gen-

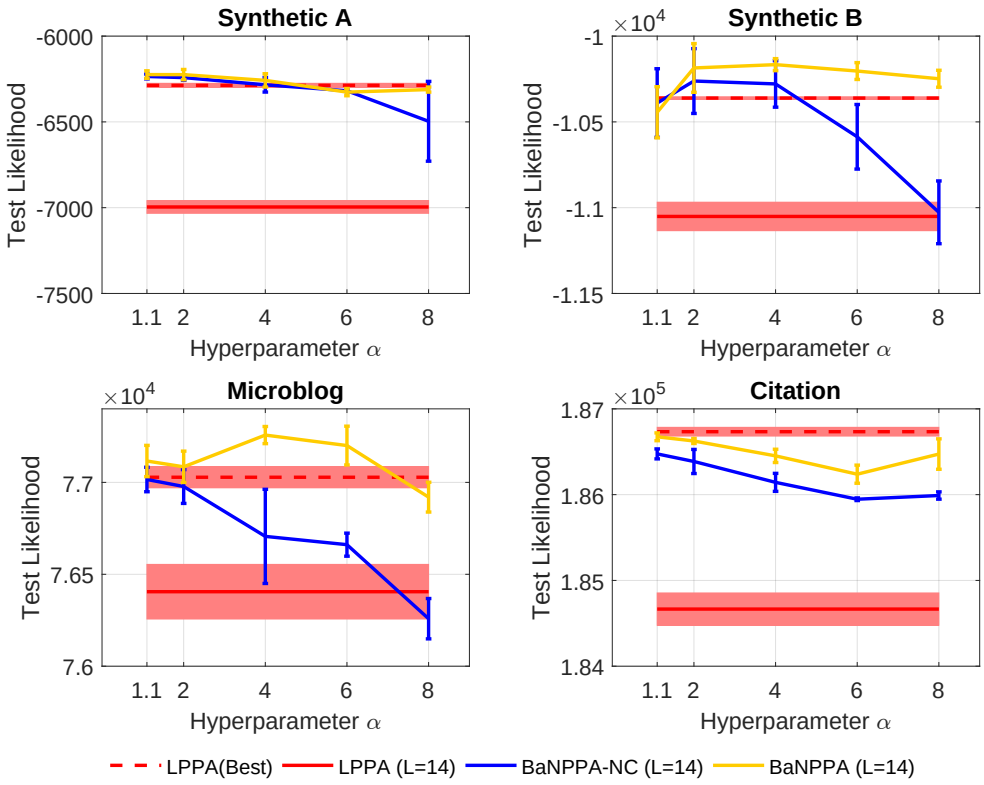

Figure 4.9: For a variety of hyperparameter values, BaNPPA gives the best performance which is also comparable to the best performance of LPPA and much better than LPPA with $L = 14$. Performance of BaNPPA-NC degrades with increasing α while performance of BaNPPA is relatively stable.

erated as follows:

$$
\begin{aligned}
s_k &\sim \text{Gamma}(2, 3), \\
\boldsymbol{\theta}_k &\sim \text{Dir}(0.8, 0.6, 0.4, 0.4, 0.4, 0.2, 0.2, 0.2, 0.1, 0.1) \\
\tilde{f}(x; \psi_l) &= \exp(-(x - 15 + 10l)^2/10),\ l = 1, \ldots, 10, \\
\lambda_k(x) &= s_k \sum_{l=1}^{10} \theta_{kl} \tilde{f}(x; \psi_l),\ x \in [0, 60].
\end{aligned}
$$

In the experiment, we fix the hyper-parameter a_0 and b_0. We also fix the length-scale hyper-parameters in all κ_l to be 4.3081 (Close to half of the span of $\tilde{f}(x; \psi_l)$). This means we only optimize the mixture weights and the variational distribution $q(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for Gaussian processes.

We vary the hyper-parameter $\alpha = [1.1, 2, 3, 4, 5]$. The result is given in Figure 4.12. We can see that BaNPPA-NC tends to over-shrink the components even when $\alpha = 5$ and gets a worse result.

4.6 Derivations Related to the ELBO

In this section, we give the details of the calculation of the ELBO as well as its derivatives with respect to all parameters in the framework. A derivation of the test likelihood used in the experiment will also be provided.

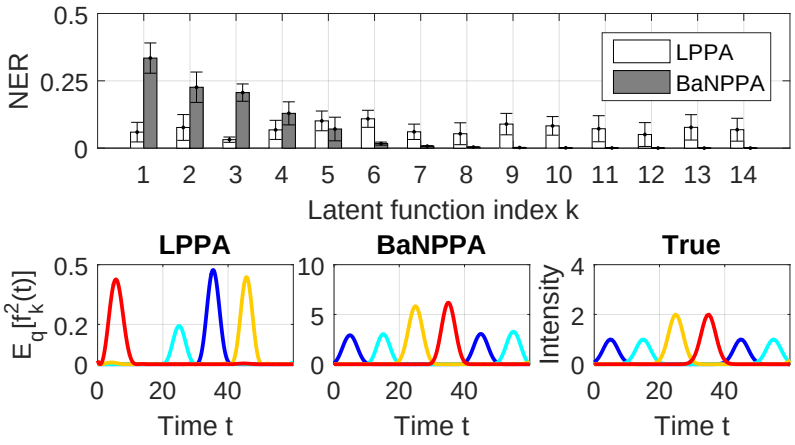

Figure 4.10: This figure shows that BaNPPA can reliably identify true latent functions for the Synthetic A data set. The top plot shows the NER scores for BaNPPA and LPPA for $L = 14$ where we see that, under LPPA, all latent functions have nonzero NER, while, under BaNPPA, only a handful of them have significant NER scores. The bottom plot shows the top four latent functions (sorted according to NER) obtained for both the methods along with the true latent functions. We see that BaNPPA recovers functions very similar to the true functions.

4.6.1 Derivation of the ELBO

Using Jensen's inequality, we bound the marginal log likelihood of the observed sequence $\ln p(\mathcal{D})$. Recall that the variational distribution is

$$q(\boldsymbol{s}, \boldsymbol{f}, \{\bar{\boldsymbol{f}}_l\}_{l=1}^{L}, \Theta) \triangleq \prod_{l=1}^{\infty} p(f_l|\bar{\boldsymbol{f}}_l) q(\bar{\boldsymbol{f}}_l) \prod_{k=1}^{K} \prod_{l=1}^{\infty} q(\theta'_{kl}) \prod_{k=1}^{K} q(s_k).$$

Hereafter we omit hyper-parameters $a_0, b_0, \alpha, \boldsymbol{H}$ in $\ln p(\mathcal{D}; a_0, b_0, \alpha, \boldsymbol{H})$ for simplicity.

$$\begin{aligned}
\ln p(\mathcal{D}) &= \ln \Big[\int \Big(\prod_{k=1}^{K} p(\boldsymbol{d}_k|\boldsymbol{\theta}_k, s_k, \boldsymbol{f}) p(s_k) p(\theta'_k) \Big) \prod_{l=1}^{\infty} p(f_l|\bar{\boldsymbol{f}}_l) p(\bar{\boldsymbol{f}}_l) d\boldsymbol{\theta}'_{\boldsymbol{k}} d\boldsymbol{f} \Big] \\
&\geq \sum_{k=1}^{K} \mathbb{E}_q \ln p(\boldsymbol{d}_k|\boldsymbol{\theta}_k, s_k, \boldsymbol{f}) \\
&+ \sum_{k=1}^{K} \sum_{l=1}^{L-1} \mathbb{E}_q \ln \frac{p(\theta'_{kl})}{q(\theta'_{kl})} + \sum_{k=1}^{K} \mathbb{E}_q \ln \frac{p(s_k)}{q(s_k)} + \sum_{l=1}^{L} \mathbb{E}_q \ln \frac{p(\bar{\boldsymbol{f}}_l)}{q(\bar{\boldsymbol{f}}_l)} \\
&\triangleq \mathcal{L}_0(q).
\end{aligned}$$

Using Lemma 2.6.3, we could further bound the first term to allow for a practical variational inference. This result is the same as the one obtained by

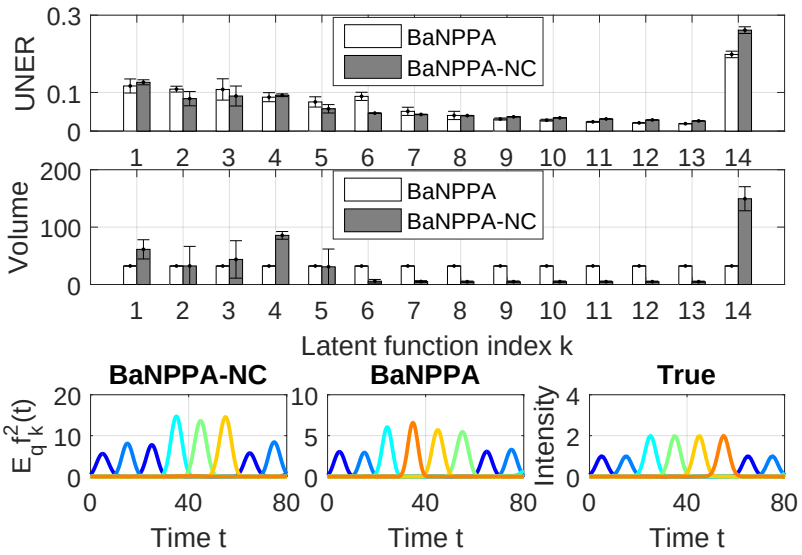

Figure 4.11: This figures shows that the volume constraint in BaNPPA is crucial to discover the true latent functions. Both BaNPPA and BaNPPA-NC obtain similar UNER score (top plot), yet the top latent functions obtained with the two methods are different (bottom plot). The imbalance in the volumes for BaNPPA-NC (middle plot) is the reason behind this difference. See the text for details.

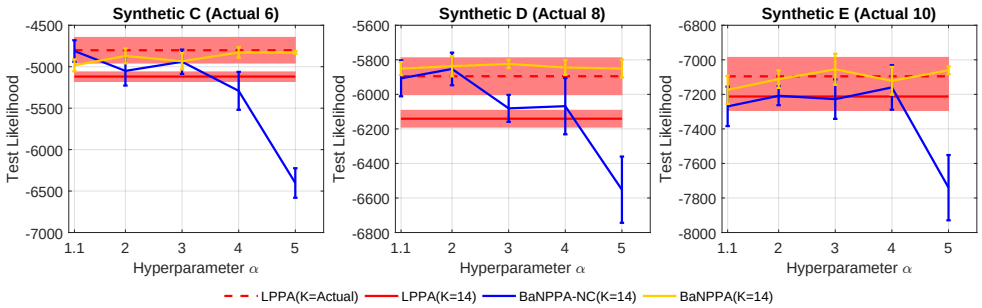

Figure 4.12: The comparison of the test likelihood for three additional data sets (L=14) when fixing the hyper-parameter $\alpha = [1.1, 2, 4, 6, 8]$. Error bars and shaded area represent the 95% confidence intervals.

following the methodology in LPPA [61].

$$
\begin{aligned}
&\mathbb{E}_q \ln p(\boldsymbol{d}_k|\boldsymbol{\theta}_k, s_k, \boldsymbol{f}) \\
&= \sum_{n=1}^{N_k} \Big(\ln \eta_k + \mathbb{E}_q \ln \sum_{l=1}^{L} \exp(\ln \theta_{kl} + \ln f_l^2(x_n^{(k)})) \Big) - \eta_k \int_{\mathcal{X}} \mathbb{E}_q \sum_{l=1}^{L} \theta_{kl} f_k^2(s) ds \\
&\geq \sum_{n=1}^{N_k} \Big(\ln \eta_k + \ln \sum_{l=1}^{L} \exp(\mathbb{E}_q \ln \theta_{kl} + \mathbb{E}_q \ln f_l^2(x_n^{(k)})) \Big) - \eta_k \int_{\mathcal{X}} \mathbb{E}_q \sum_{l=1}^{L} \theta_{kl} f_k^2(s) ds.
\end{aligned} \tag{4.14}
$$

Using Equation (4.14), we implicitly collapse the indicator variables and ob-

tain a lower bound of ELBO:

$$\begin{aligned}\mathcal{L}_1(q) \triangleq & \sum_{k=1}^{K}\sum_{n=1}^{N_k}\Big(\ln\eta_k + \ln\sum_{l=1}^{L}\exp(\mathbb{E}_q\ln\theta_{kl} + \mathbb{E}_q\ln f_l^2(x_n^{(k)}))\Big) \\ & - \sum_{k=1}^{K}\sum_{l=1}^{L}\eta_k\int_{\mathcal{X}}\mathbb{E}_q\theta_{kl}f_l^2(s)ds \\ & + \sum_{k=1}^{K}\sum_{l=1}^{L-1}\mathbb{E}_q\ln\frac{p(\theta'_{kl})}{q(\theta'_{kl})} + \sum_{k=1}^{K}\mathbb{E}_q\ln\frac{p(s_k)}{q(s_k)} + \sum_{l=1}^{L}\mathbb{E}_q\ln\frac{p(\bar{\boldsymbol{f}}_l)}{q(\bar{\boldsymbol{f}}_l)}. \end{aligned} \tag{4.15}$$

Now the posterior of the function f_l is a Gaussian process $\mathcal{GP}(u_l(x), B_l(x,x'))$, where

$$\begin{aligned} u_k(x) &= \boldsymbol{\kappa}_{l,x\bar{\boldsymbol{X}}}\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\mu}_k, \\ B_k(x,x') &= \kappa_{l,xx'} - \boldsymbol{\kappa}_{l,x\bar{\boldsymbol{X}}}\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\kappa}_{l,\bar{\boldsymbol{X}}x'} + \boldsymbol{\kappa}_{l,x\bar{\boldsymbol{X}}}\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\Sigma}_l\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\kappa}_{l,\bar{\boldsymbol{X}}x'}. \end{aligned}$$

And the expectation parts in Equation (4.15) can be computed as:

$$\begin{aligned} \mathbb{E}_q\ln p(\theta'_{kl}) &= \ln\alpha + (\alpha-1)\mathbb{E}_q[\ln(1-\theta'_{kl})], \\ \mathbb{E}_q\ln q(\theta'_{kl}) &= \ln\frac{\Gamma(\tau_{kl,0}+\tau_{kl,1})}{\Gamma(\tau_{kl,0})\Gamma(\tau_{kl,1})} \\ &\quad + (\tau_{kl,1}-1)\mathbb{E}_q[\ln(1-\theta'_{kl})] + (\tau_{kl,0}-1)\mathbb{E}_q[\ln\theta'_{kl}], \\ \mathbb{E}_q[\ln(1-\theta'_{kl})] &= \psi(\tau_{kl,1}) - \psi(\tau_{kl,0}+\tau_{kl,1}), \\ \mathbb{E}_q[\ln(\theta'_{kl})] &= \psi(\tau_{kl,0}) - \psi(\tau_{kl,0}+\tau_{kl,1}), \\ \mathbb{E}_q\ln p(s_k) &= a_0\ln b_0 - \ln\Gamma(a_0) + (a_0-1)\ln\eta_k - b_0\eta_k, \\ \mathbb{E}_q\ln\frac{p(\bar{\boldsymbol{f}}_l)}{q(\bar{\boldsymbol{f}}_l)} &= \frac{1}{2}\ln\frac{|\boldsymbol{\Sigma}_l|}{|\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}|} + \frac{m}{2} - \frac{1}{2}\mathrm{tr}\Big(\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}(\boldsymbol{\Sigma}_l + (\mu_k-g)(\mu_k-g)^\top)\Big), \\ \mathbb{E}_q[\ln f_l^2(x_n^{(k)})] &= g\Big(\frac{u_l(x_n^{(k)})^2}{2B_l(x_n^{(k)},x_n^{(k)})}\Big) - \gamma + \ln\Big(\frac{B_l(x_n^{(k)},x_n^{(k)})}{2}\Big), \\ \int_{\mathcal{X}}\mathbb{E}_q[f_l^2(s)]ds &= c|\mathcal{X}| - \mathrm{tr}(\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\Psi}_l) + \mathrm{tr}(\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\Psi}_l\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}(\boldsymbol{\Sigma}_l + \boldsymbol{\mu}_l\boldsymbol{\mu}_l^\top)). \end{aligned}$$

$g(x), x \le 0$ is calculated by a precomputed multi-resolution look-up table. γ is Euler's constant and $\boldsymbol{\Psi}_l \in \mathbb{R}^{M\times M}$. A detailed description can be found in Section 2.6.3 and 2.6.4.

After adding augmented Lagrangian penalty function, the modified evidence lower bound is:

$$\begin{aligned}\mathcal{L}_{\boldsymbol{v_i}}(\Phi, \boldsymbol{w_i}) \triangleq & \mathcal{L}_1(q) - \sum_{l=1}^{L}w_{il}\Big(\int_{\mathcal{X}}\mathbb{E}_q[f_l^2(s)]ds - A\Big) \\ & - \sum_{l=1}^{L}\frac{v_{il}}{2}\Big(\int_{\mathcal{X}}\mathbb{E}_q[f_l^2(s)]ds - A\Big)^2. \end{aligned} \tag{4.16}$$

4.6.2 The Variational Bayesian Expectation-Maximization Algorithm

Based on the modified evidence lower bound in Equation (4.16), we could derive the parameter learning method. In the E-step, we update the parameters $\{\boldsymbol{\tau}, \{\boldsymbol{\mu}\}_{l=1}^L, \{\boldsymbol{L}_l\}_{l=1}^L, \boldsymbol{s}\}$ and in the M-step, we update $\{\boldsymbol{H}, a_0, b_0, \alpha\}$.

Rate $q(s_k;\eta_k)$

We list the term related to η_k in Equation (4.16) first.

$$\mathcal{L}_{\eta_k} \triangleq N_k \ln \eta_k - \eta_k \int_{\mathcal{X}} \sum_{l=1}^{L} \mathbb{E}_q\Big(\theta_{kl} f_l^2(s)\Big) ds - \eta_k b_0 + (a_0 - 1)\ln \eta_k.$$

There is a closed form update for η_k

$$\eta_k = \frac{N_k + a_0 - 1}{b_0 + \int_{\mathcal{X}} \sum_{l=1}^{L} \mathbb{E}_q\Big(\theta_{kl} f_l^2(s)\Big) ds}.$$

Mixture Weights $q(\theta'_{kl};\tau_{kl,0},\tau_{kl,1})$

We list the term related to these parameters in Equation (4.16) first.

$$\begin{aligned}\mathcal{L}_{\tau_{kl}} \triangleq & \sum_{n=1}^{N_k} \Big[\ln \sum_{l=1}^{L} \exp\Big(\mathbb{E}_q[\ln \theta_{kl}] + \mathbb{E}_q[\ln f_l^2(x_n^{(k)})]\Big)\Big] - \eta_k \int_{\mathcal{X}} \mathbb{E}_q \sum_{l=1}^{L} \theta_{kl} f_l^2(s) ds \\ & + \Big(\ln \frac{\Gamma(\tau_{kl,0})\Gamma(\tau_{kl,1})}{\Gamma(\tau_{kl,0}+\tau_{kl,1})} - (\tau_{kl,0}-1)\mathbb{E}_q \ln \theta'_{kl} + (\alpha - \tau_{kl,1})\mathbb{E}_q \ln(1-\theta'_{kl})\Big).\end{aligned}$$

Let

$$\begin{aligned}L_{knl} &\triangleq \exp\Big(\mathbb{E}_q[\ln \theta_{kl}] + \mathbb{E}_q[\ln f_l^2(x_n^{(k)})]\Big) \\ &= \exp\Big(\psi(\tau_{kl,0}) + \sum_{j=1}^{l-1} \psi(\tau_{kj,1}) - \sum_{j=1}^{l} \psi(\tau_{kj,0}+\tau_{kj,1}) + \mathbb{E}_q[\ln f_l^2(x_n^{(k)})]\Big), \\ V_l &\triangleq \int_{\mathcal{X}} \mathbb{E}_q f_l^2(s) ds.\end{aligned}$$

There is no closed form update for these variables, we use coordinate ascent method.

$$\begin{aligned}\frac{\partial \mathcal{L}_{\tau_{kl}}}{\partial \tau_{kl,0}} =& -\eta_k\Big(V_k \frac{\partial \mathbb{E}[\theta_{kl}]}{\partial \tau_{kl,0}} + \sum_{j=l+1}^{L} V_l \frac{\partial \mathbb{E}[\theta_{kj}]}{\partial \tau_{kl,0}}\Big) \\ & - \Big(\tau_{kl,0} - 1 - \sum_{n=1}^{N_k} \frac{L_{knl}}{\sum_{v=1}^{L} L_{knv}}\Big)\psi'(\tau_{kl,0}) \\ & + \Big(\tau_{kl,0} - 1 + \tau_{kl,1} - \alpha - \sum_{n=1}^{N_k} \frac{\sum_{v=l}^{L} L_{knv}}{\sum_{v=1}^{L} L_{knv}}\Big)\psi'(\tau_{kl,0}+\tau_{kl,1}), \\ \frac{\partial \mathcal{L}_{\tau_{kl}}}{\partial \tau_{kl,1}} =& -\eta_k\Big(V_k \frac{\partial \mathbb{E}[\theta_{kl}]}{\partial \tau_{kl,1}} + \sum_{j=l+1}^{L} V_l \frac{\partial \mathbb{E}[\theta_{kj}]}{\partial \tau_{kl,1}}\Big) \\ & - \Big(\tau_{kl,1} - \alpha - \sum_{n=1}^{N_k} \frac{\sum_{v=l+1}^{K} L_{knv}}{\sum_{v=1}^{L} L_{knv}}\Big)\psi'(\tau_{kl,1}) \\ & + \Big(\tau_{kl,0} - 1 + \tau_{kl,1} - \alpha - \sum_{n=1}^{N_k} \frac{\sum_{v=l}^{L} L_{knv}}{\sum_{v=1}^{L} L_{knv}}\Big)\psi'(\tau_{kl,0}+\tau_{kl,1}).\end{aligned}$$

The derivatives can be computed by

$$\frac{\partial \mathbb{E}[\theta_{kl}]}{\partial \tau_{kl,0}} = \frac{\tau_{kl,1}}{(\tau_{kl,0}+\tau_{kl,1})^2}\prod_{j=1}^{l-1}\frac{\tau_{kj,1}}{\tau_{kj,0}+\tau_{kj,1}},$$
$$\frac{\partial \mathbb{E}[\theta_{kl}]}{\partial \tau_{kl,1}} = -\frac{\tau_{kl,0}}{(\tau_{kl,0}+\tau_{kl,1})^2}\prod_{j=1}^{l-1}\frac{\tau_{kj,1}}{\tau_{kj,0}+\tau_{kj,1}},$$
$$\frac{\partial \mathbb{E}[\theta_{kj}]}{\partial \tau_{kl,0}} = -\frac{\tau_{kj,0}}{\tau_{kj,0}+\tau_{kj,1}}\frac{\tau_{kl,1}}{(\tau_{kl,0}+\tau_{kl,1})^2}\prod_{v=1,v\neq l}^{j-1}\frac{\tau_{kv,1}}{\tau_{kv,0}+\tau_{kv,1}},\ j>l$$
$$\frac{\partial \mathbb{E}[\theta_{kj}]}{\partial \tau_{kl,1}} = \frac{\tau_{kj,0}}{\tau_{kj,0}+\tau_{kj,1}}\frac{\tau_{kl,0}}{(\tau_{kl,0}+\tau_{kl,1})^2}\prod_{v=1,v\neq l}^{j-1}\frac{\tau_{kv,1}}{\tau_{kv,0}+\tau_{kv,1}},\ j>l.$$

Pseudo Inputs $q(\bar{\boldsymbol{f}}_l;\boldsymbol{\Sigma}_l,\boldsymbol{\mu}_l)$

Let $\varphi_l=\{\boldsymbol{\Sigma}_l,\boldsymbol{\mu}_l\}$. We list the term related to these parameters in Equation (4.16) first.

$$\mathcal{L}_{\varphi_l} = \sum_{k=1}^{K}\sum_{n=1}^{N_k}\ln\Big(\sum_{l=1}^{L}L_{knl}\Big) - \sum_{k=1}^{K}\sum_{l=1}^{L}\eta_k\mathbb{E}_q[\theta_{kl}]V_l - w_{il}(V_l-A) - \frac{v_{il}}{2}(V_l-A)^2 + \Big[\frac{1}{2}\ln|\boldsymbol{\Sigma}_l| - \frac{1}{2}\mathrm{tr}\Big(\boldsymbol{K}^{-1}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}(\boldsymbol{\Sigma}_l+(\boldsymbol{\mu}_l-g)(\boldsymbol{\mu}_l-g)^\top)\Big)\Big].$$

Taking derivatives with respect to $\boldsymbol{\Sigma}_l,\boldsymbol{\mu}_l$, we obtain

$$\frac{\partial \mathcal{L}_{\varphi_l}}{\partial \boldsymbol{\mu}_l} = \sum_{k=1}^{K}\Big(\sum_{n=1}^{N_k}\frac{1}{\sum_{v=1}^{L}L_{knv}}\frac{\partial L_{knl}}{\partial \boldsymbol{\mu}_l}\Big) - \Big(w_{il}+v_{il}(V_k-A)+\sum_{k=1}^{K}\eta_k\mathbb{E}_q[\theta_{kl}]\Big)\frac{\partial V_l}{\partial \boldsymbol{\mu}_l} - \boldsymbol{K}^{-1}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}(\boldsymbol{\mu}_l-g),$$
$$\frac{\partial \mathcal{L}_{\varphi_l}}{\partial \boldsymbol{\Sigma}_l} = \sum_{k=1}^{K}\Big(\sum_{n=1}^{N_k}\frac{1}{\sum_{v=1}^{L}L_{knv}}\frac{\partial L_{knl}}{\partial \boldsymbol{\Sigma}_l}\Big) - \Big(w_{il}+v_{il}(V_k-A)+\sum_{k=1}^{K}\eta_k\mathbb{E}_q[\theta_{kl}]\Big)\frac{\partial V_l}{\partial \boldsymbol{\Sigma}_l} + \frac{1}{2}\boldsymbol{\Sigma}_l^{-1} - \frac{1}{2}\boldsymbol{K}^{-1}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}},$$

Let $u_{l,kn}\triangleq u_l(x_n^{(k)})$, $B_{l,kn}\triangleq B_l(x_n^{(k)},x_n^{(k)})$ and $\boldsymbol{e}_{ln}\triangleq \boldsymbol{K}^{-1}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}\boldsymbol{\kappa}_{l,\bar{\boldsymbol{X}}x_n^{(k)}}\in\mathbb{R}^{M\times 1}$. The four gradients can be computed by

$$\frac{\partial L_{knl}}{\partial \boldsymbol{\mu}_l} = \Big(g'\Big(\frac{u_{l,kn}^2}{2B_{l,kn}}\Big)\frac{u_{l,kn}}{B_{l,kn}}\Big)\mathbf{e}_{ln},\ \frac{\partial V_l}{\partial \boldsymbol{\mu}_l} = 2\boldsymbol{K}^{-1}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}\boldsymbol{\Psi}_l\boldsymbol{K}^{-1}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}\boldsymbol{\mu}_l,$$
$$\frac{\partial L_{knl}}{\partial \boldsymbol{\Sigma}_l} = \Big(-g'\Big(\frac{u_{l,kn}^2}{2B_{l,kn}}\Big)\frac{u_{l,kn}^2}{2B_{l,kn}^2}+\frac{1}{B_{l,kn}}\Big)\mathbf{e}_{ln}\mathbf{e}_{ln}^\top,\ \frac{\partial V_l}{\partial \boldsymbol{\Sigma}_l} = \boldsymbol{K}^{-1}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}\boldsymbol{\Psi}_l\boldsymbol{K}^{-1}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}.$$

Finally, since we are using the parametrization $\boldsymbol{\Sigma}_l=\boldsymbol{L}_l\boldsymbol{L}_l^\top$, we have

$$\frac{\partial V_l}{\partial \boldsymbol{L}_l} = 2\frac{\partial V_l}{\partial \boldsymbol{\Sigma}_l}\boldsymbol{L}_l,\ \frac{\partial L_{knl}}{\partial \boldsymbol{L}_l} = 2\frac{\partial L_{knl}}{\partial \boldsymbol{\Sigma}_l}\boldsymbol{L}_l.$$

Hyper-parameter in the GP

Let ς_l denote the hyper-parameters in the prior for the lth latent function. We list the term related to these parameters in Equation (4.16) first.

$$\mathcal{L}_{\varsigma_l} = \sum_{k=1}^{K}\sum_{n=1}^{N_k} \ln\Big(\sum_{l=1}^{L} L_{knl}\Big) - \sum_{k=1}^{K}\sum_{l=1}^{L} \eta_k \mathbb{E}_q[\theta_{kl}] V_l - w_{il}(V_l - A) - \frac{v_{il}}{2}(V_l - A)^2 + \Big[-\frac{1}{2}\ln|\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}| - \frac{1}{2}\mathrm{tr}\Big(\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}(\boldsymbol{\Sigma}_l + (\boldsymbol{\mu}_l - g)(\boldsymbol{\mu}_l - g)^\top)\Big)\Big].$$

Taking derivatives with respect to ς_l, we obtain

$$\begin{aligned}\frac{\partial \mathcal{L}_{\varsigma_l}}{\partial \varsigma_l} =& \sum_{k=1}^{K}\Big(\sum_{n=1}^{N_k}\frac{1}{\sum_{v=1}^{L} L_{knv}}\frac{\partial L_{knl}}{\partial \varsigma_l}\Big) - \Big(w_{il} + v_{il}(V_k - A) + \sum_{k=1}^{K}\eta_k \mathbb{E}_q[\theta_{kl}]\Big)\frac{\partial V_l}{\partial \varsigma_l} \\ &- \frac{1}{2}\mathrm{tr}\Big(\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\frac{\partial \boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}}{\partial \varsigma_l}\Big) \\ &+ \frac{1}{2}\mathrm{tr}\Big(\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\frac{\partial \boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}}{\partial \varsigma_l}\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}(\boldsymbol{\Sigma}_l + (\boldsymbol{\mu}_l - g)(\boldsymbol{\mu}_l - g)^\top)\Big).\end{aligned}$$

The two gradients can be computed by

$$\begin{aligned}\frac{\partial L_{knl}}{\partial \varsigma_l} =& \Big(g'\Big(\frac{u_{l,kn}^2}{2B_{l,kn}}\Big)\frac{u_{l,kn}}{B_{l,kn}}\Big)\boldsymbol{\mu}_l^\top \frac{\partial \mathbf{e}_{ln}}{\partial \varsigma_l} \\ &+ 2\Big(-g'\Big(\frac{u_{l,kn}^2}{2B_{l,kn}}\Big)\frac{u_{l,kn}^2}{2B_{l,kn}^2} + \frac{1}{B_{l,kn}}\Big)\mathrm{tr}\Big(\boldsymbol{\Sigma}_l \frac{\partial \mathbf{e}_{ln}}{\partial \varsigma_l}\mathbf{e}_{ln}^\top\Big), \\ \frac{\partial V_l}{\partial \varsigma_l} =& \mathrm{tr}\Big(\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\Psi}_l\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\frac{\partial \boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}}{\partial \varsigma_l} - \boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\frac{\partial \boldsymbol{\Psi}_l}{\partial \varsigma_l}\Big) \\ &+ \mathrm{tr}\Big(\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}(\boldsymbol{\mu}_l\boldsymbol{\mu}_l^\top + \boldsymbol{\Sigma}_l)\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\Big(\frac{\partial \boldsymbol{\Psi}_l}{\partial \varsigma_l} - 2\frac{\partial \boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}}{\partial \varsigma_l}\boldsymbol{K}_{l,\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}}^{-1}\boldsymbol{\Psi}_l\Big)\Big).\end{aligned}$$

Beta distribution prior α

We list the term related to α in Equation (4.16) first.

$$\mathcal{L}_\alpha \triangleq K(L-1)\ln\alpha + (\alpha - 1)\sum_{k=1}^{K}\sum_{l=1}^{L-1}(\psi(\tau_{kl,1}) - \psi(\tau_{kl,0} + \tau_{kl,1})).$$

Then we have a closed form update for α.

$$\alpha = \frac{K(L-1)}{\sum_{k=1}^{K}\sum_{l=1}^{L-1}\Big(\psi(\tau_{kl,1} + \tau_{kl,0}) - \psi(\tau_{kl,1})\Big)}. \tag{4.17}$$

Gamma distribution prior a_0, b_0

We list the term related to a_0, b_0 in Equation (4.16) first.

$$L_{a_0,b_0} = -\sum_{k=1}^{K}\Big(a_0\ln b_0 - \ln\Gamma(a_0) - \eta_k b_0 + (a_0 - 1)\ln\eta_k\Big).$$

Then we have

$$\frac{\partial L_{a_0,b_0}}{\partial a_0} = -K\ln b_0 + K\psi(a_0) - \sum_{k=1}^{K}\ln\eta_k,\ b_0 = \frac{Ka_0}{\sum_{k=1}^{K}\eta_k}.$$

4.6.3 Derivation of the Lower Bound of the Test Likelihood

In LPPA, the allocation matrix Θ is treated as hyper-parameters and all the parameters are $\{\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{H}, \Theta\}$. Let $\Phi = \{\boldsymbol{H}, \Theta\}$. In variational inference we use the variational distribution $q(\boldsymbol{f}; \Phi)$ to approximate the posterior $p(\boldsymbol{f}|\mathcal{D}_{\text{train}}; \Phi)$. The test likelihood can be lower-bounded as follows.

$$
\begin{aligned}
&\ln p(\mathcal{D}_{\text{test}}|\mathcal{D}_{\text{train}}; \Phi) = \ln \int p(\mathcal{D}_{\text{test}}|\boldsymbol{f}; \Phi) p(\boldsymbol{f}|\mathcal{D}_{\text{train}}; \Phi) d\boldsymbol{f} \\
\approx &\ln \int p(\mathcal{D}_{\text{test}}|\boldsymbol{f}; \Phi) q(\boldsymbol{f}; \Phi) d\boldsymbol{f} \\
\geq &\int q(\boldsymbol{f}; \Phi) \ln \frac{p(\mathcal{D}_{\text{test}}|\boldsymbol{f}; \Phi) q(\boldsymbol{f}; \Phi)}{q(\boldsymbol{f}; \Phi)} d\boldsymbol{f} = \mathbb{E}_q \ln p(\mathcal{D}_{\text{test}}|\boldsymbol{f}; \Phi) \\
\geq &\sum_{k=1}^{K} \sum_{n=1}^{N_k^{\text{test}}} \ln \sum_{l=1}^{L} \theta_{kl} \exp\left[\mathbb{E}_q(\ln f_l^2(x_n^{(k)}))\right] - \sum_{k=1}^{K} \sum_{l=1}^{L} \theta_{kl} \int_{\mathcal{X}} \mathbb{E}_q[f_l^2(s)] ds \triangleq \mathcal{L}_{\text{test}}.
\end{aligned}
\tag{4.18}
$$

In BaNPPA, all the parameters to be optimized are $\{\boldsymbol{\eta}, \boldsymbol{\tau}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{H}, a_0, b_0, \alpha\}$. Let $\Phi = \{\boldsymbol{H}, a_0, b_0, \alpha\}$. However, if we follow the same deduction as LPPA, we will not arrive at a fair comparison since the inequality in Equation (4.18) is different in principle for LPPA and BaNPPA, and therefore, we draw V samples from variational distribution $q(\boldsymbol{s}, \boldsymbol{\Theta}; a_0, b_0, \alpha)$ for $\boldsymbol{\Theta}$ and then follow the lower bound in Equation (4.18).

$$
\begin{aligned}
&\mathbb{E}_q \ln p(\mathcal{D}_{\text{test}}|\boldsymbol{s}, \boldsymbol{\Theta}, \boldsymbol{f}; \Phi) \\
= &\int q(\boldsymbol{s}, \boldsymbol{\Theta}, \boldsymbol{f}; \Phi) \ln p(\mathcal{D}_{\text{test}}|\boldsymbol{s}, \boldsymbol{\Theta}, \boldsymbol{f}; \Phi) d\boldsymbol{s} d\boldsymbol{\Theta} d\boldsymbol{f} \\
\approx &\frac{1}{V} \sum_{v=1}^{V} \int q(\boldsymbol{f}; H) \ln p(\mathcal{D}_{\text{test}}|\boldsymbol{s}, \boldsymbol{\Theta}_v, \boldsymbol{f}; H) d\boldsymbol{f} \\
\geq &\frac{1}{V} \sum_{v=1}^{V} \sum_{k=1}^{K} \Big(\sum_{n=1}^{N_k^{\text{test}}} \ln \Big(s_k \sum_{l=1}^{L} \theta_{v,kl} e^{\mathbb{E}_q(\ln f_l^2(x_n^{(k)}))} \Big) - s_k \sum_{l=1}^{L} \theta_{v,kl} \int_{\mathcal{X}} \mathbb{E}_q[f_l^2(s)] ds \Big).
\end{aligned}
\tag{4.19}
$$

Chapter 5

Conclusion and Future Work

In this chapter, we conclude the thesis and present several possible directions for future research.

5.1 Discussion and Conclusion

Time-sequence data can generally be divided into two categories: recurrent event data and panel count data [91]. The thesis was devoted to addressing several technical problems in the variational inference when we have panel count data or recurrent event data.

In Chapter 3, we presented the first framework for GP-modulated Poisson processes when data appear in the form of panel counts. To simplify the problem, we make the assumption that all time-sequences share the same intensity function. Thanks to this assumption, we can obtain an estimate of the average intensity function. We derived a tractable lower bound for the intractable evidence lower bound when modeling the panel count data using the GP-modulated intensity function. Our model, the Gaussian-process-modulated Poisson process for panel count data (GP4C), outperforms a non-Bayesian method using the maximum likelihood criterion in terms of the test likelihood and achieves comparable results in terms of computation time. Generally speaking, GP4C serves as an alternative to the current mainstream point-estimates for the machine learning researchers and practitioners who are interested in modeling and understanding panel count data.

In Chapter 3, we made an assumption that in the data set, all time-sequences share the same underlying intensity function. However, this assumption prevents us from inferring the diversity among multiple time-sequences. As the starting point to study the diversity among multiple time-sequences, we incorporated an additional variable for each time-sequence to model the diversity. In the scenario of the clinical trial, this variable can be interpreted as the level of severity in each patient. We name this model the GP4C model with individual weight (GP4CW). We showed through experiments that GP4CW outperforms the GP4C model in terms of the test likelihood. For medical practitioners, GP4C can estimate the average rate of events while GP4CW can provide an estimate of the severity of each patient.

In Chapter 4, we further generalize the assumptions in GP4C and GP4CW. Instead of using only one latent function, we assume that there exists a set of latent functions and the intensity function of each time-sequence can be obtained by linearly combining all the latent functions. We proposed Bayesian nonparametric Poisson process allocation (BaNPPA), to automatically infer the number of latent functions. We combined Bayesian nonparametric methods with the exist-

ing latent Poisson process allocation (LPPA) method and showed that this naive combination might result in over-shrinkage of the latent functions. We solved this problem by imposing a volume constraint within the variational Bayesian inference framework. We demonstrated that the proposed model outperforms the LPPA model and the integral constraints we imposed on the objective function help the inference of the underlying latent functions. For medical practitioners, BaNPPA can automatically identify different patterns of symptoms and help develop individual treatments for each patient.

5.2 Future Work

In this section, we present several possible directions for future research.

5.2.1 Two-Sample Test

In Chapter 3, we estimate the mean intensity function for both the treatment and the placebo group. A natural question is that whether the mean intensity functions of the two groups are significantly different.

Let the mean intensity functions for the treatment group and the placebo group be $\lambda_1(x)$ and $\lambda_0(x)$ respectively. Cook and Lawless [15] and [58] made the following proportional mean function assumption:

$$\lambda_1(x) = \exp(\beta)\lambda_0(x).$$

A score test for the real number β was then conducted by utilizing a point-estimate of the mean intensity function $\lambda_0(x)$ [15, 58]. The null hypothesis $\mathcal{H}_0$ and the alternative hypothesis $\mathcal{H}_1$ are given as follows.

$$\begin{aligned}\mathcal{H}_0 :&\ \beta = 0,\\ \mathcal{H}_1 :&\ \beta \neq 0.\end{aligned}$$

When we can not reject the null hypothesis $\beta = 0$, the mean intensity functions for both groups are the same, that is, $\lambda_0(x) = \lambda_1(x)$. This implies that the treatment is not effective.

With the inference methods in Lloyd et al. [60] and Chapter 3, we can obtain a Bayesian estimate of the mean intensity function. However, we can not directly use the point-estimate tools in Cook and Lawless [15]. It would be interesting to investigate the benefits we could obtain from the Bayesian estimate in the task of hypothesis test. For example, we can assume that the mean intensity functions for both groups are generated by the following model.

$$\begin{aligned}\lambda_1(x) &= (f(x) + g(x))^2,\\ \lambda_0(x) &= f^2(x),\\ f(x) &\sim \mathcal{GP}(m(x), \kappa_f(x, x')),\\ g(x) &\sim \mathcal{GP}(m(x), \kappa_g(x, x')).\end{aligned}$$

Note that since a linear combination of two Gaussian processes is a Gaussian process [77], $f(x) + g(x)$ still follows a Gaussian process. After we obtain the Bayesian estimate of the function $g(x)$, a test can be performed by examining whether $g(x) = 0$ using the hypothesis test method in Benavoli and Mangili [6].

5.2.2 Analysis of the Error in Corollary 3.3.1

In Chapter 3, we did not give an upper bound of the error in Corollary 3.3.1. The actual error in Corollary 3.3.1 is

$$\begin{aligned} g_{\text{error}} =& \mathbb{E}_q\Big[\ln \int_{\mathcal{X}_i^{(k)}} f^2(x)dx\Big] - \ln\Big(\int_{\mathcal{X}_i^{(k)}} e^{\ln(\mathbb{E}_q^2 f(x)+b\text{Var}_q f(x))+\xi}dx\Big) \\ =& \Big(\mathbb{E}_q\Big[\ln \int_{\mathcal{X}_i^{(k)}} f^2(x)dx\Big] - \ln\Big(\int_{\mathcal{X}_i^{(k)}} e^{\mathbb{E}_q \ln f^2(x)}dx\Big)\Big) \\ &+ \Big(\ln\Big(\int_{\mathcal{X}_i^{(k)}} e^{\mathbb{E}_q \ln f^2(x)}dx\Big) - \ln\Big(\int_{\mathcal{X}_i^{(k)}} e^{\ln(\mathbb{E}_q^2 f(x)+b\text{Var}_q f(x))+\xi}dx\Big)\Big) \\ =& \underbrace{\mathbb{E}_q\Big[\ln \int_{\mathcal{X}_i^{(k)}} f^2(x)dx\Big] - \ln\Big(\int_{\mathcal{X}_i^{(k)}} e^{\mathbb{E}_q \ln f^2(x)}dx\Big)}_{g_0} + h(\varphi(x_c); b), \end{aligned}$$

where $h(\varphi; b)$ is defined in Section 3.3.4 and $x_c \in \mathcal{X}_i^{(k)}$. We used a heuristic method to estimate the error of $h(\varphi(x_c); b)$ in Section 3.3.4.

For the first part g_0 of the error g_{error}, to the best of our knowledge, the analysis of the error has not been studied before. The derivation of the inequality can be found in Paisley [74] and this inequality has also been used implicitly in the previous study [61]. We illustrate the bias after applying this inequality by the following simple toy experiment.

$$Y_1 = X_1^2,\ Y_2 = X_2^2,\ X_1 \sim \mathcal{N}(2,1),\ X_2 \sim \mathcal{N}(2,4),$$

where $\mathcal{N}(\cdot)$ is the normal distribution. Using Lemma 2.6.3, we can arrive at the following inequality:

$$\begin{aligned} \mathcal{L}_{\text{left}} &\triangleq \mathbb{E}_{p(Y_{1:2})} \ln\Big(wY_1 + (1-w)Y_2\Big) \\ &\geq \ln\Big(w\exp(\mathbb{E}\ln Y_1) + (1-w)\exp(\mathbb{E}\ln Y_2)\Big) \\ &\triangleq \mathcal{L}_{\text{right}},\ w \in [0,1]. \end{aligned} \tag{5.1}$$

The experiment is to maximize both sides with respect to w in Inequality 5.1. We vary the value of w and calculate $\mathcal{L}_{\text{right}}$ and $\mathcal{L}_{\text{left}}$. The result is given in Figure 5.1. We see that the for the right-most $\mathcal{L}_{\text{right}}$ the optimal value of w is $w = 1$ while for $\mathcal{L}_{\text{left}}$ the optimal value is between 0 and 1. With this toy experiment, we confirm that after applying the inequality an additional bias is added to the maximization result.

An intuitive explanation for the bias is that the logarithm function will punish values which are closer to zero harder. Since Y_2 has a large variance, there will be a large proportion of samples near zero. This makes the corresponding $\mathbb{E}\ln Y_2$ smaller and less favorable.

In this direction, an analysis of the error g_0 would help understand the accuracy of our Bayesian estimate.

5.2.3 Poisson Process Allocation for Panel Count Data

The combination of the panel count data model in Chapter 3 with the Poisson process allocation model in Chapter 4 would be an interesting topic. Taking the clinical experiment as an example, in Figure 3.1 we notice that the diversity among patients can not be easily neglected and a careful study of the reactions from different patients would help doctors make a specific plan for each patient.

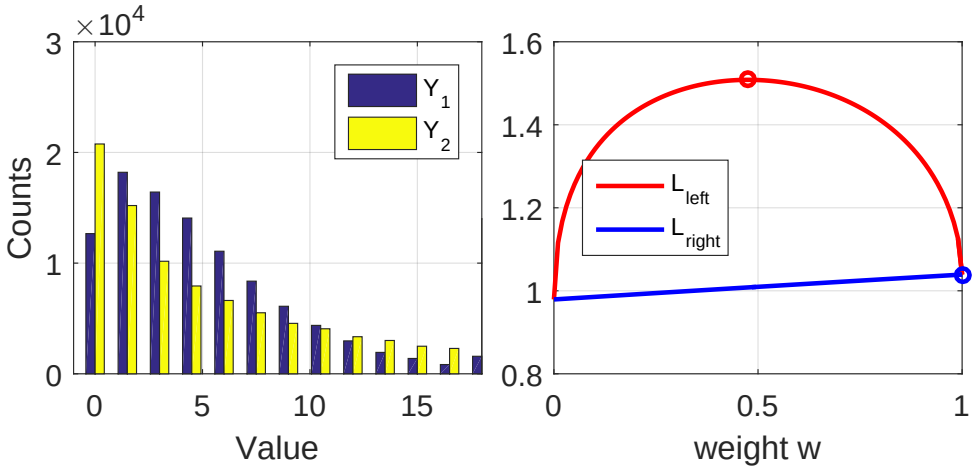

Figure 5.1: Bias in the inference with lower bound. Left: The histogram of Y_1 and Y_2 when sampling both variables 10^5 times. Right: $\mathcal{L}_{\text{left}}$ (Blue) versus $\mathcal{L}_{\text{right}}$ (Red) and the round marker indicates the maximum of the curve.

In this direction, we can combine the generative process of LPPA [61] with the panel count data model. We name this model the Poisson process allocation for panel count data (PPA-PCD). The generative process for the PPA-PCD model is given in Algorithm 13. In PPA-PCD, we are provided with additional censoring intervals $\{\mathcal{X}_i^{(k)}\}, k = 1, \ldots, K$. Another difference between PPA-PCD and LPPA is that we need to add an additional censoring step after we sample the recurrent event data for each subject since the exact time-stamps in the panel count data are not revealed to the observer.

A challenge in this direction is that a large data set of read-world panel count data from patients may not be easily accessible since it is related to the privacy of the patients.

5.2.4 Pattern Mining From Multiple Time-Sequences

In Chapter 4, we present the BaNPPA model to automatically infer the number of latent functions for multiple time-sequences. Although BaNPPA can model the multiple time-sequences well in terms of the test likelihood, we observe that in the experiment that it is difficult to gain insights from the latent functions of the inference results from BaNPPA.

We plot the inference results of the latent functions in the Microblog data set in Figure 5.2. A description of the Microblog data set can be found in Section 4.5. In Figure 5.2, we notice that each latent function contains only one peak. However, a more meaningful pattern may contain multiple peaks [45] since there might be some triggering mechanisms in multiple peaks. Thus we may find it more interesting to investigate whether there is a group of time-sequences sharing the same multiple-peak pattern. A careful examination of the information from this group may help us understand the mechanism in the generative process better.

Algorithm 13: The generative process for the PPA-PCD model.

Input : The number of latent function L, the number of the time-sequences K, the mixture weights $\{\theta_{kl}\}$, the mean value m_0, the covariance functions in L Gaussian processes $\{\kappa_l\}$ and the observation intervals $\{\mathcal{X}_i^{(k)}\}, k = 1, \ldots, K$.

Output: The time-sequence data $\mathcal{D} = \{\boldsymbol{d}_k\}_{k=1}^K$.

for *each basis function* $l = 1, \ldots, L$ **do**
  Sample $f_l \sim \mathcal{GP}(m_0(x), \kappa_l(x, x'))$.
end
for *each subject* $k = 1, \ldots, K$ **do**
  Calculate the intensity function.

$$\lambda_k(x) = \sum_{l=1}^{L} \theta_{kl} f_l^2(x), \quad \theta_{kl} \geq 0.$$

  Sample $\boldsymbol{d}_k \sim \text{IPP}(\lambda_k(x))$ on the time window $\mathcal{X}^{(k)} = \cup_i \mathcal{X}_i^{(k)}$.
  for *each observation interval* $i = 1, \ldots, N_k$ **do**
    Censoring the recurrent events $\boldsymbol{d}_k$ with $\mathcal{X}_i^{(k)}$.

$$m_i^{(k)} = \#\{x | x \in \mathcal{X}_i^{(k)}, x \in \boldsymbol{d}_k\}.$$

  end
  Set $\boldsymbol{d}_k = \{(m_i^{(k)}, \mathcal{X}_i^{(k)})\}_{i=1}^{N_k}$.
end

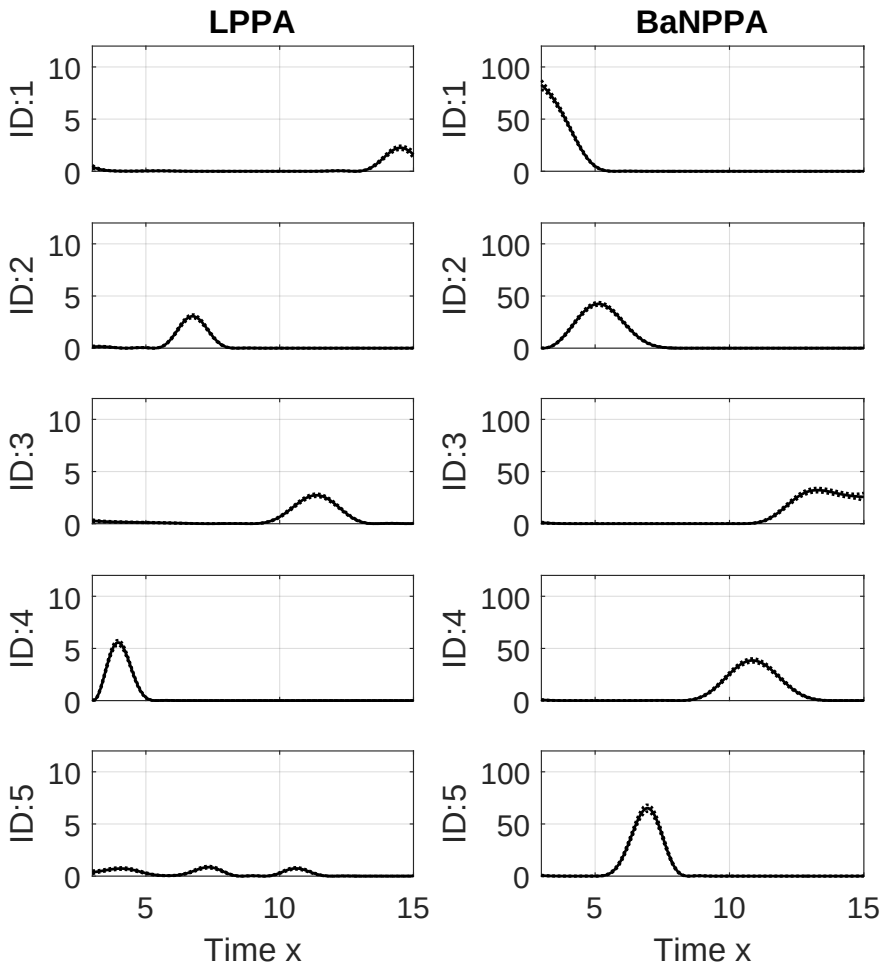

Figure 5.2: Latent functions in the Poisson process allocation of the Microblog data set. (Left column) First five latent functions from LPPA with $L = 14$. (Right column) First five latent functions from BaNPPA.

References

[1] Abramowitz, M. and Stegun, I. A. (1965). *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation.

[2] Adams, R. P., Murray, I., and MacKay, D. J. (2009). Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 9–16. ACM.

[3] Ancarani, L. and Gasaneo, G. (2008). Derivatives of any order of the confluent hypergeometric function f 1 1 (a, b, z) with respect to the parameter a or b. *Journal of Mathematical Physics*, 49(6):063508.

[4] Bauer, M., van der Wilk, M., and Rasmussen, C. E. (2016). Understanding probabilistic sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems*, pages 1533–1541.

[5] Bayarri, M. J. and Berger, J. O. (2004). The interplay of Bayesian and frequentist analysis. *Statistical Science*, pages 58–80.

[6] Benavoli, A. and Mangili, F. (2015). Gaussian processes for Bayesian hypothesis tests on regression functions. In *Artificial Intelligence and Statistics*, pages 74–82.

[7] Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., West, M., et al. (2003). The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures. *Bayesian statistics*, 7:453–464.

[8] Bertsekas, D. P. (2014). *Constrained optimization and Lagrange multiplier methods*. Academic press.

[9] Betensky, R. A., Lindsey, J. C., Ryan, L. M., and Wand, M. P. (1999). Local em estimation of the hazard function for interval-censored data. *Biometrics*, 55(1):238–245.

[10] Blackwell, D., MacQueen, J. B., et al. (1973). Ferguson distributions via pólya urn schemes. *The annals of statistics*, 1(2):353–355.

[11] Blei, D. M., Jordan, M. I., et al. (2006). Variational inference for Dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143.

[12] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

[13] Box, G. E. and Tiao, G. C. (2011). *Bayesian inference in statistical analysis*, volume 40. John Wiley & Sons.

[14] Campbell, M., Hoane Jr, A. J., and Hsu, F.-h. (2002). Deep blue. *Artificial intelligence*, 134(1-2):57–83.

[15] Cook, R. J. and Lawless, J. (2007). *The Statistical Analysis of Recurrent Events*. Springer Science & Business Media.

[16] Daley, D. J. and Vere-Jones, D. (2007). *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media.

[17] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38.

[18] Diggle, P. (1985). A kernel method for smoothing point process data. *Applied statistics*, pages 138–147.

[19] Diggle, P. J., Moraga, P., Rowlingson, B., and Taylor, B. M. (2013a). Spatial and spatio-temporal log-Gaussian Cox processes: extending the geostatistical paradigm. *Statistical Science*, pages 542–563.

[20] Diggle, P. J., Moraga, P., Rowlingson, B., Taylor, B. M., et al. (2013b). Spatial and spatio-temporal log-Gaussian Cox processes: extending the geostatistical paradigm. *Statistical Science*, 28(4):542–563.

[21] Ding, H. and Wu, J. (2015). Predicting retweet scale using log-normal distribution. In *Multimedia Big Data (BigMM), 2015 IEEE International Conference on*, pages 56–63. IEEE.

[22] Donner, C. and Opper, M. (2018). Efficient Bayesian inference of sigmoidal Gaussian Cox processes. *arXiv preprint arXiv:1808.00831*.

[23] Durrett, R. (2010). *Probability: theory and examples*. Cambridge university press.

[24] Famoye, F. (1995). Continuous Univariate Distributions, Volume 1.

[25] Fan, C.-P. S., Stafford, J., and Brown, P. E. (2011). Local-EM and the EMS algorithm. *Journal of Computational and Graphical Statistics*, 20(3):750–766.

[26] Felsenstein, J. (1985). Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, 39(4):783–791.

[27] Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230.

[28] Flaxman, S., Teh, Y. W., Sejdinovic, D., et al. (2017). Poisson intensity estimation with reproducing kernels. *Electronic Journal of Statistics*, 11(2):5081–5104.

[29] Flaxman, S., Wilson, A., Neill, D., Nickisch, H., and Smola, A. (2015). Fast Kronecker inference in Gaussian processes with non-Gaussian likelihoods. In *International Conference on Machine Learning*, pages 607–616.

[30] Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York, NY, USA:.

[31] Gallager, R. G. (2013). *Stochastic processes: theory for applications*. Cambridge University Press.

[32] Gao, S., Ma, J., and Chen, Z. (2015). Modeling and predicting retweeting dynamics on microblogging platforms. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 107–116. ACM.

[33] Gilks, W. R., Richardson, S., and Spiegelhalter, D. (1995). *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC.

[34] Gopalan, P., Ruiz, F. J., Ranganath, R., and Blei, D. (2014). Bayesian nonparametric Poisson factorization for recommendation systems. In *Artificial Intelligence and Statistics*, pages 275–283.

[35] Görür, D. and Rasmussen, C. E. (2010). Dirichlet process Gaussian mixture models: Choice of the base distribution. *Journal of Computer Science and Technology*, 25(4):653–664.

[36] Gray, R. M. and Gray, R. (1988). *Probability, random processes, and ergodic properties*. Springer.

[37] Greer, J. E. and McCalla, G. I. (2013). *Student modelling: the key to individualized knowledge-based instruction*, volume 125. Springer Science & Business Media.

[38] Gunter, T., Lloyd, C., Osborne, M. A., and Roberts, S. J. (2014). Efficient Bayesian nonparametric modeling of structured point processes. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pages 310–319. AUAI Press.

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

[40] Hensman, J., Durrande, N., and Solin, A. (2017). Variational Fourier features for Gaussian processes. *The Journal of Machine Learning Research*, 18(1):5537–5588.

[41] Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, page 282. Citeseer.

[42] Hjort, N. L., Holmes, C., Müller, P., and Walker, S. G. (2010). *Bayesian nonparametrics*, volume 28. Cambridge University Press.

[43] Hjort, N. L. and Jones, M. C. (1996). Locally parametric nonparametric density estimation. *The Annals of Statistics*, pages 1619–1647.

[44] Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

[45] Ihler, A. T. and Smyth, P. (2007). Learning time-intensity profiles of human activity using non-parametric Bayesian models. In *Advances in Neural Information Processing Systems*, pages 625–632.

[46] Jarrett, R. (1979). A note on the intervals between coal-mining disasters. *Biometrika*, 66(1):191–193.

[47] John, S. and Hensman, J. (2018). Large-scale Cox process inference using variational Fourier features. *arXiv preprint arXiv:1804.01016*.

[48] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

[49] Khan, M., Mohamed, S., Marlin, B., and Murphy, K. (2012). A stick-breaking likelihood for categorical data analysis with latent Gaussian models. In *Artificial Intelligence and Statistics*, pages 610–618.

[50] Khan, M. E., Aravkin, A., Friedlander, M., and Seeger, M. (2013). Fast dual variational inference for non-conjugate latent gaussian models. In *International Conference on Machine Learning*, pages 951–959.

[51] Khan, M. E. and Lin, W. (2017). Conjugate-computation variational inference: Converting variational inference in non-conjugate models to inferences in conjugate models. *arXiv preprint arXiv:1703.04265*.

[52] Kingman, J. F. C. (1992). *Poisson processes*, volume 3. Clarendon Press.

[53] Kottas, A. (2006). Dirichlet process mixtures of beta distributions, with applications to density and intensity estimation. In *Workshop on Learning with Nonparametric Bayesian Methods, 23rd International Conference on Machine Learning (ICML)*.

[54] Le Gall, F. (2014). Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th international symposium on symbolic and algebraic computation*, pages 296–303. ACM.

[55] Lewis, P. W. and Shedler, G. S. (1979). Simulation of nonhomogeneous Poisson processes by thinning. *Naval research logistics quarterly*, 26(3):403–413.

[56] Lian, W. (2015). *Predictive Models for Point Processes*. PhD thesis, Duke University.

[57] Lian, W., Henao, R., Rao, V., Lucas, J., and Carin, L. (2015). A multi-task point process predictive model. In *International Conference on Machine Learning*, pages 2030–2038.

[58] Lin, D. Y., Wei, L.-J., Yang, I., and Ying, Z. (2000). Semiparametric regression for the mean and rate functions of recurrent events. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):711–730.

[59] Liu, H. and Brown, D. E. (2003). Criminal incident prediction using a point-pattern-based density model. *International journal of forecasting*, 19(4):603–622.

[60] Lloyd, C., Gunter, T., Osborne, M., and Roberts, S. (2015). Variational inference for Gaussian process modulated Poisson processes. In *International Conference on Machine Learning*, pages 1814–1822.

[61] Lloyd, C., Gunter, T., Osborne, M., Roberts, S., and Nickson, T. (2016). Latent point process allocation. In *Artificial Intelligence and Statistics*, pages 389–397.

[62] Mayo, D. G. and Spanos, A. (2006). Philosophy of statistics. In *Philosophy of science: an encyclopedia*. Citeseer.

[63] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092.

[64] Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (2013). *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.

[65] Mikami, H., Suganuma, H., Tanaka, Y., Kageyama, Y., et al. (2018). Imagenet/resnet-50 training in 224 seconds. *arXiv preprint arXiv:1811.05233*.

[66] Miller, A., Bornn, L., Adams, R., and Goldsberry, K. (2014). Factorized point process intensities: A spatial analysis of professional basketball. In *International Conference on Machine Learning*, pages 235–243.

[67] Mitchell, T. M. (2006). *The discipline of machine learning*, volume 9. Carnegie Mellon University, School of Computer Science, Machine Learning Department Pittsburgh, PA.

[68] Møller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998). Log Gaussian Cox processes. *Scandinavian journal of statistics*, 25(3):451–482.

[69] Moser, S. M. (2007). Some expectations of a non-central chi-square distribution with an even number of degrees of freedom. In *TENCON 2007-2007 IEEE Region 10 Conference*, pages 1–4. IEEE.

[70] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

[71] Nasrabadi, N. M. (2007). Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901.

[72] Ogata, Y. (1981). On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31.

[73] Orbanz, P. and Teh, Y. W. (2011). Bayesian nonparametric models. In *Encyclopedia of Machine Learning*, pages 81–89. Springer.

[74] Paisley, J. (2010). Two useful bounds for variational inference. Technical report, Technical report, Department of Computer Science, Princeton University, Princeton, NJ.

[75] Paisley, J., Blei, D. M., and Jordan, M. I. (2012). Variational Bayesian inference with stochastic search. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1363–1370. Omnipress.

[76] Pitman, J., Tran, N. M., et al. (2015). Size-biased permutation of a finite sequence with independent and identically distributed terms. *Bernoulli*, 21(4):2484–2512.

[77] Rasmussen, C. E. (2004). Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer.

[78] Rasmussen, J. G. (2018). Lecture notes: Temporal point processes and the conditional intensity function. *arXiv preprint arXiv:1806.00221*.

[79] Rifkin, J. (1995). *The end of work: The decline of the global labor force and the dawn of the post-market era*. ERIC.

[80] Robert, C. and Casella, G. (2013). *Monte Carlo statistical methods*. Springer Science & Business Media.

[81] Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pages 832–837.

[82] Ross, S. (2009). *A First Course in Probability 8th Edition*. Pearson.

[83] Ross, S. M. (1996). Stochastic processes. 1996.

[84] Samo, Y.-L. K. and Roberts, S. (2015). Scalable nonparametric Bayesian inference on point processes with Gaussian processes. In *International Conference on Machine Learning*, pages 2227–2236.

[85] Schmidt, M., Berg, E., Friedlander, M., and Murphy, K. (2009). Optimizing costly functions with simple constraints: A limited-memory projected quasi-Newton algorithm. In *Artificial Intelligence and Statistics*, pages 456–463.

[86] Seeger, M., Williams, C., and Lawrence, N. (2003). Fast forward selection to speed up sparse Gaussian process regression. In *Artificial Intelligence and Statistics 9*, number EPFL-CONF-161318.

[87] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484.

[88] Snelson, E. and Ghahramani, Z. (2006). Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264.

[89] Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959.

[90] Stark, H. and Woods, J. W. (1986). *Probability, random processes, and estimation theory for engineers*. Prentice-Hall, Inc.

[91] Sun, J. and Zhao, X. (2016). *Statistical Analysis of Panel Count Data*. Springer.

[92] Sun, S. and Xu, X. (2011). Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 12(2):466–475.

[93] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.

[94] Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2005). Sharing clusters among related groups: Hierarchical Dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392.

[95] Thall, P. F. and Lachin, J. M. (1988). Analysis of recurrent events: Nonparametric methods for random-interval count data. *Journal of the American Statistical Association*, 83(402):339–347.

[96] Titsias, M. K. (2009). Variational model selection for sparse Gaussian process regression. *Report, University of Manchester, UK*.

[97] Turner, R. E. and Sahani, M. (2011). Two problems with variational expectation maximisation for time-series models. *Bayesian Time series models*, 1(3.1):3–1.

[98] Walder, C. J. and Bishop, A. N. (2017). Fast Bayesian intensity estimation for the permanental process. In *International Conference on Machine Learning*, pages 3579–3588.

[99] Wang, C., Paisley, J., and Blei, D. (2011). Online variational inference for the hierarchical Dirichlet process. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 752–760.

[100] Wellner, J. A. and Zhang, Y. (2000). Two estimators of the mean of a counting process with panel count data. *Annals of Statistics*, pages 779–814.

[101] Wellner, J. A., Zhang, Y., et al. (2007). Two likelihood-based semiparametric estimation methods for panel count data with covariates. *The Annals of Statistics*, 35(5):2106–2142.

[102] Williams, C. K. and Rasmussen, C. E. (1996). Gaussian processes for regression. In *Advances in neural information processing systems*, pages 514–520.

[103] Wu, L. L. and Tuma, N. B. (1990). Local hazard models. *Sociological methodology*, pages 141–180.

[104] Wuertz, D., Wuertz, M. D., and Team, R. C. (2007). The fasianoptions package.

[105] Zhang, Y. and Jamshidian, M. (2003). The gamma-frailty Poisson model for the nonparametric estimation of panel count data. *Biometrics*, 59(4):1099–1106.