

Statistical Analysis of Recurrent events by Point process

Gyeongtae Hong, B.Sc., M.Sc.

SUBMITTED FOR THE DEGREE OF DOCTOR OF STATISTICS AT
LANCASTER UNIVERSITY.

JANUARY 16, 2019



Abstract

Characterising the neuron spike train firing as a function of external stimulus applied in an experiment and intrinsic dynamics of neurons such as absolute and relative refractory periods, history effects are important in neuroscience. Such a characterisation is very complex and the broad class of models to capture such details are required consistently. One of the useful method which characterising neuron spike trains activity is a point process model. For instance, they have successfully characterised spiking activity of rat hippocampal place cells and sea hare nerve cells.

In general there are two approaches estimating the point process. One is the parametric modelling and there are many parametric point process models based on likelihood analysis. The self exciting process is carried out with history dependence which were selected by decaying function of effect of history. A simulation study is performed by Thinning method to check the self exciting process with selected history dependence reflects well neuron firings. Another is non-parametric method, point process based on B-spline basis function is carried out to characterise the single neuron firing rate and the FPCA (Functional Principal Component Analysis) is also performed to consider the dominant mode of variation of the functional data from the same session. In addition, the mFPCA (Multivariate Functional Principal Component Analysis) is applied to take into account the variation of a different session as well. The comparison of these methods with the same data set is performed in Chapter 6.

Acknowledgement

Above all things, I would like to express a debt of gratitude to my supervisor Juhyun Park. My depth of gratitude is too great to express in words. I did not have enough capacity of knowledge to carry on the Ph.d, however she always led me to right way with wise instruction. After every meeting she provided a helpful commend and discussed about next step of the study. Without her guidance I would not finish this study.

I would like to thank my family. Whenever I felt desperate because of the Ph.d life, my family gave me an infinite encouragement. Our married life started in U.K, because we came to U.K just after marriage. Therefore there were any friends of my wife, but she withstood loneliness in foreign country showed me a smile everyday. She has been a very important support.

Everyone in B18 I have got many encourage from them, even if the field of topic were different they tried to help me out to solve any of problems. They always create an atmosphere conducive to academic pursuit, hence it might lead to concentrate on my work.

Contents

1	Introduction, Motivation and Preliminaries of the study	10
1.1	Introduction	10
1.1.1	Neuron	11
1.1.2	Motivation	12
1.2	A motivating example	13
1.3	Organisation of the Thesis	16
1.4	Preliminaries	18
1.4.1	Parametric modelling of Point process data	18
1.4.2	Non-parametric modelling for point process data	22
2	Parametric point process modelling for single neuron	27
2.1	Introduction and example of data	27
2.2	Point process	29
2.2.1	Self exciting process	29
2.3	Simulation study	31
2.3.1	Inversion method	31
2.3.2	Thinning method	32
2.3.3	Analysing the simulated dataset	34
2.4	Application to data	35
2.5	Model checking ; Time-rescaling theorem	36
2.5.1	KS plots and Q-Q plots of a experimenta earthquake data	39
2.6	Discussion and conclusion	40
3	Non-parametric point process modelling for single neuron	41
3.1	Introduction	41

3.2	Methodology	42
3.2.1	Point process model	42
3.2.2	Nonparametric modeling of intensity functions of point process . .	45
3.2.3	History effects	47
3.2.4	Estimation inference	48
3.3	Algorithm	49
3.3.1	Iterative re-weighted least squares for non-linear model	49
3.4	Application to neuron spike train data	50
3.4.1	Experimental procedures	50
3.4.2	Application to data	50
3.5	Discussion and conclusion	52
4	Non-parametric point process modelling for multiple neurons	53
4.1	Introduction	53
4.2	Method	55
4.2.1	Principal component analysis	55
4.2.2	Functional density model with point process data	58
4.2.3	Kernel density	59
4.2.4	Choice of the number of principal components	62
4.2.5	Component as perturbations of the mean	63
4.3	Simulation study	64
4.3.1	Analysing the simulated dataset	64
4.4	Application to data	73
4.4.1	Individual intensity function	73
4.4.2	Functional principal component analysis	75
4.4.3	Model checking	77
4.5	Discussion and conclusion	79
5	Multivariate functional principal component analysis with multiple neurons	81
5.1	Introduction	81
5.2	Method	82

5.2.1	Multivariate principal component analysis	82
5.3	Simulation study	84
5.3.1	Scenario 1	84
5.3.2	Scenario 2	86
5.4	Application to data	88
5.5	Discussion and conclusion	89
6	Case study - The comparison of three approaches	92
6.1	Motivation and data description	92
6.2	Application to data.	99
6.2.1	Characterising single neuron activities by non-parametric modelling	99
6.2.2	Characterising neuron activities using functional data analysis . .	101
6.2.3	Characterising single neuron activities by parametric modelling . .	104
6.2.4	Characterising single neuron activities by Semi-parametric modelling	108
6.3	Discussion and conclusion	110
	Appendix A Graphics	112
A.1	Graphics for Chapter 3	112
A.2	Graphics for chapter 4	117

List of Figures

1.1	Drawing of neuron network	12
1.2	The raster plot of the Mwheel data.	14
1.3	The peri-stimuli histogram of the Mwheel data.	15
2.1	The earthquake events (left panel) and the distribution of the stress drop with its magnitude (right panel)	28
2.2	Intensity for the simulated data: true intensity (left panel), non-parametric es- timation (middle panel) and conditional intensity (Model 1, 2 and 3)	34
2.3	Intensity for the experimental earthquake events: non-parametric estimation (left panel) and conditional intensity (Model 1, 2 and 3)	36
2.4	Q-Q plot and KS plot for distribution of rescaled intervals for the model 1, 2, and 3.	39
3.1	Bspline basis function	47
3.2	Kernel estimation using Bspline basis function and Kolmogorov-Smirnov plot, (black line-original data, red line-modified data)	51
3.3	Kernel estimation using Bspline basis function and Kolmogorov-Smirnov plot, (black line-original data, red line-modified data)	52
4.1	The intensity function of the true and simulated data.	66
4.2	The kolmogrov-smirnov plot for the simulated data.	66
4.3	The intensity function of the true and simulated data.	67
4.4	The kolmogrov-smirnov plot for the simulated data.	67
4.5	The intensity function of the true and simulated data.	69
4.6	The kolmogrov-smirnov plot for the simulated data.	69
4.7	The intensity function of the true and simulated data.	71

4.8	The kolmogrov-smirnov plot for the simulated data.	71
4.9	The individual neuron firing rate function.	73
4.10	The eight days for Kolmogrove-Smirnov plot for the task on the Mwheel (EC 3).	74
4.11	The first four principal component curves of the neuron firing data on the Mwheel task.	75
4.12	The overall mean neuron firing rate function and the effect of adding and subtracting a suitable multiple of each principal component curve.	77
4.13	The random rate function for the neuron firing data.	78
4.14	Kolmogrov-smirnov plot for the neuron firing on the Mwheel task.	79
5.1	Intensity function of the true and simulated neuron and Kolmogrov-smirnov plot.	84
5.2	Intensity function of the true and simulated neuron and Kolmogrov-smirnov plot.	85
5.3	Intensity function of the true and simulated neuron and Kolmogrov-smirnov plot.	86
5.4	Intensity function of the true and simulated neuron and Kolmogrov-smirnov plot.	87
5.5	The individual neuron firing rate function.	89
5.6	The first four principal component curves of the neuron firing.	90
5.7	The Kolmogrove-Smirnov plot for the task on the Bigsquare.	91
6.1	Spike of eight neurons within 1 ms bin in 3 seconds	93
6.2	Spike of eight neurons within 5 ms bin in 3 seconds	94
6.3	Histograms of the inter-spike time for eight neurons of EC582.	94
6.4	Histograms of the inter-spike time for eight neurons of EC397.	95
6.5	Histograms of the inter-spike time for eight neurons of EC430.	95
6.6	Histograms of the inter-spike time for eight neurons of EC448.	96
6.7	The individual intensity functions of four sessions by kernel density estimation.	98
6.8	The individual intensity functions of four sessions by bspline basis function.	99
6.9	The Kolmogorov-smirnov plot of four sessions by bspline basis function.	100
6.10	The first four principal component curves of the EC582.	101
6.11	The first four principal component curves of the EC397.	102
6.12	The mean functions of the EC582 with four principal component curves.	103
6.13	The individual intensity functions of the EC582 with four principal component curves.	105
6.14	The score of the all session and trials on the first two principal components.	105

6.15	The individual intensity functions of the EC582 with four principal component curves after deleting outliers.	106
6.16	The Kolmogorov-smirnov plot of four sessions by bspline basis function.	106
6.17	The individual intensity functions by Hawkes process.	107
6.18	The Kolmogorov-smirnov plot of session EC 582 by Hawkes process and FPCA.	109
6.19	The individual intensity functions by Hawkes process with time varying base intensity function.	109
6.20	The individual intensity functions by Hawkes process with time varying base intensity function.	110
A.1	Raster plot of 25 input data	112
A.2	Raster plot of 18 output data	113
A.3	Kernel estimation using Bspline basis function	113
A.4	Kernel estimation using Bspline basis function	114
A.5	kormogrov-smirnov plot, (black line-original data, red line-modified data)	114
A.6	kormogrov-smirnov plot, (black line-original data, red line-modified data)	115
A.7	kormogrov-smirnov plot, (black line-original data, red line-modified data)	115
A.8	kormogrov-smirnov plot, (black line-original data, red line-modified data)	116
A.9	kormogrov-smirnov plot, (black line-original data, red line-modified data)	116
A.10	The comparison of true intensity function and the individual intensity function from functional data analysis without PCA.	117

List of Tables

1.1	Duration of the session by days	15
1.2	The number of neuron firings	16
2.1	Estimated parameters for conditional intensity function and the negative Log-likelihood value	34
2.2	Estimated parameters for conditional intensity function and the - log-likelihood value	35
4.1	Mean squared error depending on the number of simulations	65
4.2	Comparison of mean squared error of the individual intensity and object-specific intensity	68
4.3	Comparison of mean squared error of the individual intensity and object-specific intensity	68
5.1	Proportion of principal component and the proportion of contribution of each region.	91
6.1	Summary of the inter-spike time for the EC582	96
6.2	Summary of the inter-spike time for the EC397	97
6.3	Summary of the cleaned inter-spike time for the EC582	97
6.4	Summary of the cleaned inter-spike time for the EC397	97

Chapter 1

Introduction, Motivation and Preliminaries of the study

1.1 Introduction

One type of complex recurrent events observed in many modern neuroscience experiments, is known as spike train data. In simple terms, these are represented as a sequence of event times and are considered to represent our brain activities. Understanding and characterising the activity patterns and their variations constitute important steps in many scientific investigations.

A point process modelling framework is commonly employed to analyse these types of data [7, 8, 1, 2, 9]. Nevertheless, due to the advancement in technology, more and more complex forms of data, from multiple brain regions under various experimental conditions for a longer period in a much finer resolution, are available and thus there is a continuous need to develop more sophisticated tools to utilise additional information and exploit the potentially rich structure in such data. Also with increasing amount of data collection, there is much higher variation across different recordings which could potentially be related to the quality of data. Scientists tend to perform various cleaning and preprocessing steps to ensure the quality and to choose appropriate experimental datasets to analyse but are generally less interested in quantifying the variations themselves. Whereas it could be argued that understanding and quantifying these variations are the principal objectives of statistical analysis, which could help us gain a deeper understanding and a better insight.

Motivated by these modern challenges, this thesis investigates various statistical approaches to handling the complexity of such types of data, with a particular attention to flexible nonparametric approaches. A point process modelling approach is well developed for a single process, but its extension to multiple point processes is quite challenging in general. Also the analysis of multivariate recurrent events tends to focus on low frequency events, whereas the spike neuron dataset tends to be of very high frequency. To address some of these limitations, we will focus on more flexible nonparametric approaches, in the spirit of functional data analysis [21]. Functional data analysis provides a flexible and efficient tool to simultaneously analyse data from multiple subjects. However, traditionally functional data analysis is considered for the analysis of continuous and curve-like datasets, but rarely used for recurrent events of high frequency. In this work, we consider both parametric and nonparametric approaches in the standard point process modelling framework and also develop its extension by combining ideas from functional data analysis viewpoint. Instead of directly working with the observations as events, we can utilise the functional representation of the intensity function of the events. This perspective is also more advantageous when dealing with high volume of data, that is, the number of observation is huge.

Before proceeding, we give a brief background of the subject on neurons and some related literature.

1.1.1 Neuron

Neurons are nerve cells and carry information between the brain and other parts of the body. Many scientists believe that the communication system of neurons could be viewed as an input output system of a particular structure which has important functions. Brain areas accumulate, process and transmit information through the stereotype of electrical events called spike trains or action potentials. Figure 1.1 shows the structure of a typical neuron. A neuron consist of a cell body (soma), dendrites, synapse and an axon. The dendrites typically branch abundantly and they collect the information. It is propagated to the soma and along the axon then activates the synapse to communicate with other neurons.

As we mention briefly the mechanism of propagation of the information, at first the dendrite branches absolve the input from other neurons through the chemical process

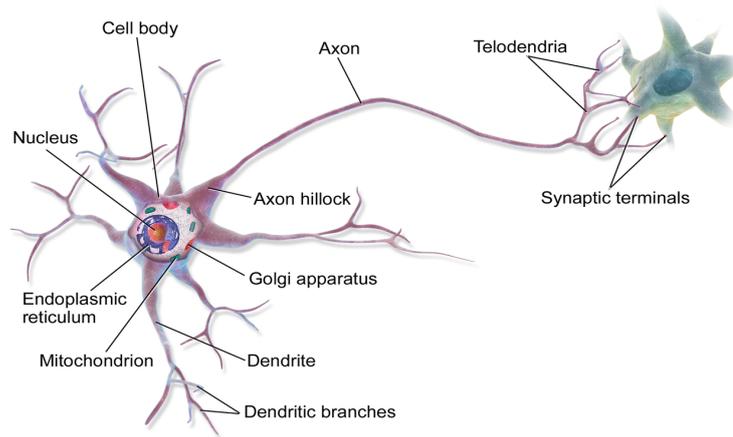


Figure 1.1: Drawing of neuron network .

then this input is delivered to the soma. From that time the input is converted to a membrane potential. At the axon hillock, which is considered as a trigger zone, the membrane potential sometimes reaches a threshold and then fires. This stage creates the action potential or spike train. This action potential propagates along the axon to the synapses and sends message to other neurons.

1.1.2 Motivation

Neuroscience is the scientific study of the nervous system. It is recognized as a branch of biology. However, it is currently an interdisciplinary science that collaborates with other fields such as chemistry, cognitive science, computer science, engineering and mathematics. This concerns with how the elements of the nervous system function and work together. The operation of the nervous system involves the chemical mechanisms, physical arrangement and electrical mechanism. A fundamental question is how a neuron responds to stimuli, transmits information and changes with alterations of the environment. In order to answer those questions, many researchers have tried to collect varied and extensive data sets, which include photographs made by electron microscopes, fluctuating voltages and electroencephalograms to record electrical activities of the brain.

In the earlier times, scientists studied the movements of the individual ions [7], [8]. They tried to represent the single neuron activity with several methods. A general mathematical tool for describing sequences of events occurring in time and space is the theory of point processes [4]. For example, a history-dependent conditional intensity model was considered for modelling a Hippocampal Place Cell[33].

Thanks to the rapid advancement in technology, nowadays the interest extends to the joint behaviour of the multiple components of the brain, such as Hippocampal ensemble activity, by spatio-temporal point processes [6] [20]. Similarly, the examination of multiple regions of hippocampal loop relationships is made in [16]. The brain region is often modelled as multiple-input/multiple-output (MIMO) or nonlinear dynamic systems [30], [31] [27] for the identification and modelling of functional connectivity in the brain using spike train data [32].

Although there have been many attempts to develop an effective statistical model, the modern neuron datasets are still too complex to be easily represented in a common framework. Although as the experimental study tends to focus on a very specific problem, it is difficult to generalise the approaches developed for one to the other scenarios. Besides, the massive datasets collected in any given experiment also create a huge variation within the dataset, and it is not clear how to compare the results from different experiments. Hence, given our limited knowledge in brain science, it would be useful to develop in parallel more advanced statistical modelling approaches to understand and characterise these variations.

As quantitative modelling has become an important component of neuroscience, many consortiums were created to make experimental neuron datasets publicly available to promote the practice of shared knowledge in science. This thesis uses the modern experimental neuron datasets available from <http://crcns.org> to study and develop statistical modelling approaches. Much of these datasets have not been considered in a real data analysis. As most of these datasets are directly coming from the experiments, compared to the well studied neuron data examples shown in the literature, they are also quite noisy in high volume and very complex. Although scientific implications can go much deeper, our main interest lies in statistical modelling of such types of data in a more realistic setting.

1.2 A motivating example

As one of our interests is to understand statistical variations, we study the spike neuron datasets that cover multiple experiments. Here we introduce one of our motivating examples. This data set contains multi-unit recordings from different rat hippocampal

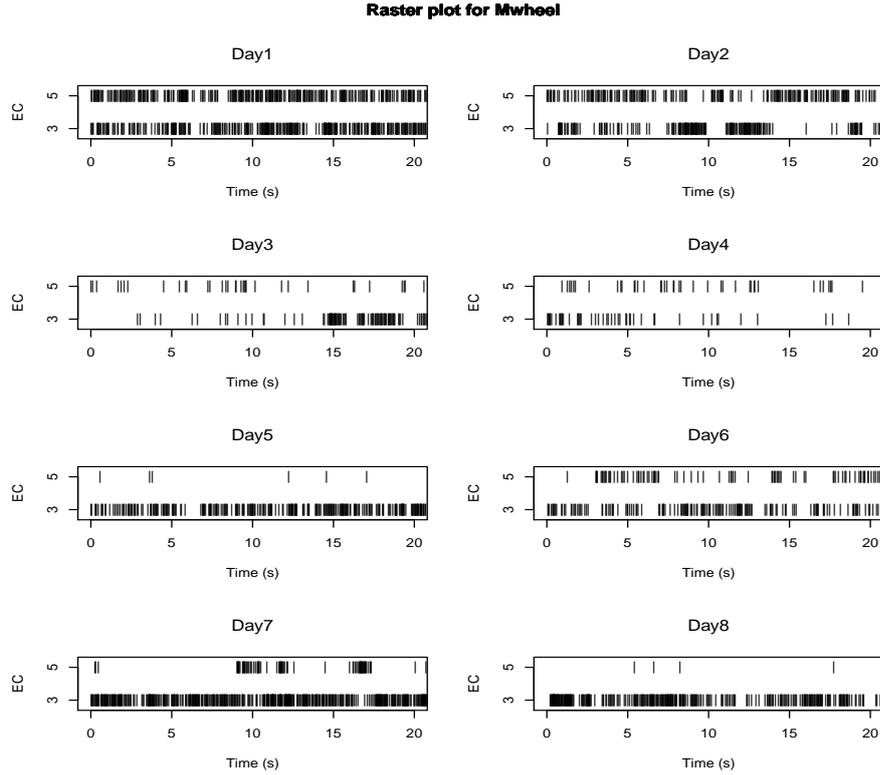


Figure 1.2: The raster plot of the Mwheel data.

regions while the animals were performing multiple behaviour as tasks. The data were recorded on two different brain regions which are EC3 and EC5 and the number of cells recorded are 201 and 110 respectively. There are two types of neuron cells which are called principal neurons and inter neurons. In this study, we focus on the principal neurons. The firing rate of inter neurons was too high relatively, almost acting as a background noise so it can dominate the effect of intensities.

The data set was obtained during 70 sessions during which the animal performed one of 3 different tasks such as Big-square, Mwheel and Linear, with 24, 28 and 18 sessions respectively. The Big-square is a $180\text{cm} \times 180\text{cm}$ box, the Mwheel is running a wheel and the Linear is a 250cm linear maze. These sessions were performed during the 16 days, 8 days for performing Mwheel task and 8 days for performing Big-square and Linear task. The Big-square task and Linear task were performed on the same day. Table 1.1 shows the duration of each session by days. The Big-square task seems to take longer to perform than the Linear task. The variation of the duration between days is similar and but there are some differences between tasks. Table 1.2 shows the number of neuron firings on each performing day and task. Even if a rat performed the same

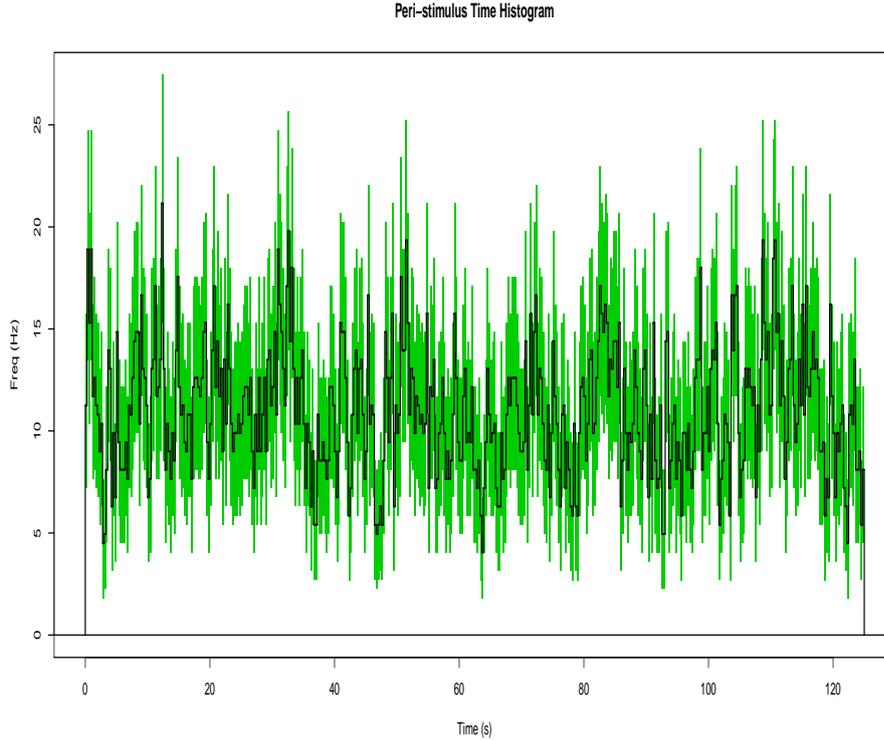


Figure 1.3: The peri-stimuli histogram of the Mwheel data.

task, the number of neuron firings recorded was quite different. It may happen that one day rat finished 10 repetitions of their task but one other day they only finished few repetitions of their task. The largest number of neuron firings was observed on the Big-square task and the smallest number was collected on the Linear task.

Table 1.1: Duration of the session by days

Days	Mwh EC3	Mwh EC5	B-S EC3	B-S EC5	Linear EC3	Linear EC5
1	2183.8	2210.0	3419.6	3413.6	1109.1	1103.0
2	1971.2	1971.0	3470.3	3476.1	820.9	823.0
3	2460.4	2451.7	3484.8	3483.7	1032.3	1031.1
4	2765.8	2766.1	3079.5	3126.8	1134.0	1035.8
5	2078.8	2273.7	3538.4	3533.7	809.1	803.3
6	2166.9	2167.1	2804.9	2796.2	1152.2	1016.1
7	2111.3	2110.2	3464.2	3464.3	1088.5	1088.6
8	1943.0	1943.1	3472.3	3471.6	824.2	823.7

This experiment was conducted for several days and it is recorded as a single data set. This single data contains up to ten repetitions of the accomplished task. Figure 1.2 is an example of visualisation of the raw data. These plots are called the raster plot which depicts the sequence of times of neuron firing events under the Mhwheel task for

Table 1.2: The number of neuron firings

Days	Mwhl EC3	Mwhl EC5	B-S EC3	B-S EC5	Linear EC3	Linear EC5
1	27407	25257	48140	5693	25549	2629
2	3902	7545	7344	11934	745	5871
3	11695	15774	61366	10698	14362	1991
4	9983	8218	44691	14040	2405	2232
5	37630	1911	16319	6292	6619	773
6	14884	12889	28312	849	13355	6812
7	43196	9703	21301	18456	11836	5447
8	33480	9501	70617	29470	25626	2891

8 days. Although helpful, the raster plot is exploratory and has limited information, so it is difficult to infer its firing pattern and any connection between EC3 and EC5. The main objectives of the analysis would be to develop and fit some analytic models for the basic features of neuron behaviour and to investigate any connection between neurons. In addition, the firing rate seems to differ from recording days. Except for two raster plots on the top of panel, EC3 brain region is more likely to fire than region EC5.

The peri-stimulus time histogram, abbreviated PSTH histogram, provides another view on the raw data through a histogram of the times between which neurons fire. An example of the peri-stimulus histogram is shown in Figure 1.3, the black line indicates the mean of instantaneous neuron firing rate for Mwheel task on the brain region EC 3. The 95% of confidence interval is shown as interval filled with green colour.

We will consider various statistical modelling approaches later appropriate to analyse such types of data. The main objective of this work is to systematically assess the performance of these methods under realistic settings.

1.3 Organisation of the Thesis

Our main methodology is driven by the need for the analysis of spike neuron data, but is not limited to them. Historically, similar methodologies have been considered for both neurophysiology and seismology [1]. This is especially true for low frequency data. We will begin with a relatively small dataset from seismology to demonstrate parametric approaches in Chapter 2. Nonparametric approaches in later chapters are mostly applied to spike neuron data. We note that the illustrating dataset may vary across the chapters, and will be introduced within each chapter if necessary. The choice simply reflects the availability of the particular dataset while working on a particular project during the

study period. For better comparison across the chapters, the final chapter is prepared as a case study to illustrate various approaches discussed in this work with a common dataset.

The thesis is organised as follows. It consists of six chapters:

- The first part of Chapter 1 explains what is a neuron and how the information propagate through the neuron, and the main purpose of studying neuron activities. A motivating example dataset was presented with descriptive analysis, raster plot and peri-stimulus time histogram. The remaining of the Chapter 1 covers some background knowledge of parametric and non-parametric modelling for point processes.
- The second chapter introduces the parametric point process modelling with self-exciting process (Hawkes process) and epidemic type aftershock sequence model. The example data set is an experiment involving earthquake data. The conditional intensity function is estimated using self-exciting process and the simple inhomogeneous Poisson process, and model checking is performed by time rescaling theorem.
- The third chapter introduces the non-parametric point process with, which the example of the neuron data. The generalised linear model is used to model the conditional intensity function of point process data by discretization of time into bins.
- The fourth chapter defines functional principal component analysis, and introduces functional density model with point process.
- The fifth chapter discusses multivariate functional principal component analysis. FPCA in chapter 4 considers a variation of the mode of each trial in the session whereas mFPCA (multivariate functional principal component analysis) takes into account the variation of the mode of the trials from different region of the brain.
- The final chapter gives a case study. Since the data set has been changed four times so the different data set were used in each chapter. Hence this chapter shows the comparison of three different methods with a new data set.

All implementations are done using statistical language R. The codes for computation are collected in the Appendix.

1.4 Preliminaries

1.4.1 Parametric modelling of Point process data

A temporal point process is a stochastic process composed of a time-series of binary events that occur in continuous time[5]. It describes data that are localised at a finite set of time points. As opposed to continuous-valued processes, which can take countless values at each point in time, a point process can take on only values indicating whether or not an event occurs at that time. Theoretically, this makes the probability models used to describe point process data relatively easy to express mathematically.

For purposes of both modeling and statistical analysis, the concept of intensity function and counting processes may be useful. Let T_1, T_2, \dots be random variables of the event time of a point process, where T_i is the time of the i th event. The realisation of a point process is the event $T_1 = t_1, T_2 = t_2, \dots$ for some collection of times $0 < t_1 < t_2 < \dots$. Let X_1, X_2, \dots be a set random variables describing the inter arrival time. In order to describe a set of events in terms of the number of events observed during the time interval, the total number of events until time point t is denoted by $N(t)$. In addition, let $N(s, t)$ denote the number of events occurred between time interval $(s, t]$. Events occur in continuous time but the model for recurrent events can be formulated in discrete time. The events in terms of discrete increments is denoted by $\Delta N(t) = N(t + \Delta t^-) - N(t^-)$, denoting the number of events in the interval $[t, t + \Delta t]$. Here t^- indicates the times that are infinitesimally smaller than t .

Poisson Process

The Poisson process is one of the most commonly used models for recurrent events, which explains how the events occur randomly and how many events occur in independent time intervals. The homogeneous Poisson process can be defined in three different ways.

- One event occurs in an infinitesimal time interval Δt and the probability of an event occurrence is $\lambda \Delta t$.

- The number of events $N(t)$ in interval up to time t follows the Poisson(λt) distribution.

$$Pr\{N(t) = n\} = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$$

- The inter arrival times are independent and they follow the Exp(λ) distribution.

$$Pr\{X_i > t\} = e^{-\lambda t}$$

These three approaches are all equivalent and use the same intensity function to characterise an event. The homogeneous Poisson process, as it was defined above, is characterised by a constant arrival rate λ , but it is inappropriate to use when the arrival rate varies as a function of time. In order to consider a time varied rate function, the intensity function $\lambda(t)$ which expresses the instantaneous occurring probability and completely defines a probability model for the point process [23].

$$\lambda(t) = \lim_{\Delta t \rightarrow 0} \frac{Pr(\Delta N_{(t, t+\Delta t)})}{\Delta t} \quad (1.1)$$

For example, we can now define a Poisson process function as the limit of a Bernoulli process with an event probability determined by the intensity function. As the time interval we consider becomes sufficiently small, no more than one events occur in any bin, so we only need to characterise the probability. The probability of not occurring an event in that bin is then $1 - p$, and the probability of occurring more than one event in that bin is negligibly small. Then the probability of an event occurring can be analysed as Bernoulli process, the probability of an event in small interval;

$$Pr(\Delta N(t) = 1) \approx \lambda(t)\Delta t$$

and that of no event in that small interval:

$$Pr(\Delta N(t) = 0) \approx 1 - \lambda(t)\Delta t.$$

We can compute the distribution of the time to the next experimental earthquake occurrence given the previous occurring time by noting that the event that the time until the

next event is greater than some time t_i , i.e. $\{T_i > t_i \mid T_{i-1} = t_{i-1}\}$, is equivalent to the event that no event occurs in the interval (t_{i-1}, t_i) . Therefore $Pr(T_i > t_i \mid T_{i-1} = t_{i-1}) = Pr(\Delta N_{(t_{i-1}, t_i]} = 0) = \exp\{-\int_{t_{i-1}}^{t_i} \lambda(u)du\}$, and the cumulative distribution function (CDF) is $Pr(T_i \leq t_i \mid T_{i-1} = t_{i-1}) = 1 - \exp\{-\int_{t_{i-1}}^{t_i} \lambda(u)du\}$. The probability density function of the next experimental earthquake occurrence time is given by derivation of CDF, $f_{T_i}(t_i \mid T_{i-1} = t_{i-1}) = \frac{d}{dt_i} \left[1 - \exp\{-\int_{t_{i-1}}^{t_i} \lambda(u)du\} \right]$ which implies,

$$f_{T_i}(t_i \mid T_{i-1} = t_{i-1}) = \lambda(t_i) \exp\left\{-\int_{t_{i-1}}^{t_i} \lambda(u)du\right\}$$

$$\lambda(t) = \frac{f(t)}{1 - F(t)} \quad (1.2)$$

where $f(t)$ is the density function and $F(t)$ is the distribution function of the inter arrival times. We now derive the probability density function for an event process that is observed over the fixed time interval $[0, \tau]$. From the equation (1.2), the probability density function is given by $f(t) = \lambda(t)(1 - F(t))$. The probability density of the event occurrence (n events, at time $t_1 < \dots < t_n$) for a process with intensity over the specified interval is

$$\prod_{i=1}^n \lambda(t_i) \cdot \exp\left\{-\int_0^\tau \lambda(u)du\right\}.$$

The parametric rate function can be specified as $\exp(\beta_0 + \beta_1 t_i)$. The likelihood function for non-homogeneous Poisson process is given by

$$\begin{aligned} l(\theta) &= \prod_{i=1}^n \{\lambda(t_i)\} \cdot \exp\left\{-\int_s^t \lambda(u)du\right\} \\ &= \prod_{i=1}^n \{\exp\{\beta_0 + \beta_1 t_i\}\} \cdot \exp\left\{-\int_s^t \exp(\beta_0 + \beta_1 u)du\right\} \\ &= \prod_{i=1}^n [\exp\{\beta_0 + \beta_1 t_i\}] \cdot \exp\left[\left\{\frac{\exp\{\beta_0 + \beta_1 s\}}{\beta_1} - \frac{\exp\{\beta_0 + \beta_1 t\}}{\beta_1}\right\}\right] \\ l(\theta) &= \sum_{i=1}^n \{\beta_0 + \beta_1 t_i\} \cdot \left[\left\{\frac{\exp\{\beta_0 + \beta_1 s\}}{\beta_1} - \frac{\exp\{\beta_0 + \beta_1 t\}}{\beta_1}\right\}\right] \end{aligned} \quad (1.3)$$

Point Process with history dependence

Many systems that produce point process data have a history dependent structure that makes the Poisson process inappropriate. It is useful to define a unified mathematical construct that will allow to describe a physically relevant point process. A new model for recurrent event data can be specified by considering the probability distribution for the number of events in short intervals $[t, t + \Delta t]$, given the history of the previous event. Now consider the history dependence on the non-homogeneous model which can be specified very generally by considering the probability distributions for the number of events in short intervals $[t, t + \Delta t]$, given the history of the events before the time t . The history $H(t)$ of the event process at time t is defined as

$$H(t) = \{N(s) : 0 \leq s < t\}$$

When the value $N(0)$ is included in $H(t)$, typically $H(t)$ should be treated as 0.

We consider the conditional intensity function [24] for the event process defined as

$$\lambda(t | H(t)) = \lim_{\Delta t \rightarrow 0} \frac{Pr\{\Delta N(t) = 1 | H(t)\}}{\Delta t}, \quad (1.4)$$

where $Pr\{\Delta N(t) = 1 | H(t)\}$ is the instantaneous conditional probability of an event. The Δt value should be sufficiently small in order to avoid multiple events occurring in a single bin. Additional important intuition behind the conditional intensity function can be gained by choosing Δt to be a small interval and re-expressing equation (1.4) as

$$Pr\{\Delta N(t) = 1 | H(t)\} \approx \lambda(t | H(t))\Delta t. \quad (1.5)$$

Equation (1.5) states that the conditional intensity function multiplied by Δt gives the probability of an occurrence in a small time interval Δt . The likelihood function, for a process with intensity function (1.4), over the specified interval $[0, \tau]$ is

$$\prod_{i=1}^n \lambda(t_i | H(t_i)) \cdot \exp \left\{ - \int_0^\tau \lambda(u | H(u)) du \right\}. \quad (1.6)$$

The first term of $\prod_{i=1}^n \lambda(t_i | H(t_i))$ characterises the distribution of the event occurrences at exactly the observed times and the second term of $\exp \left\{ - \int_0^\tau \lambda(u | H(u)) du \right\}$ gives

the probability of not occurring any events during the observation interval $(0, \tau]$. Hence the equation (1.6) represents the distribution of events occurring only at the observed times. The exponential term in (1.6) is obtained from the product of integration for the interval defined by each observed event times. If the parametric intensity function can be specified as $\exp(\beta_0 + \beta_1 t + \beta_2(t - t_{N(t^-)}))$, then the simplified log-likelihood function is shown as;

$$\begin{aligned} l(\theta) &= \sum_{i=1}^n \log \left[\lambda(t_i | H(t_i)) \cdot \left\{ - \int_0^\tau \lambda(u | H(u)) du \right\} \right] \\ &= \sum_{i=1}^n (\beta_0 + \beta_1 t + \beta_2 z(t)) - \int_0^\tau \exp(\beta_0 + \beta_1 u + \beta_2 z(u)) du \end{aligned} \quad (1.7)$$

where $z(t) = I(N(t^-) > 0)(t - t_{N(t^-)})$ denotes the time since the last event.

1.4.2 Non-parametric modelling for point process data

Generalised linear model

Generalised linear models are a large class of statistical model for relating responses to linear combinations of predicted variables, including many commonly encountered types of dependent variables and error structures as special cases. The GLM approach is attractive because it provides a general theoretical framework for many commonly encountered statistical models and it enables the implementation of these different models in statistical software, since essentially the same algorithm can be used for the estimation, inference and accessing model adequacy for all GLMs.

$$y_i = x_i \beta_i^T + \epsilon_i, \quad i = 1, \dots, n$$

where x_i is a vector of k independent variables, β is a k by 1 vector of unknown parameters and the ϵ_i are mean-zero stochastic disturbances. Typically, the ϵ_i are assumed to be independent across observations with constant variance σ^2 and distributed normal. That is, normal linear regression model is characterised by following features;

- stochastic components: the y_i are assumed to have independent normal distributions with $E(y_i) = \lambda_i$, with constant variance σ^2

- systematic components: the covariates x_i combines linearly with coefficients to form the linear predictor $\eta_i = x_i^T \beta$
- link between the random and systematic components: the linear predictor $x_i^T \beta = \eta_i$ is a function of mean parameter λ_i via a link function, $g(\lambda_i)$.

In this study we assume the intensity function is always positive, then we fit the Poisson regression which assumes

$$N(t) \sim \text{Poisson}(\lambda_i), i = 1, \dots, N$$

The canonical link is the log,

$$\log(\lambda_i) = \eta = x_i^T \beta.$$

A Poisson model with a log link is commonly called a loglinear model. Under the Poisson model the variance function is $V(\lambda) = \lambda$. When overdispersion is evident, it is usually handled by a scale parameter, $Var(Y_i) = \pi \lambda_i$. To fit this model, iterative weighted least square method was used. Suppose the current estimates are $\hat{\beta}$, yielding estimated probability λ_i . We form the linearised response $z_i = \eta + (Y_i - \lambda_i)/w_i$. A new $\hat{\beta}$ is obtained by weighted linear regression of z_i onto the x_i with weight w_i .

$$\hat{\beta} = (X^T W X)^{-1} X^T W z,$$

This will repeated until $\hat{\beta}$ converges.

The dependent variable y follows the exponential family,

$$f(y; \theta) = \exp \left\{ \frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi) \right\},$$

where θ is the canonical parameter and $E(y) = \lambda = b'(\theta)$ and $\text{var}(y) = a(\phi)b''(\theta)$. For example in Poisson distribution, $y \sim \text{Poisson}(\lambda)$, then the log-probability mass function is

$$\log f = y \log \lambda - \lambda + c$$

where c does not involve λ , therefore the canonical parameter is

$$\theta = \log(\lambda),$$

and the b -function is

$$b(\theta) = \lambda.$$

The linear predictors is

$$\eta = X\beta,$$

A smooth and invertible linearising link function $g(\cdot)$, which transforms the expectation of the dependent variable, $\lambda_i \equiv E(y_i)$, to the linear predictor:

$$g(\lambda) = \eta = X\beta.$$

Fitting GLM via Fisher scoring

Maximum likelihood estimation for β may be carried out via Fisher scoring,

$$\beta^{(k+1)} = \beta^k + \left[-El''(\beta^{(t)}) \right]^{-1} l'(\beta^{(t)})$$

where l is the loglikelihood function for the entire sample y_1, \dots, y_N . Ignoring the constants, the loglikelihood is

$$l(\theta; y) = \frac{y\theta - b(\theta)}{a(\phi)}$$

The score vector is

$$\frac{\partial l}{\partial \beta_j} = \left(\frac{\partial l}{\partial \theta} \right) \left(\frac{\partial \theta}{\partial \lambda} \right) \left(\frac{\partial \lambda}{\partial \eta} \right) \left(\frac{\partial \eta}{\partial \beta_j} \right) \quad (1.8)$$

The first factor is

$$\frac{\partial l}{\partial \theta} = \frac{y - b'(\theta)}{a(\phi)} = \frac{y - \lambda}{a(\phi)}$$

Because $\lambda = b'(\theta)$, the second factor is

$$\frac{\partial \theta}{\partial \lambda} = \frac{1}{b''(\theta)} = \frac{1}{V(\lambda)} = \frac{a(\phi)}{Var(y)}$$

where $V(y) = b''(\theta)$ is the variance function, the third factor $\partial \lambda / \partial \eta$, will depend on the link function. The fourth factor is $\partial \eta / \partial \beta_j = x_{ij}$, where the x_{ij} is the j th element of covariate vector $x_i = x$ for the i th observation. Putting all together we have

$$\frac{\partial l}{\partial \beta_j} = \frac{y - \lambda}{Var(y)} \left(\frac{\partial \lambda}{\partial \eta} \right) x_{ij}.$$

If we are using the canonical link $\eta = \theta$, then $\partial\lambda/\partial\eta = \partial\lambda/\partial\theta = b'(\theta)$, so the score becomes

$$\frac{\partial l}{\partial\beta_j} = \frac{y - \lambda}{\text{Var}(y)} b''(\theta) x_{ij} = \frac{y - \lambda}{a(\phi)} x_{ij}.$$

To find the expected second derivatives, we can use the property

$$\begin{aligned} -E\left(\frac{\partial^2 l}{\partial\beta_j \partial\beta_k}\right) &= E\left[\left(\frac{\partial l}{\partial\beta_j} \frac{\partial l}{\partial\beta_k}\right)\right] \\ &= E\left(\frac{y - \lambda}{\text{Var}(y)}\right)^2 \left(\frac{\partial\lambda}{\partial\eta}\right)^2 x_{ij} x_{ik} \\ &= \frac{1}{\text{Var}(y)} \left(\frac{\partial\lambda}{\partial\eta}\right)^2 x_{ij} x_{ik} \end{aligned} \tag{1.9}$$

With the canonical link, this becomes

$$E\left(\frac{\partial^2 l}{\partial\beta_j \partial\beta_k}\right) = -\frac{b''(\theta)}{a(\phi)} x_{ij} x_{ik}$$

We have just shown that

$$\begin{aligned} \frac{\partial l}{\partial\beta_j} &= \frac{y - \lambda}{\text{Var}(y)} \left(\frac{\partial\lambda}{\partial\eta}\right) x'_{ij} \\ -E\left(\frac{\partial^2 l}{\partial\beta_j \partial\beta_k}\right) &= \frac{1}{\text{Var}(y)} \left(\frac{\partial\lambda}{\partial\eta}\right)^2 x_{ij} x_{ik} \end{aligned}$$

It can be written as

$$\frac{\partial l}{\partial\beta} = X^T A (y - \lambda)$$

where $X = (x_1, \dots, x_N)^T$,

$$A = \text{Diag} \left[\text{Var}(y) \left(\frac{\partial\eta_i}{\partial\lambda_i}\right) \right]^{-1}$$

The expected Hessian matrix becomes

$$-E\left(\frac{\partial^2 l}{\partial\beta_j \partial\beta_k}\right) = X^T W X$$

where

$$W = \text{Diag} \left[\text{Var}(y_i) \left(\frac{\partial\eta_i}{\partial\lambda_i}\right)^2 \right]^{-1}$$

An iteration of Fisher scoring is then

$$\beta^{(k+1)} = \beta^{(k)} + (X^T W X)^{-1} X^T A(y - \lambda)$$

Chapter 2

Parametric point process modelling for single neuron

2.1 Introduction and example of data

Many scientific investigations involve collecting and observing repeated events over time. Common examples include recurrent episodes of epileptic fits in clinical studies, recurrent episodes of a firing of an action potential in neuron or repeated events of earthquake occurrences. A typical question of interest is how these patterns of occurrences can be described. When the number of events is small or relatively regular for a short period of time, the pattern of events can be summarised as frequency, or rate per unit time, and subsequent analyses are carried out based on these numbers. When the number of events is relatively large and varies a lot over time, it is not easy to summarise the pattern of events in a single number, then, understanding and characterising the sequence of event times poses a serious challenge.

In statistical analysis, the sequence of the event times is referred to as time-to-event data: time-to-event data consist of the times to any number of repeated events for each sample unit. In this chapter, we present a statistical analysis of such type of data representing experimental earthquake occurrence events. The experimental earthquake events have been collected in the lab in Lancaster Environmental Centre and this data set includes the times of earthquakes and magnitudes for a given time period under controlled experimental conditions. The main questions of interest regarding the earthquake occurrence events are; is there any systematic pattern in the sequences of the occurrences?

What type of stochastic models are suitable for explaining structure within recurrent events? We try to answer these questions through searching for suitable statistical models. In this work we consider a parametric class of models.

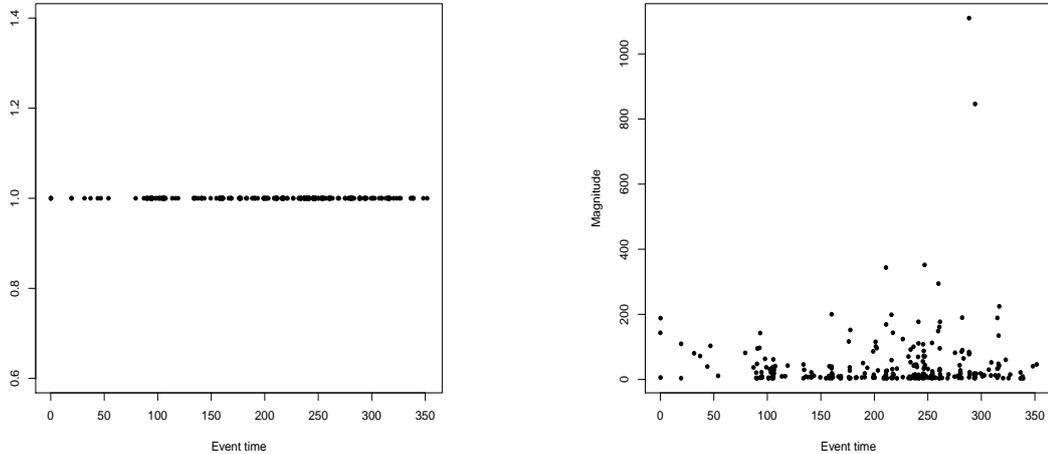


Figure 2.1: The earthquake events (left panel) and the distribution of the stress drop with its magnitude (right panel)

The left hand side of figure 2.1 shows the times of events, the right hand side of figure 2.1 shows the magnitude of the earthquakes during the time interval. The rate of events does not appear to be constant and the magnitude also varies a lot. In the original data set, the first event occurred at time 135.39, then 266 more earthquakes occurred until time 487.11 with varying rate. In order to simplify the analysis the time period has been shifted, from 0 for the first event to 351.7 for the last occurrence.

The occurrence of events could be considered as point process in time. One of the simplest models for recurrent event data is the Poisson process, which assumes that the occurrences in non-overlapping time intervals are independent. The Poisson process is the canonical model for recurrent events and more general models may be formulated as extensions of it. Even if the observed data of occurrences are considered to be statistically independent, the inter-arrival times between events may not be identically distributed.

The independence assumption may be easily violated when the occurrence of the current events depends on its past. Then a history-dependent model is more appropriate. For example, the earthquake events tend to occur in clusters. The most basic illustration of history dependence in earthquake occurrence events is linked to the mechanism of how the aftershock events occur after the previous main earthquakes. To accurately describe

the event occurrence in recurrent data such as earthquakes, we consider the case where the probability of an event occurring at a given time may depend on previous event occurrences.

The rest is organised as follows. In Section 2 we briefly introduce fundamentals of point process modelling. An extension of the Poisson process modelling framework is considered by adding history dependence and a self-exciting process. In section 3, two simulation methods are reviewed. One is the inversion method and another is the thinning method. The results of model fitting to the real data are presented in Section 4, with its goodness-of-fit investigated in Section 5.

2.2 Point process

In the previous chapter, we introduced the temporal point process and the Poisson process which is the one of the commonly used model for point process data. Moreover we defined how to construct the conditional intensity function for the history dependent point process. In this section, we will present two methods, one is the self-exciting process. Intuitively, a process is self-exciting if the occurrence of past points makes the occurrence of future points more probable. Another method is the Epidemic-type aftershock sequence (ETAS) model, which is based on point process modelling of the earthquake events with three assumptions: One is that the occurrence of earthquake events follows the Poisson process, another one is that the number of aftershocks is proportional to $\exp(\alpha M)$ for a magnitude M earthquake, and the finally degree of aftershock follows the Omori law.

2.2.1 Self exciting process

A Poisson process always has a deterministic conditional intensity $\lambda(t)$. If the process is stationary then $\lambda(t)$ is a constant. If we assume positive correlations between present and nearby events, then this is called a self-exciting point process. This means that if an event occurs, another event becomes more likely to occur locally in time and space. This is not true for a Poisson process, since the increments are dependent. We investigate a special class of point processes termed a Hawkes Process [12], which can be represented

by the conditional intensity function:

$$\begin{aligned}\lambda(t | H_t) &= \lambda_0(t) + \int_{-\infty}^t \nu(t-u) dN_u \\ &= \lambda_0(t) + \sum_{t_i < t} \nu(t-t_i)\end{aligned}\tag{2.1}$$

where N is the normal counting process [13]. In Equation (2.2), $\lambda_0(t)$ represents the background rate of events, which is in the simplest model assumed to be constant in time. The second half of the sum describes the self-exciting part of the process. Many choices for the kernel ν have been proposed [12]. We use an exponential kernel

$$\lambda(t | H_t) = \lambda_0 + \sum_{t_i < t} \alpha e^{-\beta(t-t_i)}\tag{2.2}$$

Here, α is a rate of decay constant for the triggering kernel controlling how quickly the overall rate λ returns to its baseline level λ_0 after an event occurs.

ETAS model

Japanese reasercher Omori [19] put forward his law, which shows the inverse power relation between the cumulative number of events and the time after the main shocks, and the exponent is -1 in his formula. Ogata [18] builds up such a model named Epidemic-type Aftershock-sequence (ETAS) model, which combines the Omori law with frequency magnitude relation for history dependent occurence rate of a point process. The general ETAS model is a spatio-temporal model, but I consider only the temporal model. The intensity function assumes that certain earthquake aftershock sequences can be modelled like an epidemic, i.e., large earthquakes including more aftershocks in a given interval of time and also, the aftershock sequence extends for a long time after the main shock. The conditional intensity function is

$$\lambda(t | H_t) = \lambda_0 + A \sum_{t_i < t} e^{\alpha(M_i - M_0)} \left(1 + \frac{t-t_i}{c}\right)^{-p}, t > 0, M_i \geq M_0\tag{2.3}$$

The parameters $(\lambda_0, A, \alpha, c, p)$ are all positive, t_i is the time of the i th event with magnitude M_i , and M_0 is threshold magnitude. The history of the process up to time t includes all event times and magnitudes but not including t . The first term under

the summation determines that larger events raise the intensity more, and the last term describes the lasting effect of the previous events.

2.3 Simulation study

2.3.1 Inversion method

The event times from a Poisson process with history dependence can be generated by the inversion method. The inversion method is based on generating a non-homogeneous Poisson process from the inter arrival times [25]. We can consider the $i+1$ th inter arrival time $X_{i+1} = T_{i+1} - T_i$ conditional on the first i event times $T_1 = t_1, T_2 = t_2, \dots, T_i = t_i$. F_{t_i} is the distribution function of the $(i+1)$ th inter arrival time given times up to i th event times. Then the cumulative density function of X_i conditional on T_1, T_2, \dots, T_i can be derived as

$$\begin{aligned}
F_{t_i}(x) &= Pr\{X_{i+1} \leq x \mid T_j = t_j, j = 1, 2, \dots, i\} \\
&= Pr\{N_{t_i+x} - N_{t_i} \geq 1 \mid T_j = t_j, j = 1, 2, \dots, i\} \\
&= Pr\{N_{t_i+x} - N_{t_i} \geq 1 \mid H_{t_i}\} \\
&= 1 - Pr\{N_{t_i+x} - N_{t_i} = 0 \mid H_{t_i}\} \\
&= 1 - exp\left\{-\int_{t_{i-1}}^{t_i+x} \lambda(u \mid H_{t_i})du + -\int_{t_{i-1}}^{t_i} \lambda(u \mid H_{t_i})du\right\} \\
&= 1 - exp\{-\Lambda(t_i + x) + \Lambda(t_i)\}
\end{aligned} \tag{2.4}$$

where $\Lambda(t_i) = \int_{t_{i-1}}^{t_i} \lambda(u \mid H_{t_i})du$. The cdf in (2.4) is inverted to simulate the $(i+1)$ th event time, given the previous i th event times. The algorithm for inversion method is following;

1. Initialise $t = 0$.
2. Generate $x \sim F_t$ given by (2.4)
3. Set $t \leftarrow t + x$
4. Deliver t
5. Go to step 2

2.3.2 Thinning method

When we are at time t , we need to find out where to place the next point $\tau_i > t$. To do this we simulate a homogeneous Poisson process on some interval $[t, t + l(t)]$ for some chosen function. Here $l(t)$ is the maximum distance we may go forward in time from t and it may be infinite. This Poisson process has a chosen constant intensity on $[t, t + l(t)]$ which fulfills $\lambda^* \geq \sup_{s \in [t, t + l(t)]} \lambda(s | H_s)$. Let us denote $U_{[0,1]}$ the uniform distribution on the interval $[0, 1]$ and $[t, t + l(t)]$ the time interval on which the process is to be simulated [10].

Algorithm:

1. Initialisation : Set $\lambda^* = \lambda_0(0)$, $n = 1$.
2. First event : Generate $U \sim Unif(0, 1)$ and set $s_1 = -\frac{1}{\lambda^*} \ln U$.
 - If $s \leq t + l(t)$, then set $t_1 = s_1$
 - Else go to step 4
3. General routine : Set $n = n + 1$
 - (a) Update maximum intensity : Set $\lambda^* = \sup_{s_n \in [t, t + l(t)]} \lambda(s_n | H_s)$.
 - (b) New event : Generate $U_n \sim Unif(0, 1)$ and set $s_n = s_{n-1} - \frac{1}{\lambda^*} \ln(U_n)$.
 - If $s_n \geq t + l(t)$, then go to step 4
 - (c) Rejection Test : Generate $D_n \sim Unif(0, 1)$
 - If $D_n \leq \frac{\lambda(s_n)}{\lambda^*}$, then $t_n = s_n$ and go through the general routine again.
4. Output : Retrieve the simulated process $\{t_n\}$ on $[t, t + l(t)]$

Proof :

Consider the problem of generating samples of a counting process $N(t) = \sum_{i \geq 1} \mathbf{1}_{\{t_i \leq t\}}$ with intensity $\lambda(s)$. This amounts to sampling jump times t_i , from the knowledge of $\lambda(s)$.

The thinning method requires:

- To select λ^* as an upper bound for $\lambda(s)$ on its domain.
- Perform an acceptance-rejection test on each t_i which consists of sampling random variable $\mathbf{1}_{\{\lambda^* U_i \leq \lambda(t_i)\}}$.

- Accept jump time of N provided that the test succeeds

To prove this, we need to check that the counting process of the accepted jump times $N(t) = \sum_{i:s < t_i \leq t} \mathbf{1}_{\{\lambda^* U_i \leq \lambda(t_i)\}}$ has intensity λ that is

$$E(N(t) - N(s) | F_s^N) = E\left(\int_s^t \lambda(u) du | F_s^N\right)$$

where F_s^N is the completed filtration generated by N . It can be interpreted that the expected number of jump time occurring in an interval $(s, t]$ is given by occruing the jump intensity λ over the same interval. We assume that the function λ is superiorly bounded and set $\lambda^* = \sup(\lambda(t))$. Using the linearity property and the rule of iterated conditional expectation, we have

$$\begin{aligned} E(N(t) - N(s) | F_s^N) &= E\left(\sum_{i:s < t_i \leq t} \mathbf{1}_{\{\lambda U_i \leq \lambda(t_i)\}} | F_s^N\right) & (2.5) \\ &= E\left(\sum_{i:s < t_i \leq t} [E(\mathbf{1}_{\{\lambda U_i \leq \lambda(t_i)\}} | F_{t_i}^N) | F_s^N]\right), F_s^N \subset F_{t_i}^N \\ &= E\left(\sum_{i:s < t_i \leq t} [P(U_i \leq \lambda(t_i)/\lambda^* | F_{t_i}^N) | F_s^N]\right) \\ &= \frac{1}{\lambda^*} E\left(\sum_{i=N(s)+1}^{N(t)} \lambda(t_i) | F_s^N\right) \\ &= \frac{1}{\lambda^*} E\left(E\left(\sum_{i=N(s)+1}^{N(t)} \lambda(t_i) | F_t^N\right) | F_s^N\right) \end{aligned}$$

Recall that the jump times are uniformly distributed on the interval $(s, t]$ conditional on the number of jumps having occurred in the same interval. The Wald theorem leads to :

$$E(N(t) - N(s) | F_s^N) = \frac{1}{\lambda^*} E(E[N(t) - N(s) | F_t^N] \times E[\lambda(t_i) | F_t^N] | F_s^N)$$

Given F_s^N , the increment $N(t) - N(s)$ and the random variable t_i are mutually independent. Moreover, t_i is uniformly distributed in $[s, t]$ given F_t^N . The last term then becomes equals to:

$$\frac{1}{\lambda^*} \int_s^t \lambda^* du E\left(\lambda(u) \frac{du}{t-s} | F_s^N\right) = E\left(\int_s^t \lambda(u) du | F_s^N\right)$$

Parameters									
$\lambda_0 = 0.05$									
α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}
0.917	6.766	3.353	3.037	4.970	1.955	5.590	1.524	1.623	4.507
β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8	β_9	β_{10}
3.476	5.857	3.104	2.900	0.681	1.368	4.668	1.824	1.494	4.783

Table 2.1: Estimated parameters for conditional intensity function and the negative Log-likelihood value

2.3.3 Analysing the simulated dataset

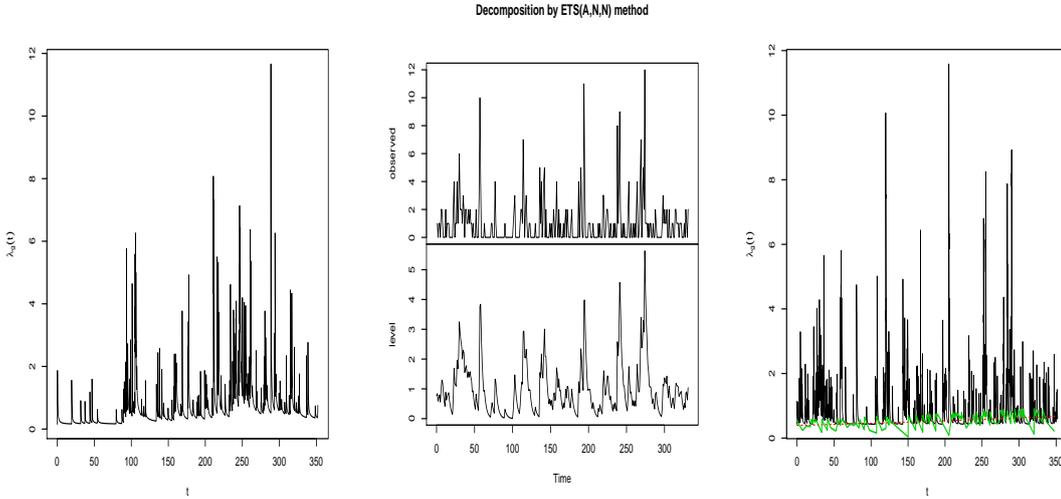


Figure 2.2: Intensity for the simulated data: true intensity (left panel), non-parametric estimation (middle panel) and conditional intensity (Model 1, 2 and 3)

We simulated a data set by using thinning algorithm, the true intensity function is shown at the left hand side of figure 2.3.3. It describes the instantaneous occurring probability of simulated experimental earthquakes at time t . The figure in the middle of figure 2.3.3 describes $\lambda(t)$ from the non-parametric estimation with the top image showing the $\lambda(t)$ from the exponential smoothing. Finally the right side hands of figure 2.3.3 describes the conditional intensity function from three different methods which we have fitted. At first we fit the data to the non-homogeneous Poisson process (Model 1), then fit the data to the point process which is an extension of the Poisson process by adding history dependence (Model 2) and finally model 3 is fitting data to the ETAS model. The black solid line indicates conditional intensity function from the Model 3, the green dotted line shows the CIF from the Model 2 and the red dotted line shows the

Model 1					
$\lambda(t) = \exp\{\beta_0 + \beta_1 t\}$					
estimates	β_0 -0.8873 (0.1230)		β_1 0.0031(0.0006)		
$-l(\theta)$	327.0508				
Model 2					
$\lambda(t H_t) = \exp\{\beta_0 + \beta_1 t + \beta_2(t - t_{N(t^-)})\}$					
estimates	β_0 0.1071 (0.1714)	β_1 0.0013 (0.0007)	β_2 -0.3184 (0.0416)		
$-l(\theta)$	275.7227				
Model 3					
$\lambda(t H_t) = \lambda_0 + A \sum_{t_i < t} e^{\alpha(M_i - M_0)} \left(1 + \frac{t - t_i}{c}\right)^{-p}$					
estimates	λ_0 0.15279	A 2.2708	α 0.1386	c 0.0752	p 1.15301
$-l(\theta)$	246.0985				

Table 2.2: Estimated parameters for conditional intensity function and the - log-likelihood value

Model 1. According to figure ??, Model 1 and 2 do not seem to explain data compared to the non-parametric estimation, but Model 3 seems to give a similar intensity function to the true intensity function and non-parametric estimation.

2.4 Application to data

We analyse the experimental earthquake occurrence data set with three different processes. The estimations for each point process are shown in table 2.2. The left hand side of figure 2.3 shows the conditional intensity function from the nonparametric estimation with the top of figure shows the conditional intensity function from the exponential smoothing. The right side of figure 2.3 shows the conditional intensity functions from the three different models. The red and green dotted lines indicate the CIF from the model 1 and model 2 respectively and the black solid line demonstrates the CIF from the model 3.

First of all, we fit the model 1 with the time varied rate to the intensity. Then add the history effect, which is since the last event, to the model 1. Table 2.2 summarises the estimates of these models. The negative log-likelihood values decrease a lot compared to the previous model which did not consider the history dependence on the intensity function. According to likelihood ratio test, a test of the model with history dependence versus the model without it gives a likelihood ratio statistics of $2(-275.7227 + 327.0508)$

$= 51.3281$. Using χ_1^2 , this gives a p -value under 0.05, hence it can be said that the latter model gives the better estimation. However, when we compare the conditional intensity function from model 1 and model 2 to the conditional intensity function from the nonparametric estimation, these two models do not explain the data. Model 3, which add the magnitude effect and all histories, seems to have similar trend with conditional intensity function from the nonparametric estimation. Also the -log-likelihood estimation decreases. Hence the model 3 gives the best estimation, but we cannot say that the model 3 fits well because there is no evidence to say that.

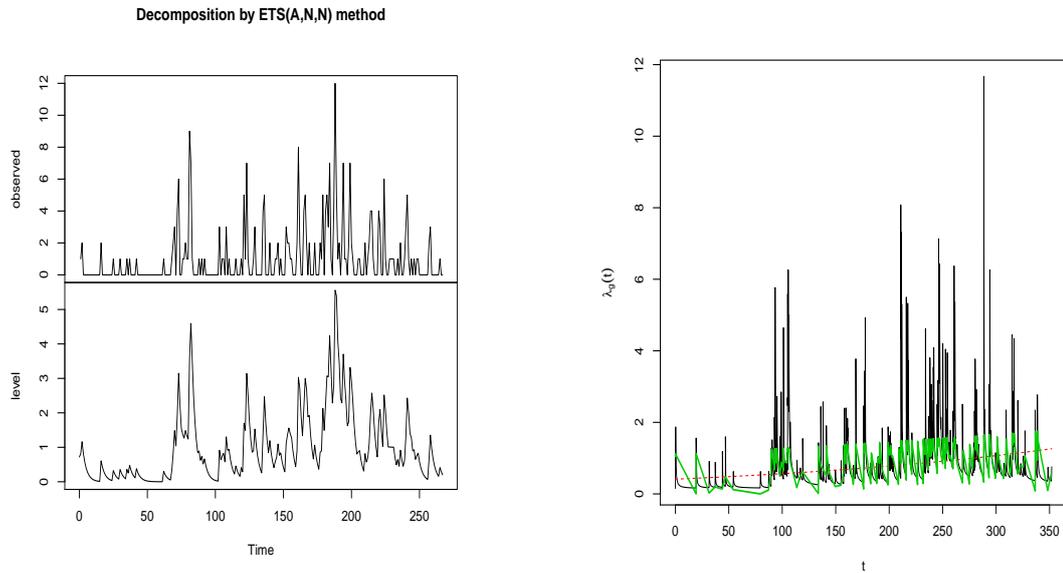


Figure 2.3: Intensity for the experimental earthquake events: non-parametric estimation (left panel) and conditional intensity (Model 1, 2 and 3)

2.5 Model checking ; Time-rescaling theorem

One essential component of any statistical modelling analysis is to check that the model accurately describes the structure observed in the data. One way of doing this is to apply the time-rescaling theorem to transform point processes data into continuous measures and then assess goodness-of-fit. Let $0 < t_1 < t_2 < \dots, t_n < T$ be a realisation from a point process with conditional intensity function $\lambda(t | H_t)$ satisfying $0 < \lambda(t | H_t)$ for all

$t \in (0, T]$. Define

$$\begin{aligned}\Lambda(t_1) &= \int_0^{t_1} \lambda(u | H_t) du, \quad \text{and} \\ \Lambda(t_i) &= \int_{t_{i-1}}^{t_i} \lambda(u | H_t) du,\end{aligned}\tag{2.6}$$

Then these $\Lambda(t_i)$ are independent, exponential random variables with rate parameter 1.

This result is called the time-rescaling theorem, since we can think of the transformation as stretching and shrinking the time axis based on the value of the conditional intensity function. If $\lambda(t | H_t)$ is constant and equal to 1 everywhere, then this is a simple Poisson process with independence and time does not need to be rescaled. Any time when $\lambda(t | H_t)$ is less than 1, the $\Lambda(t_i)$ values accumulate slowly and they represent a shrinking of time, so that distant earthquake occurrence times are brought closer together. Likewise, any time when $\lambda(t | H_t)$ is greater than 1, the $\Lambda(t_i)$ value accumulate more rapidly and they represent a stretching of time, so that nearby events are drawn further apart. Because the transformation in Equation (??) is one-to-one, any statistical assessment that measures the agreement between the $\Lambda(t_i)$ values and an exponential distribution directly evaluates how well the original model agrees with the earthquake occurrence data. If we make the further transformation

$$u_i = \Lambda(t_i | H_t), \quad \text{for } i = 1, \dots, n$$

Then $\tau_i = u_i - u_{i-1}$

$$z_i = 1 - \exp(-\tau_i)$$

then z_i 's are independent uniform random variables on the interval $(0, 1]$.

A Kolmogorov-Smirnov(KS) plot is a plot of the empirical cumulative distribution function (CDF) of the rescaled $\Lambda(t_i)$ against an exponential CDF, or the rescaled z_i against the cumulative distribution of the uniform density defined as $b_i = \frac{i-\frac{1}{2}}{n}$ for $i = 1, \dots, n$ against the ordered values $z_{(i)}$ s. If the conditional intensity model accurately describes the observed earthquake occurrence events, the KS plot should follow a 45° line. Confidence intervals for the degree of agreement between the models and the

data might be constructed by using the distribution of the Kolmogorov-Smirnov statistics. The 95% of confidence interval can be constructed as $b_i \pm 1.36/n^{1/2}$ [3]

Another approach to checking goodness-fit between the model and data is to construct a quantile-quantile plots, which plot the quantile of the rescaled time interval against the uniform distribution. To construct confidence intervals, we note that if the τ_i are independent exponential random variables with mean 1 and the z'_i s are uniform on the interval $(0, 1)$, then each $z_{(i)}$ has a beta probability density [3] with parameter i and $n - i + 1$ defined as

$$f(z | i, n - i + 1) = \frac{n!}{(n - i)!(i - 1)!} z^{i-1} (1 - z)^{n-i} \quad (2.7)$$

for $0 < z < 1$. Then the approximation of a 95% confidence interval is given by the Gaussian approximation to the binomial probability distribution as $z_i \pm 1.96[z_i(1 - z_i)/n]^{1/2}$ Likewise KS plot, exact agreement occurs between the point process and the exponential data if the points lie on the 45° line.

2.5.1 KS plots and Q-Q plots of a experimental earthquake data

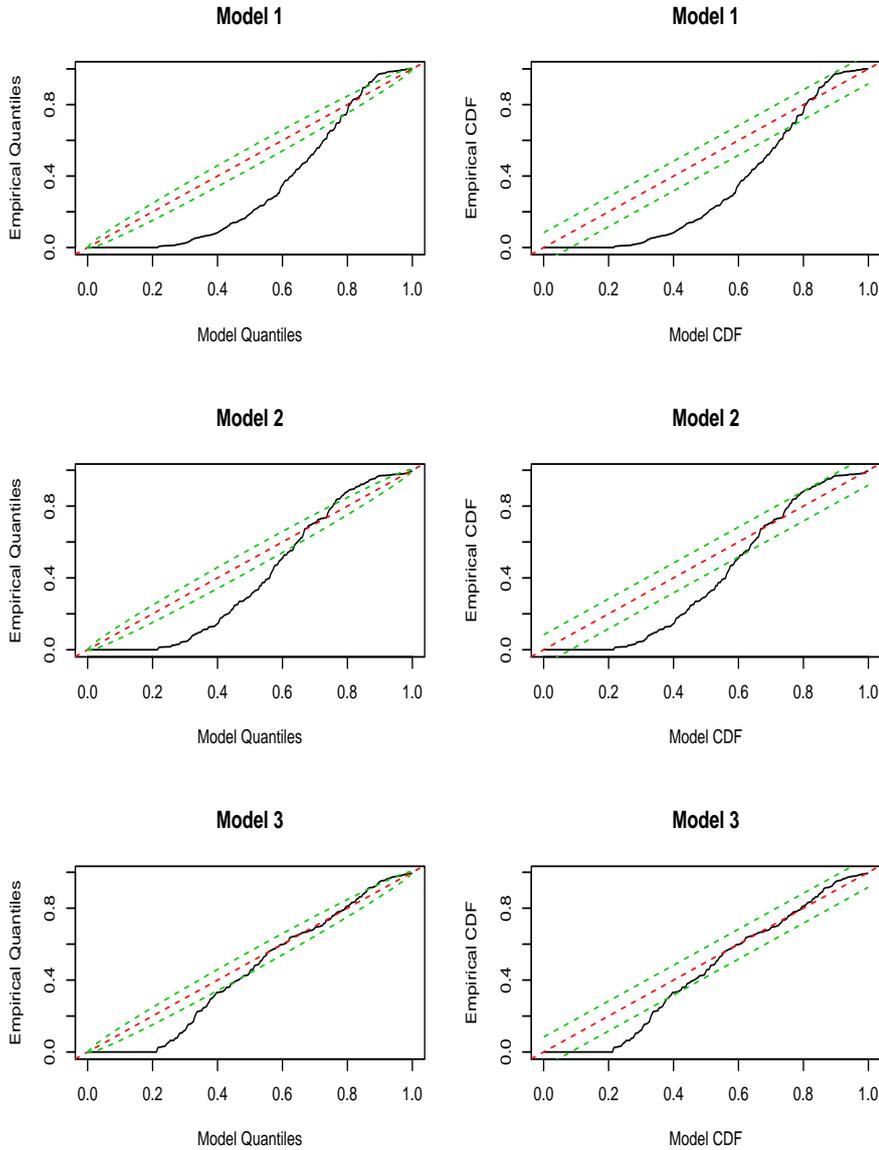


Figure 2.4: Q-Q plot and KS plot for distribution of rescaled intervals for the model 1, 2, and 3.

KS plot and QQ plot for the transformed earthquake occurrence data can be constructed. Three QQ plots for the models 1, 2 and 3 are shown in the left hand side of Figure 2.4 and KS plots for the three different models are shown in the right hand side of Figure 2.4. First of all, by examining the QQ plot and KS plot for the model 1 in Figure 2.4, we see that the model does not seem to explain the small rescaled time interval and midrange time interval. Only some parts of the large time interval were lying on the confidence interval. This suggests that this nonhomogeneous Poisson model

for the earthquake occurrence events is unable to completely describe the structure in the data. According to QQ plot and KS plot for the model 2, it also suggests that the extended model from the nonhomogeneous model by adding history dependence does not fit to this dataset. Since the results from the QQ plot and KS plot do not lie on the confidence interval. Finally, the results from QQ plot and KS plot for the self-exciting process is showing a significant improvement compared to the model 1 and model 2. It still does not describe the small rescaled time interval but the rest of part seems to lie in the confidence interval. It cannot be said that the self-exciting process completely describes the earthquake occurrence events, but it provides a much closer fit to the data.

2.6 Discussion and conclusion

Point process methods are essential for understanding how earthquake events represent information about history dependence and magnitude. In particular, point process theory provides approaches to visualising earthquake occurrence data, constructing, fitting models and assessing the ability of these models to explain structure in the data.

The most important of these methods is the conditional intensity function, which provides information for analysing point process data. The conditional intensity function practically determines the instantaneous occurring probability at time t as a function of history dependence and external covariates. In the parametric model, it can be constructed by writing down the conditional intensity function in terms of covariates and a set of parameters. According to the definition of the conditional intensity function, it is easy to find that the likelihood function of any point process model has a standard form. The model parameters that best fit the data can be obtained by maximum likelihood method.

The goodness-of-fit can be assessed by the time-rescaling theorem. Determining goodness-of-fit is an important step, since this step is fundamental for establishing whether the model is appropriate for making statistical inference and establishing how reliable those inferences are.

Chapter 3

Non-parametric point process modelling for single neuron

3.1 Introduction

Characterising the neuron spike train firing as a function of external stimulus applied in the experiment and intrinsic dynamics of neurons such as absolute and relative refractory periods, the history effect is important in neuroscience. Such a characterisation is very complex and the broad class of models to capture such details are keep required. One of the useful methods for characterising neuron spike trains activity is a point process model. For instance, they have successfully characterised spiking activity of rat hippocampal place cells[9] and sea hare nerve cells[1].

There are many parametric point process models based on likelihood analysis. ([4]; [1]; [2]; [29]). They assumed that the conditional intensity function is belonging to a class of parametric functions. Parametric models have several advantages.

- Efficiently computable
- The parameters may be related to environmental factor
- Asymptotic property- if the actual distribution lies in the assumed parameter class.

However, if the true conditional intensity function does not lie in the assumed class, it may cause large error of estimation and mislead the inference.

On the other hand, non-parametric methods for estimating conditional intensity function of a point process model are difficult to compute. However there are several advantages

of non-parametric estimation methods. First, non-parameteric methods do not rely on data belonging to any particular distribution, for which reason they are also called distribution free methods. Recently, Truccolo and Donoghue introduced a computationally efficient non-parametric approach to model neural point processes via stochastic gradient boosting regression[15].

We propose a non-parametric point process which is computationally efficient for the intrinsic dynamic model. We formulate the log of the conditional intensity function in terms of a discrete time Volterra expansion of the baseline spike rate and the neuron’s spiking history, and use a generalised linear model in a regression spline framework to address the maximum likelihood estimation problem. We illustrate our approach by fitting the model to the 25 input and 18 output neuron spike trains. The residual analysis is performed by the time-rescaling theorem[3].

3.2 Methodology

3.2.1 Point process model

- The conditional intensity function is given by:

$$\lambda(t | H(t)) = \lim_{\Delta t \rightarrow 0} \frac{Pr\{\Delta N_{(t,t+\Delta t)}\}}{\Delta t},$$

- The inter spike interval density at time t_i is given by derivation of CDF, $f_{T_i}(t_i | T_{i-1} = t_{i-1}) = \frac{d}{dt_i}[1 - \exp\{-\int_{t_{i-1}}^{t_i} \lambda(u)du\}]$ which implies,

$$f_{T_i}(t_i | T_{i-1} = t_{i-1}) = \lambda(t_i) \exp\left\{-\int_{t_{i-1}}^{t_i} \lambda(u)du\right\}$$

- The likelihood of a point process on $[0, \tau]$ is given by:

$$\prod_{i=1}^n \lambda(t_i) \cdot \exp\left\{-\int_0^\tau \lambda(u)du\right\}.$$

- The generalised linear model has recently been used to model conditional intensity functions of point processes. It requires discretised time in bins with no more than

one event in each bin. Then the likelihood function is given by:

$$Pr(\Delta N(t_1), \Delta N(t_2), \dots, \Delta N(t_n)) = \prod_{i=1}^n Pr(\Delta N(t_i) | \Delta N(t_1), \dots, \Delta N(t_{i-1}))$$

The conditional Bernoulli probability follows:

$$\begin{aligned} Pr(\Delta N(t_i) | \Delta N(t_1), \dots, \Delta N(t_{i-1})) &= (\lambda_i \Delta t)^{\Delta N(t_i)} (1 - \lambda_i \Delta t)^{1 - \Delta N(t_i)} \\ &= \exp \left[\Delta N(t_i) \log \frac{\lambda_i \Delta t}{1 - \lambda_i \Delta t} + \log \{1 - \lambda_i \Delta t\} \right] \end{aligned} \quad (3.1)$$

where $\lambda_i \equiv \lambda$. The mean of the Bernoulli probability at i-th bin is $\mu = E(\Delta N(t_i)) = \lambda_i \Delta t$. Approximation by Poisson follows:

$$\begin{aligned} Pr(\Delta N(t_i) | \Delta N(t_1), \dots, \Delta N(t_{i-1})) &= \left(\frac{\lambda_i \Delta t}{1 - \lambda_i \Delta t} \right)^{\Delta N(t_i)} (1 - \lambda_i \Delta t) \\ &\approx [\Delta N(t_i) \log(\lambda_i \Delta t) - \lambda_i \Delta t] \end{aligned} \quad (3.2)$$

The mean of the Poisson probability at i-th bin is $\mu = E(\Delta N(t_i)) = \lambda_i \Delta t$.

The spike count in a small bin is then approximated as:

$$Pr(\Delta N = n) = \frac{(\lambda \Delta t)^n}{n!} e^{-\lambda \Delta t} = \frac{(\lambda \Delta t)^n}{n!} \left[1 - \lambda \Delta t + \frac{1}{2} (\lambda \Delta t)^2 + \dots \right]$$

In particular,

$$\begin{aligned} Pr(\Delta N = 0) &= 1 \left[1 - \lambda \Delta t + \frac{1}{2} (\lambda \Delta t)^2 + \dots \right] = 1 - \lambda \Delta t + o(\Delta t) \\ Pr(\Delta N = 1) &= \lambda \Delta t \left[1 - \lambda \Delta t + \frac{1}{2} (\lambda \Delta t)^2 + \dots \right] = \lambda \Delta t + o(\Delta t) \\ Pr(\Delta N = 2) &= (\lambda \Delta t)^2 \left[1 - \lambda \Delta t + \frac{1}{2} (\lambda \Delta t)^2 + \dots \right] = o(\Delta t) \end{aligned} \quad (3.3)$$

The probability of having a spike train or no spike train is an approximation of the Poisson count distribution for a small time bin. The likelihood of discrete-time

conditional Poisson distribution is given:

$$\begin{aligned}
Pr(\Delta N(t_n) \mid \Delta N(t_1), \dots, \Delta N(t_{(n-1)})) &= \prod_{i=1}^n \exp[\Delta N(t_i) \log\{\lambda_i \Delta t\} - \lambda_i \Delta t] + o(\Delta t'') \\
&= \Delta^n \exp \left[\sum_{i=1}^n \Delta N(t_i) \log \lambda_i - \lambda_i \Delta t \right] + o(\Delta t'')
\end{aligned} \tag{3.4}$$

The likelihood of a point process by continuous time limit is given:

$$\begin{aligned}
p(t_1, t_2, \dots, t_n \cap N(T) = n) &= \lim_{\Delta t \rightarrow 0} \frac{Pr(\Delta N(t_i) \mid \Delta N(t_1), \dots, \Delta N(t_{(i-1)}))}{\Delta t^n} \\
&= \lim_{\Delta t \rightarrow 0} \frac{\exp[\sum_{i=1}^n \Delta N(t_i) \log \lambda_i - \sum_{i=1}^n \lambda_i \Delta t] + o(\Delta t'')}{\Delta t^n} \\
&= \exp \left[\int_0^T \log \lambda_u dN(t_u) - \int_0^T \lambda_u du \right]
\end{aligned} \tag{3.5}$$

The probability of having a spike or no spike is an approximation of the Poisson count distribution for a small bin. It can be represented parametrically as follows:

$$\log(\lambda(t)) = \beta_0 + \sum \beta_j \phi_j(t),$$

- History dependence - The conditional intensity function is given by:

$$\lambda(t \mid H_t) = \lim_{\Delta t \rightarrow 0} \frac{Pr\{\Delta N(t) = 1 \mid H_t\}}{\Delta t},$$

,where H_t is the history of the event occurrence up to time t as well as possibly extrinsic covariates. When the value $N(0)$ is included in H_t , typically H_t should be treated as 0. The likelihood function is :

$$p(t_1, t_2, \dots, t_n \cap N(T) = n) = \prod_{i=1}^n \lambda(t_i \mid H(t_i)) \cdot \exp \left\{ - \int_0^T \lambda(u \mid H(u_i)) du \right\}$$

Modeling the conditional mean rate via a canonical link function is given:

$$E \left[\frac{dN(t)}{dt} \right] = \lambda(t \mid H_t) = \exp(\beta_0 + \sum \beta_j \phi_j(t) + \gamma g(H_t)).$$

- The Time-Rescaling Theorem allows for a way to transform an arbitrary point process with strict positive conditional intensity function to a unit rate Poisson process. Let $0 < t_1 < t_2 < \dots, t_n < T$ be a realisation from a point process with conditional intensity function $\lambda(t | H_t)$ satisfying $0 < \lambda(t | H_t)$ for all $t \in (0, T]$.

Define

$$\begin{aligned}\Lambda(t_1) &= \int_0^{t_1} \lambda(u | H_t) du, \quad \text{and} \\ \Lambda(t_i) &= \int_{t_{i-1}}^{t_i} \lambda(u | H_t) du,\end{aligned}\tag{3.6}$$

Then these $\Lambda(t_i)$ are independent, exponential random variables with rate parameter 1.

3.2.2 Nonparametric modeling of intensity functions of point process

Using neuron data set, which discretised time into bin, we found generalised linear model can be a point process and it can be expressed as

$$\log(\lambda(t)) = \sum_j \beta_j \phi_j(t)$$

where $\phi_j(t)$ is a sequence of unknown basis functions. In this section, we will describe methods involving spline as well as piecewise polynomial methods.

The discovery that piecewise polynomials or splines could be used in place of polynomials occurred in the early twentieth century. Splines have since become one of the most popular ways of approximating non-linear functions.

Bspline basis function

B-splines constitute an appealing method for the non-parametric estimation of a range of statistical objects of interest. A spline is a function that is constructed piece-wise from a polynomial function. A B-spline function is a differentiable interpolative basis function and it is defined by order m and number of interior knots. The degree of the B-spline polynomial is $m - 1$.

Let $\boldsymbol{\tau} = \{\tau_i \mid i \in Z\}$ be a sequence of non-decreasing real numbers ($\tau_i \leq \tau_{i+1}$) such that

$$\tau_0 \leq \tau_1 \leq \cdots \leq \tau_{k+1}$$

Define the boundary knots τ_1, \dots, τ_M such that

$$\tau_1 \leq \tau_2 \leq \tau_3 \leq \cdots \leq \tau_M \leq \tau_0,$$

$\tau_{j+M} = \xi_j$ for $j = 1, \dots, k$ and

$$t_{k+1} \leq \tau_{k+M+1} \leq \cdots \leq \tau_{k+2M}.$$

The extra knots are usually chosen $\tau_1 = \cdots = \tau_M = \tau_0$ and $t_{k+1} = \tau_{k+M+1} = \cdots = \tau_{k+2M}$. The basis functions are defined as;

$$B_{i,1} = \left\{ \begin{array}{ll} 1 & \text{if } \tau_i \leq x \leq \tau_{i+1} \\ 0 & \text{otherwise} \end{array} \right\}$$

for $i = 1, \dots, k + 2M - 1$. Next, for $m \leq M$ we define

$$B_{i,m} = \frac{x - \tau_i}{\tau_{i+m-1} - \tau_i} B_{i,m-1} + \frac{\tau_{i+m} - x}{\tau_{i+m} - \tau_{i+1}} B_{i+1,m-1}$$

for $i = 1, \dots, k + 2M - m$.

Cubic spline The functions $\{B_{i,4}, i = 1, \dots, k + 4\}$ form a basis for the set of cubic splines. Therefore any spline $\phi(x)$ can be written as $\phi(x) = \sum_{j=1}^{k+4} \beta_j B_j(x)$.

B-spline basis function have compact support which makes it possible to speed up calculations. Figure 3.1 shows the cubic B-spline basis using eight equally spaced knots on $(0, 5000)$.

According to B-spline basis function ϕ is a cubic spline. Therefore we can write

$$\phi(x) = \sum_{j=1}^N \beta_j B_j(x)$$

where $N = n + 4$. We only need to find the coefficients $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \dots, \hat{\beta}_N)^T$.

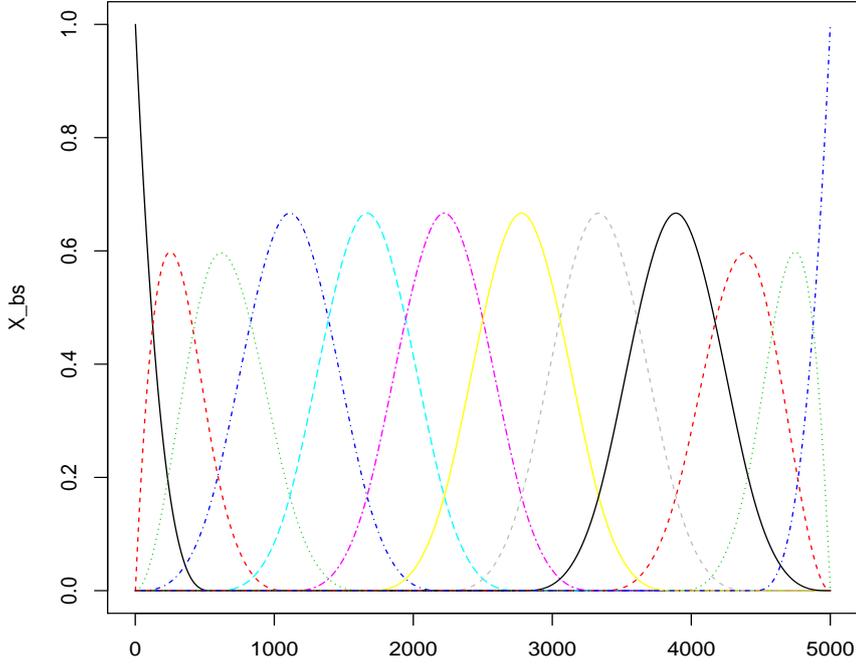


Figure 3.1: B-spline basis function

3.2.3 History effects

In the non-parametric model with B-spline basis function the function of form was $\beta_0 + \beta_j\phi(t)$. Now we will describe an expansion of the class of generalised linear models which is called generalised additive model. They extend the generalised linear model in the same manner that the additive model extends the linear regression model by replacing the linear form $\beta_0 + \beta_j\phi(t)$ with the additive form $\gamma g(t)$. The function $g(t)$ explain the history dependent effects which can explains the bursty behaviour of neuron spike train firing. We investigate a special class of point processes termed a Hawkes Process [12], which can be represented by the conditional intensity function:

$$\begin{aligned}
 \lambda(t | H_t) &= \lambda_0(t) + \int_{-\infty}^t \nu(t-u) dN_u \\
 &= \lambda_0(t) + \sum_{t_i < t} \nu(t-t_i)
 \end{aligned}
 \tag{3.7}$$

where N_u is the normal counting process [13]. In Equation (3.7), $\lambda_0(t)$ represents the background rate of events. The second half of the sum describes the self-exciting part of the process. Many choices for the kernel ν have been proposed [12]. We use an exponential kernel $\nu(t) = \alpha e^{-\gamma t}$

$$\lambda(t | H_t) = \exp \left\{ \sum_{j=1} \beta_j \phi(t) + \sum_{t_i < t} \alpha e^{-\gamma(t-t_i)} \right\}$$

Here, α is a rate of decay constant for the triggering kernel controlling how quickly the overall rate $\lambda(t)$ returns to its baseline level $\sum_{j=1} \beta_j \phi(t)$ after an event occurs.

3.2.4 Estimation inference

The first model which does not consider history effects on the single neuron spike train is a regression spline based on Bspline basis function

$$\log \lambda(t) = \sum_{j=1}^N \beta_j \phi(t).$$

We can rewrite in a more concise form as

$$\log \lambda(t) = \mathbf{X} \beta$$

where \mathbf{X} is $(n \times k)$ matrix of covariates. It follows that the log likelihood function for estimating β is

$$l(\beta) = \sum_{i=1}^n N(t_i) \log(\lambda_i \Delta) - \lambda_i \Delta \quad (3.8)$$

An advantage of this concise form is that it shows that 3.8 is equivalent to a generalised linear model with a Poisson log likelihood function. We can, therefore, use the Fisher scoring algorithm to estimate β . It is common that iterative re-weighted least-square is used to solve Fisher scoring algorithm for GLM parameter estimation with canonical link function. The implementation is provided in the Appendix B.

3.3 Algorithm

3.3.1 Iterative re-weighted least squares for non-linear model

Given n log of intensity functions $f = (f_1, \dots, f_n)$ of k variables $\beta = (\beta_1, \dots, \beta_k)$ with $n \leq k$, the Gauss-Newton algorithm iteratively finds the minimum of the sum of squares

$$\log(\lambda_i) = f(x_i, \beta)$$

we show the iterative re-weighted least square algorithm for the GLM in the previous section, for the non-linear model the last part of 1.8 will change. In the GLM $\partial\eta/\partial\beta$ was x_{ij} , however in the non-linear model $\partial\eta/\partial\beta = \partial f/\partial\beta$. For example, let the η is a Hawkes process,

$$\log\lambda = \lambda_0(t) + \alpha \sum_{t_i < t} \exp(-\gamma(t_i - t))$$

where λ_0 is a Bspline basis function with coefficients $\tilde{\beta}_1, \dots, \tilde{\beta}_{k-1}$. An iteration of Fisher scoring is then

$$\beta^{(k+1)} = (J_f^T W J_f)^{-1} J_f^T W z,$$

where the entries of the Jacobian matrix are

$$(J_f) = \frac{\partial f}{\partial \beta_j}$$

- $\log\lambda_i = \sum_{k=1}^K \phi_k(x_i)\tilde{\beta}_k$ - where $\phi_k(x_i)$ is a Bspline basis function, it is a $n \times K$ design matrix and it has a linear combination with K vectors β_k . For a Jacobian matrix J for this the same as x_{ik} , since it has a linear combination with β .
- $\log\lambda_i = f(x_i, \beta)$ - the function $f(x_i, \beta)$ is a Hawkes process $\lambda_0(t) + \alpha \sum_{t_i < t} \exp(-\gamma(t - t_i))$. Here $\lambda_0(t)$ is a Bspline basis function which is the same as the above model and the last term explains the historical dependence. The Jacobian matrix J for the first term is the x_{ij} and J for a historical term is $\alpha \sum_{t_i < t} \exp(-\gamma(t - t_i))$
- $\log\lambda_i = \log\{\lambda_0(t) + \alpha \sum_{t_i < t} \exp(-\gamma(t - t_i))\}$

3.4 Application to neuron spike train data

3.4.1 Experimental procedures

The dataset contains 25 input of neuron spike trains and 18 output of neuron spike trains. Figure A.2 in Appendix shows the spiking activities of all such neuron firing spontaneously over a period of 30 seconds.

3.4.2 Application to data

The dataset used in this study is a 3030 s continuous recording of 43 neurons. The main aim of this study is to find out the spiking activity on each neuron. The knot sequence of the B-spline basis is determined by Bayesian information criterion.

Figure A.3 illustrates kernel functions estimated for the 25 neuron spike train input data using B-spline. Some input neuron spike trains seem to have similar pattern. The activity of the first three inputs seem to have lots of signal time between 0 to 500 and sending a signal after 2500. The signal of input for the inputs 4, 5, 6 and 7 is different to other patterns. The input signal for rest of them peak on time between 500 and 1000 and around 2500. We will find out the similar patterns using the clustering method.

Figure A.7 shows the Kolmogorov-Smirnov plot of the intensity function for the 25 input neurons. For some of the ks plots, does not lie on the confidence interval, therefore some of estimation of intensity functions are incorrect. One possible reason of this is that we set the knots for the Bspline basis function, but many neurons sent the signal in short period. In the input 1 neuron, for example, most of signal were sent at times between 0 and 500 hence the inter-arrival times were very short in that period. However other periods have large inter-arrival time. Another possible reason is that neuron spike train will affect the refractory period. We will deal with two possible reasons of this problems.

Figure A.4 shows kernel estimated for the 18 output neuron spike train using Bspline basis function. The first six neurons have a similar pattern which have high intensity time between 0 and 500 and time between 2500 and 3000. The intensity of the rest of them peak at time 1000 and increasing trend time after 2500. By visualizing, it can be shown that there are two different patterns of intensity of output neuron spike train data. The Kolmogorov-Smirnov plot A.9 for the kernel function estimated for the outputs shows that some of the ks lines does not lie in the confidence interval. However most of the KS

lines lie adjacent confidence interval.

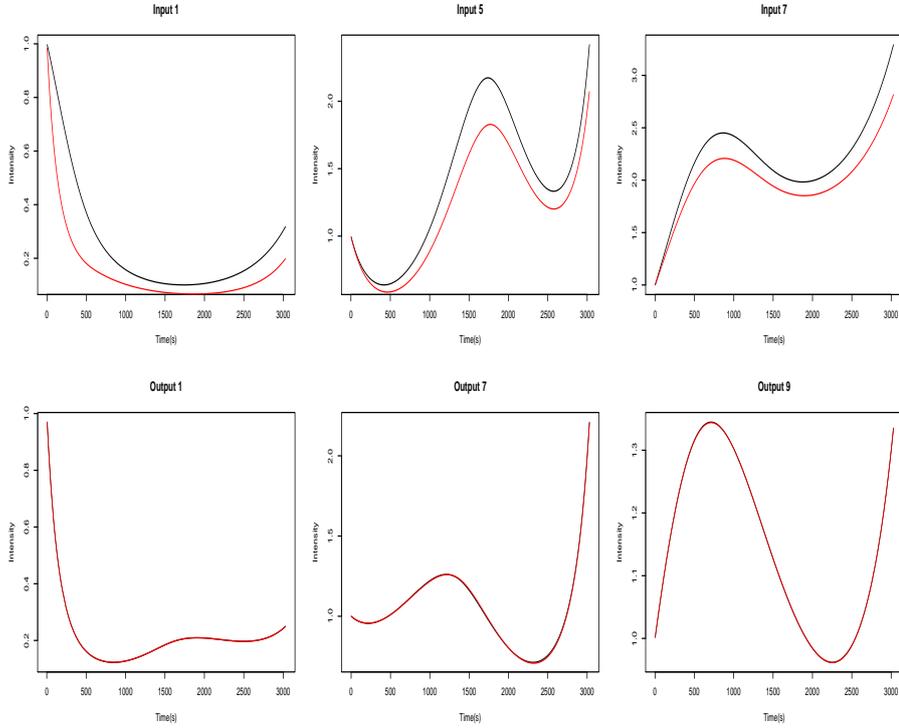


Figure 3.2: Kernel estimation using B-spline basis function and Kolmogorov-Smirnov plot, (black line-original data, red line-modified data)

We chose three kernel of which residual analysis are the bad, moderate and good from input and output data set, selected neurons are input 1, 5 and 7 and output 1, 7 and 9 respectively. Figure 3.3 shows the different estimation between the original dataset and a version in which we deleted events with inter-arrival times of less than 30m sec. The reason why we delete the data is that too small inter-arrival times may cause a wrong estimation of the model. For example, in the input 1 data, we found more than 400 times of neuron firing out of 779. In addition, by looking at raster plot A.2, we can find most of events were happening at times between 0 and 500. According to K-S plot, deleting small inter-arrival times improves the estimation, it is still bad, but the K-S line is much close, to the 45 degree line.

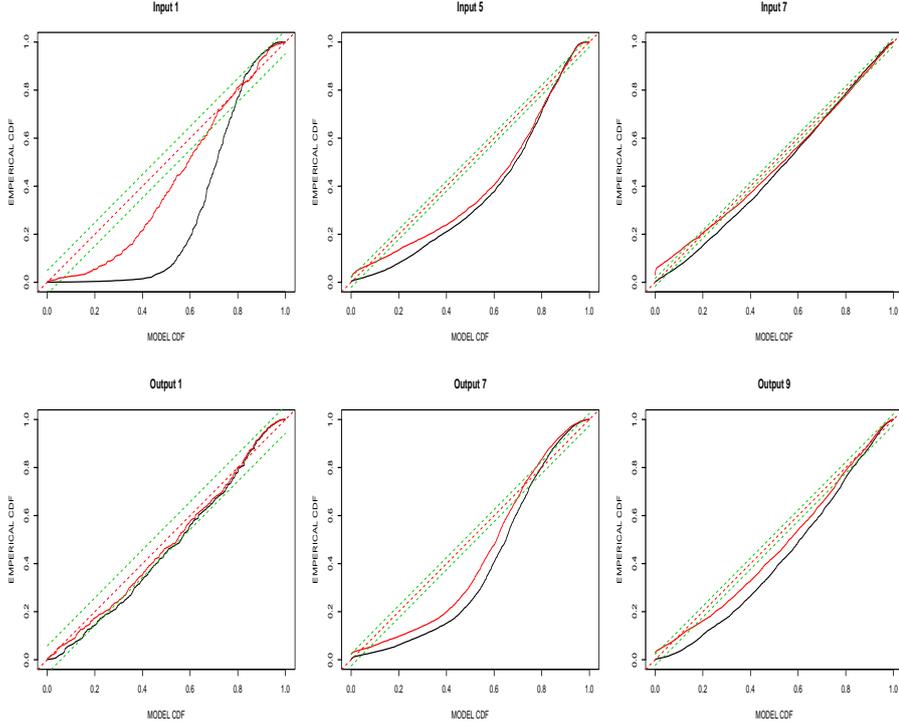


Figure 3.3: Kernel estimation using B-spline basis function and Kolmogorov-Smirnov plot, (black line-original data, red line-modified data)

3.5 Discussion and conclusion

In the previous section, some kernel have bad estimation according to residual analysis. We tried to improve the model by deleting the small inter-arrival times and using a history dependent model. We found that very small inter-arrival times may cause bad estimation and deleting them improved the model estimation, however deleting data from the original dataset is not a good idea. Hence we tried to cut the data and estimate kernel function separately for the input 1 data. It also improved the model estimation but it does not tell why other neuron have bad estimation. In addition, we used additive model, adding historical terms in the model, but there was a computation problem that adding all history dependence does not lead to convergence.

Chapter 4

Non-parametric point process modelling for multiple neurons

4.1 Introduction

In the previous chapter, we perform the parametric and non-parametric point process modelling for the single neuron data. Since we had technical problems, the result seemed quite bad. But we found that some individual neurons have similarities and the main aim of this chapter is to find the similarities within multiple neurons and show what is the proper method to analyse the multivariate neurons. One simple way of representing the multivariate neuron is to simply repeat the analysis, but it is time consuming and it may not capture the variation from the neurons. The alternative approach is to use the functional data analysis.

To begin with, functional data refers to data where each observation is a curve, a surface, as opposed to a point or a finite-dimensional vector. In this study, we want to characterise the neuron activities. Spike train data which were collected over time at different regions are producing a curve over the observation interval. Although each measurement is a discrete time point, the collection of points possess a certain smoothness property that facilitates the functional data interpretation. The main difference between multivariate analysis and FDA is that the FDA manipulates the functional representation of each individual and therefore can take into account smoothness of the considered functions.

The functional principal component(FPCA) analysis is usually used to find out the

dominant modes of the variation of functional data. In particular, the covariance function for functional data is a smooth fixed function, whereas in high-dimensional multivariate data, the dimension of covariance matrices is assumed to increase with sample size. In principle, the FPCA applies for the functional regression model, but the neuron data set consist of point process data. Hence we will show how to apply FPCA on the point process data. We tried two different approach of FPCA on the point process, one is a common approach [21],[22] which smooths the individual curves of the data, which is a established functional data analysis method. Another approach has proposed by [35], to supplement the non-parametric estimation of object-specific local intensity functions when the data have rare events. The first approach is more likely to depend on the local sparsity from the pre-smoothing individual trajectories whereas the second approach is more likely to rely on the global sparsity from the smoothing for the pooled trajectory. With four different simulation studies, these two approach will be tested through average mean squared error and Kolmogorov-Smirnov plot.

To develop functional data analysis tools for point process, we met some challenges and our proposed methods is hence different from the literature in a number of ways. First functional data analysis[?], the data consist of n independent units. However in our data set is the realisation of temporal point process and the data is correlated with time. Secondly in the classical way of FDA the functional trajectories can be directly observed, however we first tried to establish covariance structure of the process, we propose the a novel methods based on composite kernel estimation and principal component analysis. Therefore the random effects can be highlight with unusual high event rates or increasing trends in event rates.

This study is organised as follows. We introduce the definition of the classical principal component analysis and the functional principal component analysis. The functional modelling of point processes is applied to analyse the multivariate neuron spike train in section 2. In addition, a computational method for choosing the bandwidth for a kernel density and the number of principal components will be explained. A simulation study with four different scenarios is shown in section 3. Consistency results are the theme of Section 4. The conclusion of the study can be found in Section 5.

4.2 Method

4.2.1 Principal component analysis

Principal component analysis (PCA) is the most common multivariate statistical method and it is used by almost all scientific disciplines. There are several advantages of using PCA, firstly it is possible to extract the most important information from the data, secondly it can compress the size of the data set by keeping only this important information, thirdly the description of the data set can be simplified and finally it is good to analyse the structure of observations and the variables.

Principal component analysis for multivariate data

PCA computes new variables called principal components which are obtained as a linear combinations of the original variables. Conditional on a realisation f_i and N_i for the i -th neuron, we have observed event times $x_i = (x_{i1}, \dots, x_{iN_i})'$ with variance-covariance matrix $var(x_i) = \Sigma_i$. Consider the linear combination, the principal component score φ_i can be expressed

$$\varphi_i = \sum_{j=1}^{N_i} \xi_j x_{ij}, i = 1, \dots, N$$

where ξ_j is a weighting coefficient applied to the observed event time x_{ij} of the j th variable. It can be express as

$$\varphi_i = \xi' x_i, i = 1, \dots, N$$

Note that φ_i is a function of random data, and so is also random. Therefore it has a population variance and covariance

$$var(\varphi_i) = \sum_{j=1}^{N_i} \sum_{l=1}^{N_i} \xi_{ij} \xi_{il} \sigma_{jl} = \xi'_i \Sigma_i \xi_i$$

$$cov(\varphi_i, \varphi_m) = \sum_{j=1}^{N_i} \sum_{l=1}^{N_i} \xi_{ij} \xi_{ml} \sigma_{jl} = \xi'_i \Sigma \xi_m$$

For multivariate data, we choose the weight which indicates the type of variation which represents the data. Principal component analysis can be defined in terms of the following stepwise procedure. The first principal component φ_{i1} is required to have the largest

possible variance that are very strongly represented in the data. Select weighting vector $\boldsymbol{\xi}_1 = (\xi_{11}, \dots, \xi_{p1})'$ that maximize

$$var(\varphi_{i1}) = \sum_{j=1}^{N_i} \sum_{l=1}^{N_i} \xi_{1j} \xi_{1l} \sigma_{jl} = \boldsymbol{\xi}_1' \boldsymbol{\Sigma} \boldsymbol{\xi}_1$$

subject to the constraint that

$$\sum_{j=1}^{N_i} \xi_{1j}^2 = \|\boldsymbol{\xi}_1\|^2 = 1$$

The second principal component φ_{i2} is required to have a largest remaining variance as possible with constraint that the correlation between first and second principal component is 0. Select weighting vector $\boldsymbol{\xi}_2 = (\xi_{12}, \dots, \xi_{p2})'$ that maximize

$$var(\varphi_{i2}) = \sum_{j=1}^{N_i} \sum_{l=1}^{N_i} \xi_{2j} \xi_{2l} \sigma_{jl} = \boldsymbol{\xi}_2' \boldsymbol{\Sigma} \boldsymbol{\xi}_2$$

subject to the constraint that the sums of squared coefficients add up to one the same as the constraint for the first principle component and the additional constraint that these component will be uncorrelated with one another.

$$cov(\varphi_{i1}, \varphi_{i2}) = \sum_{j=1}^{N_i} \sum_{l=1}^{N_i} \xi_{1j} \xi_{2l} \sigma_{jl} = \boldsymbol{\xi}_1' \boldsymbol{\Sigma} \boldsymbol{\xi}_2 = 0$$

On the m th step, compute a new weight vector $\boldsymbol{\xi}_m$ with component ξ_{mj} and new scores $\varphi_{ml} = \boldsymbol{\xi}_{ml}' x_i$. Thus, the values φ_{ml} have maximum variance, subject to the constraint $\|\boldsymbol{\xi}\|^2 = 1$ and the $m - 1$ additional constraint

$$\sum_l \xi_{mj} \xi_{ml}' = \boldsymbol{\xi}_j' \boldsymbol{\xi}_m = 0, j < m$$

Principal component analysis and eigen analysis

Assume that the observed value x_{ij} in the multivariate context is $x_i(t)$ in the functional situation. Principal components analysis in multivariate data analysis can be defined by finding eigenvalues and eigenvectors of the covariance matrix. Let the $N \times N$ matrix \mathbf{X} contain the values x_{ij} and the vector $\boldsymbol{\xi}$ of length N_i contain the weight for linear

combination.

$$\max_{\xi' \xi = 1} \xi' \mathbf{X}' \mathbf{X} \xi$$

Since the principal component score f_i can be written as $\mathbf{X} \xi$. Let Σ is the $N_i \times N_i$ matrix to indicate the sample variance-covariance matrix $\Sigma = \frac{1}{N} \mathbf{X}' \mathbf{X}$. By eigenequation, the maximization problem is solved

$$\Sigma \xi = \rho \xi$$

There are several number of different pairs of eigenvalues and eigenvectors which satisfied this equation, and the all eigenvectors ξ_j are orthogonal. For each j , the eigenvector ξ_j satisfies the maximisation problem 4.2.1 subject to the additional constraint of being orthogonal to all eigenvectors ξ_1, \dots, ξ_{j-1} . For the functional context of PCA, the covariance function $v(s, t)$ defined as

$$v(s, t) = N^{-1} \sum_{i=1}^N x_i(s) x_i(t)$$

$$\int v(s, t) \xi(t) dt = \rho \xi(s) \quad (4.1)$$

for an appropriate eigenvalue ρ . The left side of 4.1 is an integral transform V of the weight function ξ defined by

$$\Sigma \xi = \int v(\cdot, t) \xi(t) dt$$

This integral transform is called the covariance operator Σ . Hence it the eigenequation can be expressed directly as

$$\Sigma \xi = \rho \xi$$

Principal component analysis for functional data

In the functional context, the counterpart of variable values are function values $x(s)$, so that the discrete index j in the multivariate context has been replaced by the continuous index s . The principal component score in the multivariate case was a linear combination of variable values, but in the functional case, the appropriate way of combining a weight function $\xi(s)$ with a data function $x(s)$ is a form of integration over s to define the inner product.

$$f_i = \int \xi x_i = \int \xi(s) x_i(s) ds.$$

The first procedure for functional PCA is to choose the weight function $\xi(s)$ to maximise $N^{-1} \sum_i f_{i1}^2 = N^{-1} \sum_i (\int \xi_1 x_i)^2$ subject to $\int \xi_1(s)^2 ds = \|\xi_1\|^2 = 1$ of the unit sum of squares constraint. Then as for multivariate PCA, the weight function ξ_m also need to satisfy the orthogonality constraint $\int \xi_j \xi_l = 0, j < l$ in subsequent steps.

4.2.2 Functional density model with point process data

In the concept of stochastic processes, any single real random variable is a stochastic process $\{x_t : t \in \tau\}$ with $\tau = \{t_1\}$. When $\tau = (t_1, t_2, \dots, t_k) \in \tau^k$ the stochastic process $\{x_t : t \in \tau^k\}$ becomes a sequence of random variables. Thus the point process data can be generated from the intensity function, and we assume that the neuron firing spike train produced by an intensity process. Based on this assumption, we produce the random density function. Let our neuron firing time data $t_{i1}, t_{i2}, \dots, t_{iN_i}$ observed at the i -th day during the Big-square task, the sample $x_{ij} \sim f_i, j = 1, \dots, N_i$. The neuron firing times x_{ij} is a single recurrent event process starting for simplicity at $t = 0$, and it is observed as ordered sequences $x_{i(1)} \leq x_{i(2)}, \dots \leq x_{i(N_i)}$. The sparsity of observed event times $x_{ij}, i = 1, \dots, N, j = 1, \dots, N_i$, makes it difficult to infer the individual densities with classical density estimation methods. One of the classical method performs first the principal component and principal component score for individual functions. Then those functions are applied to the spline regression for smoothing. Another way of FPCA for point process performs smoothing methods first and then performs the principal component analysis to find PC and PC score. Both of ways seems to be chasing the same thing but we assume that the later approach performs the FPCA for the original functional data, the first approach performs the FPCA for the smoothed function, hence it refer the precise random effects. In addition, the second approach has a better estimation for the small data set.

Once the random density f is estimated, then define the mean density function $Ef(x) = f_\mu(x)$ and covariance function $cov[f(s), f(t)] = v(s, t)$. The functional density can be expressed by the Karhunen-Loeve representation

$$f(x) = f_\mu(x) + \sum_k \varphi_k \xi_k(x), x \in [0, T] \quad (4.2)$$

where φ_k is the functional principal component scores. As we assume that φ_k is a

zero-mean random field with $E\varphi_k^2$ variance ρ_k , and $\sum_k \rho_k < \infty$, where the eigenvalue has ordered sequence $\rho_1 > \rho_2 > \dots$. The number of principal components k can be ∞ in theory, but is often assumed to be finite for practical considerations.

4.2.3 Kernel density

In this section, we try to predict the object-specific densities f_i from the observed sparse event data based on the expansion in 4.2. The mean density function $f_\mu = E(f_i)$ was calculated by using pooled data $N_T = \sum_{i=1}^N N_i$ and applying kernel density estimation. The estimated mean density function is defined as

$$\hat{f}_\mu = \frac{1}{N_T h_\mu} \sum_{i=1}^N \sum_{j=1}^{N_i} \kappa_1 \left(\frac{x - x_{ij}}{h_\mu} \right)$$

where h_μ is a bandwidth which was selected by the rule-of-thumb, and κ_1 is a kernel function. We tried several different kernels (Gaussian, triangular, biweight and Epanechnikov kernel), then we decided to use the Epanechnikov kernel. To estimate the covariance function $v(s, t)$ between jointly distributed random variables s and t , we use

$$v(s, t) = cov[f(s), f(t)] = E[f(s)f(t)] - Ef(s)Ef(t) = g(s, t) - f_\mu(s)f_\mu(t)$$

Then $\hat{g}(s, t)$ is estimated by a two-dimensional kernel density with pooled pairs of event data $N'_T = \sum_{i=1, N_i \geq 2}^N N_i(N_i - 1)$,

$$\hat{g}(s, t) = \frac{1}{N'_T} \sum_{i=1, N_i \geq 2}^N \sum_{j=1}^{N_i} \sum_{l=1, l \neq j}^{N_i} \kappa_2 \left(\frac{s - x_{ij}}{h_{g_1}}, \frac{t - x_{il}}{h_{g_2}} \right)$$

Here h_{g_1}, h_{g_2} are bandwidths which were selected by the rule-of-thumb and κ_2 is a symmetric two-dimensional probability density function. We use the two-dimensional Epanechnikov kernel for κ_2 . The estimated covariance function $\hat{v}(s, t)$ is then

$$\hat{v}(s, t) = \hat{g}(s, t) - \hat{f}_\mu(s)\hat{f}_\mu(t)$$

Once we obtain this covariance function $\hat{v}(s, t)$, the eigenvalues $\hat{\rho}_k$ and the eigenfunction $\hat{\xi}_k(x)$ can be obtained. Then the individual trajectories can be determined by using an object's functional principal component scores $\varphi_k, k = 1, \dots, K$. According to 4.2, we

observed $\varphi_{ik} = \int (f_i(x) - f_\mu(x))\xi_k(x)dx$, then the estimates are follows

$$\hat{\varphi}_{ik} = \frac{1}{N_i} \sum_{j=1}^{N_i} \hat{\xi}_k(x_{ij}) - \int \hat{f}_\mu(x)\hat{\xi}_k(x)dx$$

Including how many number of eigen-composition K is important, the choice of the number of principal component K will explain next section. Once K has been chosen, the prediction for the object-specific density \hat{f}_{d_i} is given by

$$\hat{f}_{d_i}(x) = \hat{f}_\mu(x) + \sum_{k=1}^K \hat{\varphi}_{ik}\hat{\xi}_k(x), x \in [0, T]$$

Given the density \hat{f}_{d_i} , intensity functions are estimated by

$$\hat{\lambda}_i(x) = N_i\hat{f}_i(x), i = 1, \dots, N \quad (4.3)$$

Bandwidth selection

A good introduction to kernel density estimation with an interesting collection of its use in data analysis is given by [28]. The choice of bandwidth h is much more important for the behaviour of \hat{f}_h than the choice of kernel κ . Small values of bandwidth h cause an overestimation which may have too many peaks, whereas too big values of the bandwidth h will lead to an over smoothing in the sense that it is biased and may not reveal structural features. The bandwidth selection methods studied in the literature can be divided into two broad classes. One class of methods is given by classical methods, such as, Cross validation, Mallows' C_p , Akaike's information criterion and the like [14]. These methods are extensions of methods used in parametric modelling. The second class of methods are plug-in methods which are based on the asymptotically best choice of h given in equation 4.4 while minimize an estimate of the asymptotic mean integrated squared error $AMISE(h)$ in order to find the optimal bandwidth \hat{h} . In the denominator $R(f^{(2)})$ is an unknown function.

It is necessary to choose a measure of distance between the true density f and the estimator \hat{f} to evaluate the performance of the kernel density estimator \hat{f}_h . The common

choice to measure the distance are the integrated squared error (ISE)

$$\text{ISE}(h) = \int \left\{ \hat{f}_h(x) - f(x) \right\}^2 dx$$

and its expected value, the mean integrated squared error (MISE),

$$\text{MISE}(h) = E \left[\int \left\{ \hat{f}_h(x) - f(x) \right\}^2 dx \right]$$

Define that \hat{h}_0 the minimizer of $\text{ISE}(h)$ and by h_0 the minimizer of $\text{MISE}(h)$.

$$\begin{aligned} \text{MISE}(h) &= \int E \left\{ \hat{f}_h(x) - f(x) \right\}^2 dx \\ &= \int E \left\{ \hat{f}_h(x) - E[\hat{f}_h(x)] + E[\hat{f}_h(x)] - f(x) \right\}^2 dx \\ &= \int E \left\{ \hat{f}_h(x) - E[\hat{f}_h(x)] \right\}^2 + \left\{ E[\hat{f}_h(x)] - f(x) \right\}^2 dx \\ &= \int \text{Var} \hat{f}_h(x) dx + \int \text{bias}^2 \hat{f}_h(x) dx \end{aligned}$$

$\text{MISE}(h)$ is the sum of the integrated variance of $\hat{f}_h(x)$, $IV(h) = \int \text{Var} \hat{f}_h(x) dx$, and $IB(h)$ is the integrate squared bias of $\hat{f}_h(x) = \int \text{bias}^2 \hat{f}_h(x) dx$. Then by change of variables and Taylor expansion of f it is easy to show that

$$\begin{aligned} IV(h) &= \frac{R(\kappa)}{N_i h} + \frac{1}{N_i} R(f) + O(N_i^{-1} h^k) \\ IB(h) &= \frac{h^{2k}}{(k!)^2} \mu_k^2(\kappa) R(f^{(k)}) + O(h^{2k+4}) \end{aligned}$$

Thus it is clear that for $\text{MISE}(h_0) \rightarrow 0$ with $N_i \rightarrow \infty$ it is necessary and sufficient that $h_0 \rightarrow 0$ such that $N_i h_0 \rightarrow \infty$ with $N_i \rightarrow \infty$. For the mathematical details see [26],[34].

Define the Asymptotic Mean Squared Error (AMISE) as

$$\text{AMISE}(h) = \frac{1}{N_i h} R(\kappa) + h^{2k} (\mu_k(\kappa)/k!)^2 R(f^{(k)}) \quad (4.4)$$

It follows that for h , such that $h \rightarrow 0$ and $N_i h \rightarrow \infty$, we have $\text{MISE}(h) = \text{AMISE}(h) + o(\text{AMISE}(h))$. Denote by \hat{h} the minimizer of $\text{AMISE}(h)$. \hat{h} is easily obtained from equation 4.4 by differentiating with respect to h and calculating the root of the derivative.

This results in the minimiser of AMISE :

$$\hat{h} = \left(\frac{R(\kappa)}{\mu_k^2(\kappa) R(f^{(k)})} \right)^{\frac{1}{2k+1}} N_i^{-\frac{1}{2k+1}}$$

We discuss the three different bandwidths which are h_0 , \hat{h}_0 and \hat{h} for the minimiser of *MISE*, *ISE* and *AMISE* respectively. These bandwidths can be an optimal bandwidth for density estimation. However the unknown density f makes it difficult to estimate the optimal bandwidth. Here is the method to estimate the optimal bandwidth. The rule of thumb replaces the unknown density function κ by a reference distribution function which is rescaled to have variance equal to the sample variance[28]. Then the estimate

$$\hat{h}_{rot} = 1.06\hat{\sigma}n^{-1/5}$$

where $\hat{\sigma}^2$ is a sample variance. A more robust version of this rule of thumb method against outliers in the sample can be constructed by using the interquartile range as a measure of spread instead of the standard deviation. The modified estimator is

$$\hat{h}_{rot} = 1.06\min\left(\hat{\sigma}, \frac{\hat{q}_{0.75} - \hat{q}_{0.25}}{1.34}\right)n^{-1/5}$$

where $\hat{q}^{0.75}$ and $\hat{q}^{0.25}$ are the 75% and 25% sample quantiles.

4.2.4 Choice of the number of principal components

One of the simplest way to determine the number of principal components is the percentage of the cumulative variance and Kaiser's stopping rules. At first, the Kaiser's stopping rule states that only the number of principal components with eigenvalues over 1.00 should be considered in the analysis. Another method that is close to Kaiser's stopping rule is the percentage of the cumulative variance. If we select all positive eigenvalues then the percentage of cumulative variance is 100%, but that does not mean anything. The acceptable level of explained variance depends on how to use principal components. It can be 80% or 90%, depending on the analysis.

Another way to determine the number of include functional principal components is by the pseudo-poisson information criterion which is borrowing the idea of pseudo-likelihood estimation. We introduce the pseudo maximum likelihood techniques, which was applied to basic Poisson models [11], to investigate how many number of principal component are needed. AIC type criteria are computationally more efficient but give similar results compared to other methods such as cross validation [36]. In section 3.4, we explained about the object-specific density $\hat{f}_i(x)$ which contains the effect of

the principal components. Denote the estimated density \hat{f}_{iK} that contains K number of principal components. Then the Poisson deviance for the observed data $x_{ij}, i = 1, \dots, N, j = 1, \dots, N_i$ is approximated by

$$D(K) = 2 \sum_{i=1}^N \sum_{j=1}^{N_i} \left[f_i(x_{ij}) \log \left(\frac{f_i(x_{ij})}{\hat{f}_{iK}(x_{ij})} \right) - (f_i(x_{ij}) - \hat{f}_{iK}(x_{ij})) \right]$$

where the density function f_i is unknown. Hence a histogram can be substituted for a smoothed version of f_i . To construct a histogram, first set a small bin size. Given the sample $x_1, \dots, x_{N_i} \in (a, b)$, the bin is defined as $a = t_0 < t_1 < \dots < t_P = b$. The bin width Δ_j of j th bin is $\Delta_j = t_j - t_{j-1}$. Let the number of observations within the j th bin is $y_j, j = 1, \dots, P$, so that $\sum_{j=1}^P y_j = m$, Then the histogram density estimation $\tilde{f}_H(x)$ of f is follows;

$$\tilde{f}_H(x) = \frac{y_j}{m\Delta_j}, j = 1, \dots, P$$

Then the deviance estimates

$$\hat{D}(K) = 2 \sum_{i=1}^N \sum_{j=1}^{N_i} \left[\tilde{f}_{iH}(x_{ij}) \log \left(\frac{\tilde{f}_{iH}(x_{ij})}{\hat{f}_{iK}(x_{ij})} \right) - (\tilde{f}_{iH}(x_{ij}) - \hat{f}_{iK}(x_{ij})) \right] \quad (4.5)$$

The pseudo-Poisson information criterion (PPIC) is then

$$PPIC(K) = \hat{D}(K) + 2K.$$

4.2.5 Component as perturbations of the mean

One of the common problems of functional PCA is that interpreting the components is usually difficult. One useful method is to examine plots of the overall mean function and the functions obtained by adding and subtracting a suitable multiple of the principal component function.

In constructing this plot, it is necessary to choose which multiple of the principal component function to use. Define a constant C to be the root-mean-square difference between $\hat{\mu}$ and its overall time average

$$C^2 = T^{-1} \|\hat{\mu} - \bar{\mu}\|^2$$

where

$$\bar{\mu} = T^{-1} \int \hat{\mu}(t) dt.$$

It is reasonable to plot $\hat{\mu}$ and $\hat{\mu} \pm \alpha C \hat{x} i$. Here α is an adjustable number while makes the interpreting the result more easily.

4.3 Simulation study

A simulation study was performed to compare the proposed multivariate functional principal component analysis with a simple simulated data set. One approach for simulating a point process is a thinning algorithm. Four different simulation studies were designed, the main purpose of the simulation study is to illustrate the implementation of the functional principal component analysis. In the first two scenarios, the simulation was designed from the equation 4.2. The third simulation was designed from the exponential form of the equation 4.2 and in fourth scenario the data were simulated from the Hawkes process.

4.3.1 Analysing the simulated dataset

Scenario 1

At first we construct the 100 true intensity functions $\lambda_i(t), i = 1, \dots, 100$. The true intensity function is the number of events multiplied by their density function $\lambda_i(t) = N_i f_i(t)$. The density function was constructed by the Karhunen-Loeve representation,

$$f_i(t) = f_\mu(t) + \sum_{k=1}^K \varphi_{ik} \xi_k(t)$$

with the mean density function $f_\mu(t) = 2 + \frac{t}{2} + \sin(t)$, and two eigen functions $\xi_1 = -\cos(\pi t/10)/\sqrt{5}$ and $\xi_2 = \sin(\pi t/10)/\sqrt{5}, 0 \leq t \leq 20$. The eigen values were chosen as $\rho_1 = 4$ and $\rho_2 = 1$, then the covariance function can be represented in terms of eigenfunctions ξ_k and eigenvalues ρ_k

$$v(s, t) = \sum \rho_k \xi_k(s) \xi_k(t)$$

The point processes data set was simulated by the thinning algorithm each true intensity function. The Table 4.1 shows that the average mean squared error, $MSE = \frac{1}{100} \sum_{i=1}^{100} \int_0^{100} \{\lambda_i(t) - \hat{\lambda}_i(t)\}^2 dt$, where $\hat{\lambda}_i(t) = \hat{f}_\mu(x) + \sum_{k=1}^K \hat{\varphi}_{ik} \hat{\xi}_k(x)$, for the true curve $\lambda_i(t)$ with the number of simulations. The number of eigenfunction K was chosen by the number of eigenvalues which represent the mode of variability over 95%. The simulation results are listed in Table 4.1. As the number of simulation increases the average mean square error decreased. The *AMSE* for the only one simulated neuron sample has approximately four times larger *AMSE* than the estimator for 100 simulated neuron sample. As increasing the number of simulated neuron sampling may reduce the *AMSE*, however we chose $n = 100$, since the discrepancy of *AMSE* from 100 to 300 or 500 seems small and consider time consumption for computation we decided to select 100 is enough.

Table 4.1: Mean squared error depending on the number of simulations

# of simulations	1	10	100	300	500
AMSE	42.81	19.52	10.86	9.89	9.03

According to the result of *AMSE*, we generated 100 simulated neurons for each true intensity function. One of our purposes was to predict the object-specific density f_i or intensities λ_i which are expressed in Equation 4.2, and Figure 4.1 shows that the object-specific intensity function for 100 simulated data for each true intensity function. The black solid lines indicate the 50 true intensity functions and the blue dotted lines show the estimated mean rate function. And the yellow dotted lines indicate the 100 individual object-specific intensity functions which contains 2 principal components. The mean intensity functions seem to well represent the true intensity function. except for the time between 19 to 20 seconds. Also all the Kolmogrov-Smirnov plot lay on the 95% confidence interval in Figure 4.2. Figure 4.2 only plotted 6 Kolmogrov-Smirnov plots for the first true intensity function, but all the other plots also lay within the confidence interval. Hence it can be said that the object-specific intensity function with Epanechnikov kernel estimation and principal component analysis well represents the true intensities.

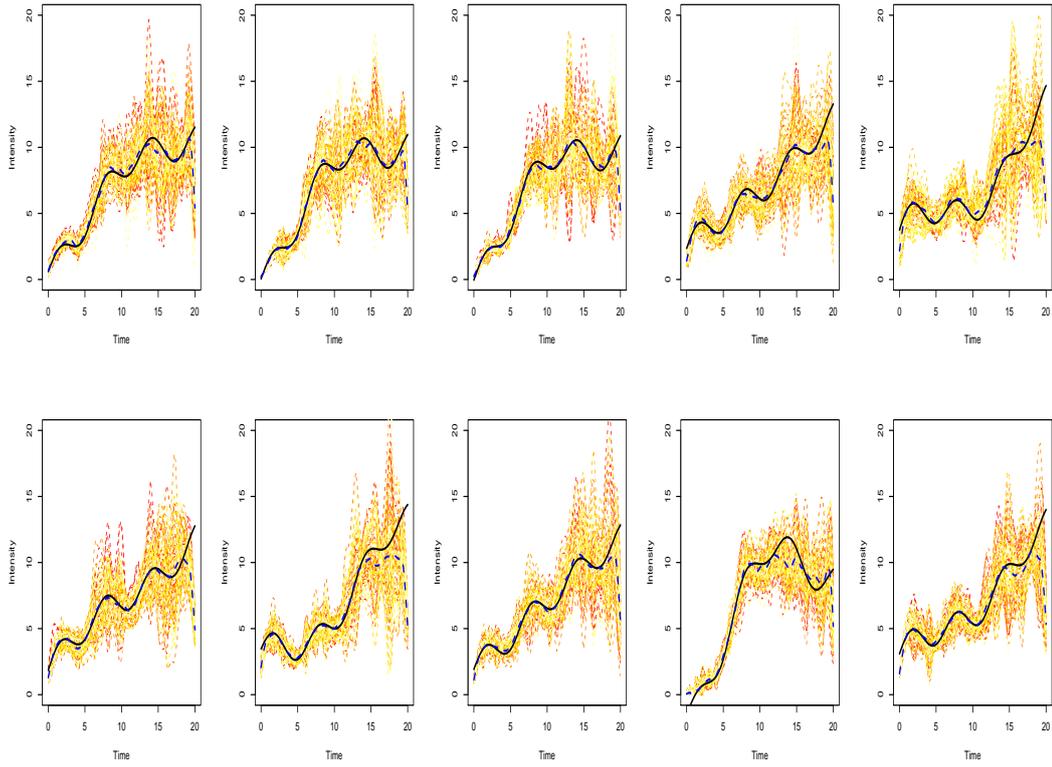


Figure 4.1: The intensity function of the true and simulated data.

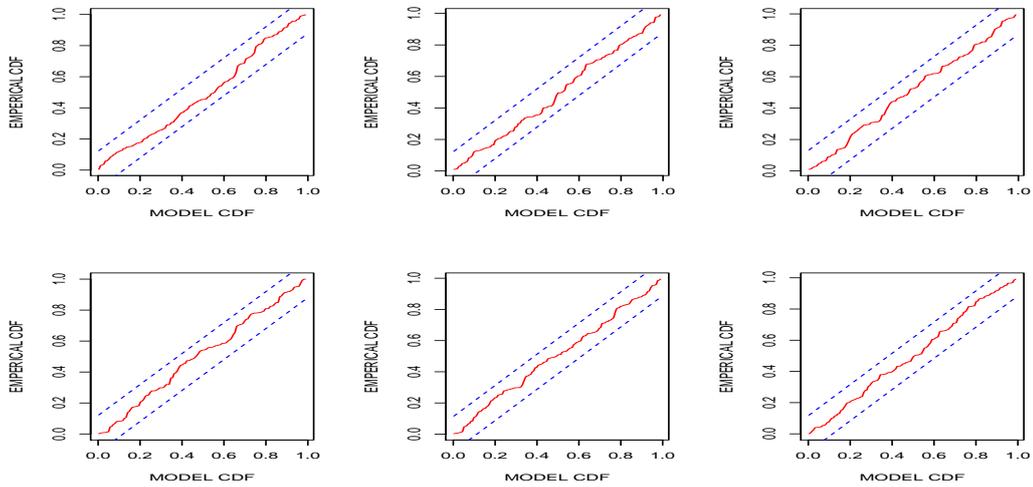


Figure 4.2: The kolmogrov-smirnov plot for the simulated data.

Scenario 2

In the second scenario, we simulate the 100 neurons which have the same and fixed mean density function $f_\mu(t)$, eigenfunction $\xi_k(t)$ and eigenvalue ρ_k which are the same

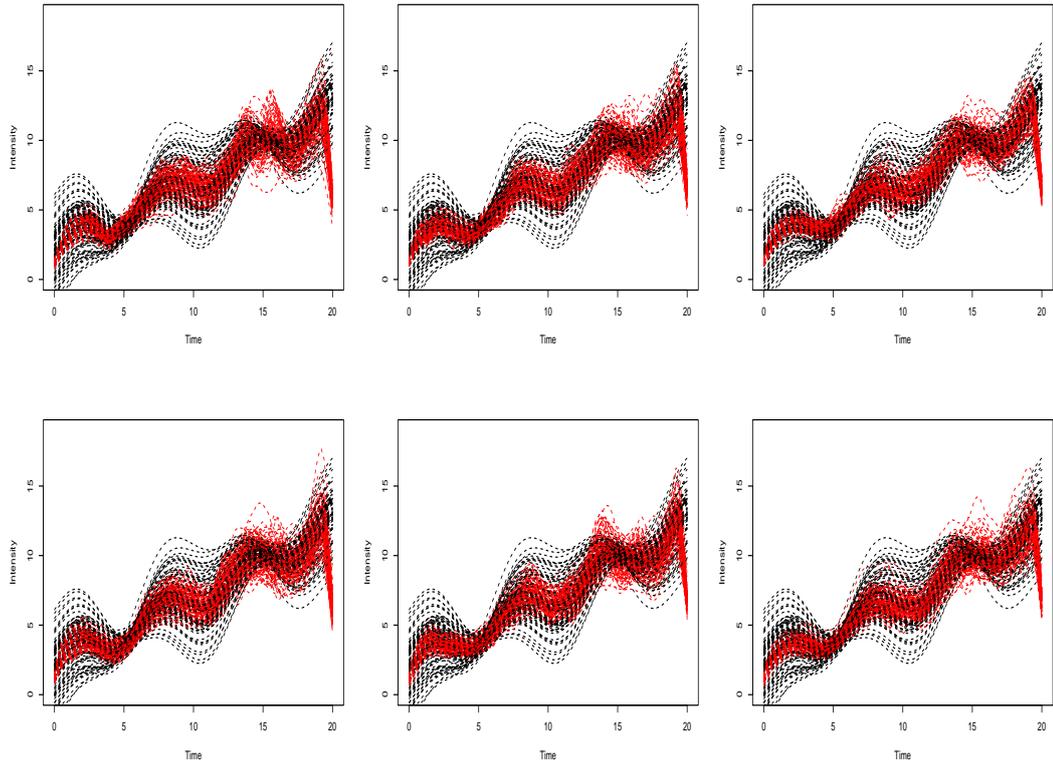


Figure 4.3: The intensity function of the true and simulated data.

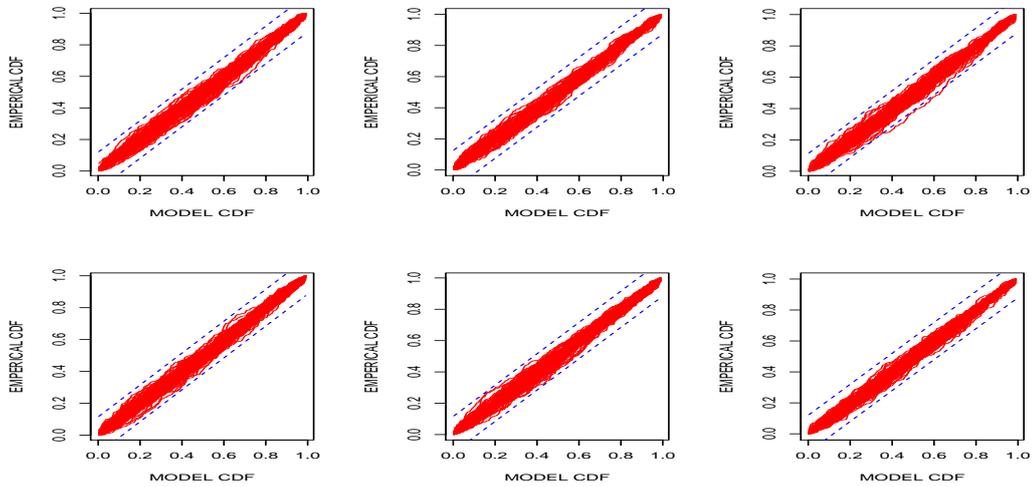


Figure 4.4: The kolmogrov-smirnov plot for the simulated data.

as the mean density function in the scenario 1 and the principal component score φ_{ik} were generated from $N(0, \rho_k)$. Each simulation was performed by thinning algorithm with rate function $\lambda_i(t)$ and this step repeated 50 times. This simulation means that we have 100 multiple neuron data set on each 50 days. The performance measure is average

mean integral squared error.

$$AMISE = \frac{1}{100 * 50} \sum_{j=1}^{50} \sum_{i=1}^{100} \int_0^T \{f_i(t) - \hat{f}_i(t)\}^2 dt$$

Table 4.2: Comparison of mean squared error of the individual intensity and object-specific intensity

	$\lambda_i(t)$	$\hat{\lambda}_{1i}(t)$	$\hat{\lambda}_{2i}(t)$
AMISE	407.58	44.58	44.83

The 100 true intensity functions and the estimated object-specific intensity functions are shown in Figure 4.3. The black dotted lines describe the true intensity functions and the red dotted lines show the estimated intensity functions. Figure 4.4 is the kolmogrov-smirnov plot for the first six repeated 100 neuron samples. In Figure 4.3, the estimated intensity functions seems to have similar pattern with true, but it does not enough to explain the variability on the peak of the true intensity function. However, most kolmogrove-smirnov plot lay in the 95% confidence interval. Therefore it can be said that the FPCA for the simulated neuron data set gives a good estimation.

Likewise the results from functional data analysis with an individual smoothing are shown in Figure ???. It is hard to define whether the intensity function is well characterised or not, there is a broad distinction between the true intensity and the estimated intensity. However the test statistics indicates that it is significantly well estimated. But compared to object-specific intensity function with average mean squared error 4.2, the FDA approach with individual smoothing has 407.58 AMISE, whereas the object-specific intensity has 44.83. Furthermore the comparison between two approaches $\hat{\lambda}_{1i}(t)$ and $\hat{\lambda}_{2i}(t)$ are shown in Table 4.2. Those two approaches do not seem to have significant difference and both approaches lead to good estimation according to AMISE and K-S plot.

Scenario 3

Table 4.3: Comparison of mean squared error of the individual intensity and object-specific intensity

	$\lambda_i(t)$	$\hat{\lambda}_i(t)$
AMISE	76.16	12.95

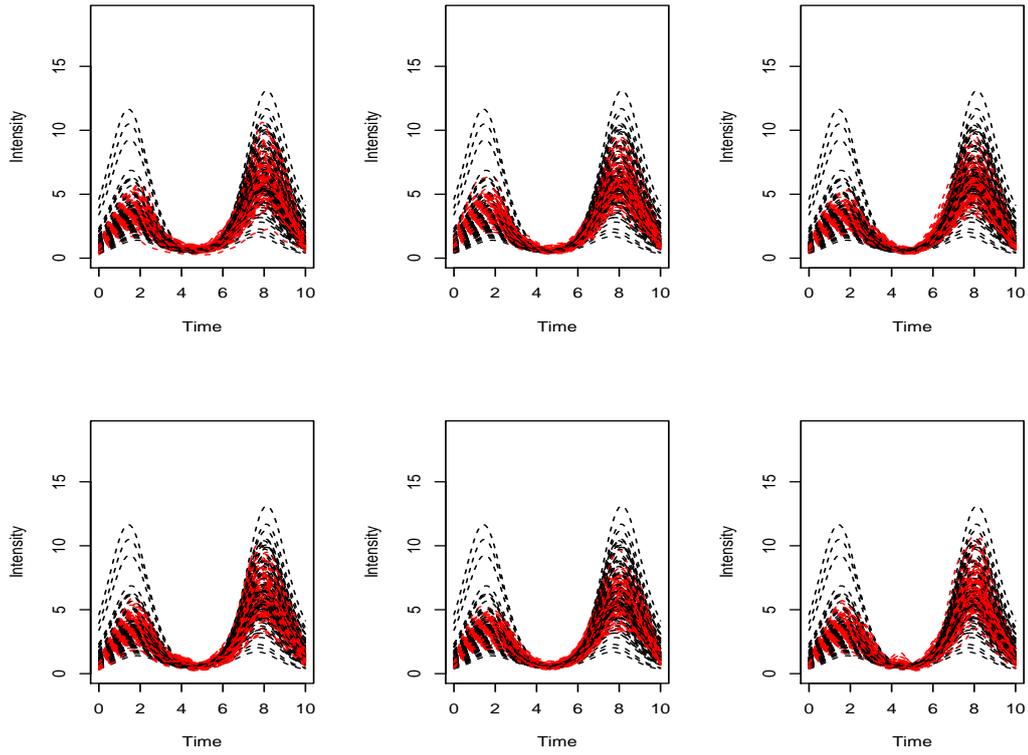


Figure 4.5: The intensity function of the true and simulated data.

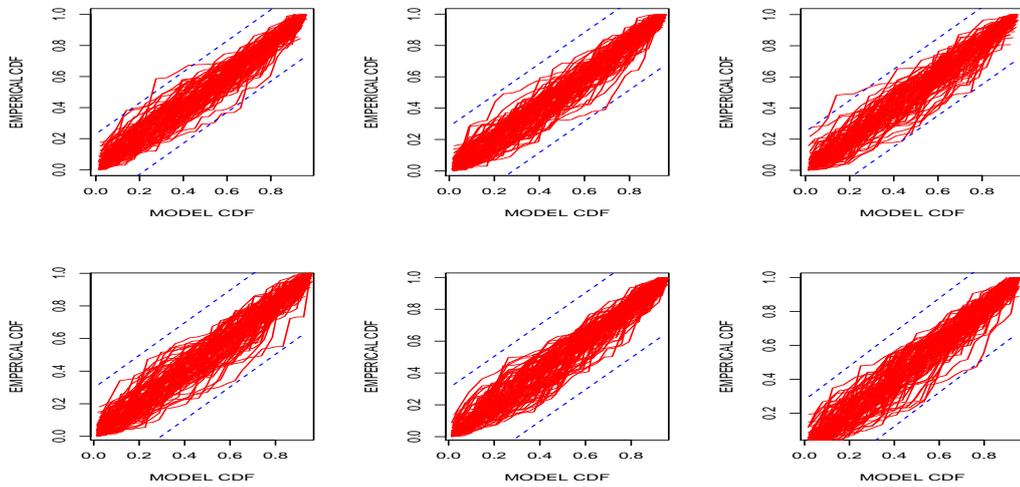


Figure 4.6: The kolmogrov-smirnov plot for the simulated data.

In the scenario 1 and 2, the density function was constructed by the Karhunen-Loeve representation,

$$f_i(t) = f_\mu(t) + \sum_{k=1}^k \varphi_{ik} \xi_k(t)$$

then we simulated 100 sample from the 1 true intensity function in scenario 1 and 100 intensity functions in scenario 2. After that we performed the functional principal component analysis, both results were quite well estimates. In scenario 3, we do the same analysis with functional principal component analysis, but the true intensity has the form of Gaussian Cox process which is not the same as Karhunen-Loeve representation. The true density function is defined as

$$f_i(t) = \exp \left(f_\mu(t) + \sum_{k=1}^k \varphi_{ik} \xi_k(t) \right)$$

where the mean density function is $f_\mu(t) = t/T + \sin(t)$ and the right side is the same as the scenario 1 and 2. We simulate 100 neuron samples from the $\lambda_i, i = 1, \dots, 100$, then perform the same analysis as in scenario 2. The average mean integral squared error (AMISE) was 12.95. The 100 true intensity functions and the estimates object-specific intensity functions are shown in Figure 4.5. The black dotted lines describe the true intensity functions and the red dotted lines show the estimated intensity functions. Figure 4.6 is the kolmogrov-smirnov plot for the first six repeated 100 neuron samples. In Figure 4.5, the estimated intensity functions seems to have very similar patterns as the true ones, only three curves have big discrepancy to the estimates. In addition, most Kolmogrove-Smirnov plots were within the 95% confidence interval. Therefore it can be said that the FPCA works well on the Gaussian Cox process.

Likewise the results from functional data analysis with an individual smoothing are shown in Appendix Figure ???. According to estimated intensity function and test statistics, they indicate that it is significantly well estimated. But compare to the object-specific intensity function with average mean squared error 4.3, the FDA approach with individual smoothing has 76.16 AMISE, whereas the object-specific intensity has 23.95. Hence the object-specific intensity has better representation than the individual smoothing approach.

Scenario 4

In this scenario, we will test that FPCA is a good method for the history dependent model. Hence we simulate the history dependent point process data set using Hawkes process. The univariate Hawkes process is defined to be a self-exciting process. If

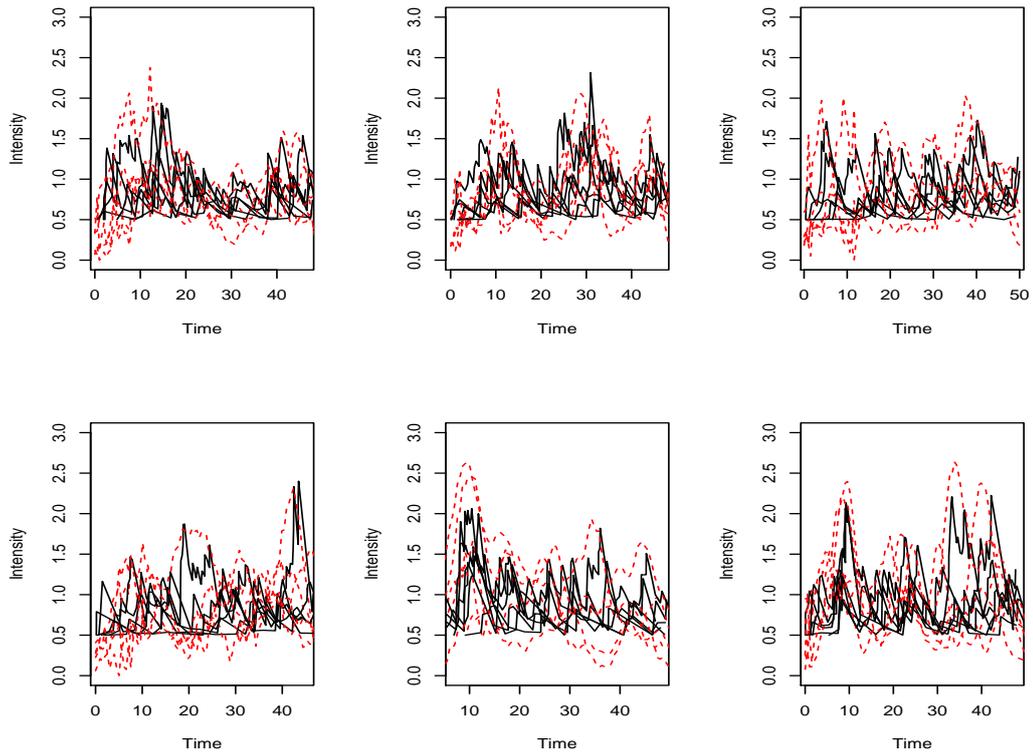


Figure 4.7: The intensity function of the true and simulated data.

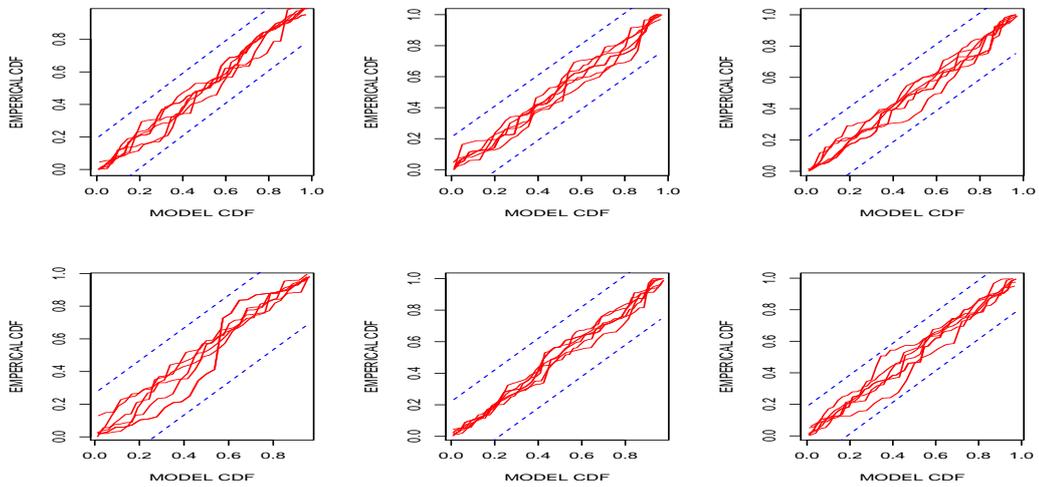


Figure 4.8: The kolmogrov-smirnov plot for the simulated data.

we assume that correlations between present and nearby events are positive, then this is called a self-exciting point process. This means that if an event occurs, another event becomes more likely to occur locally in time and space. We investigate a special class of point processes termed a Hawkes Process [12], which can be represented by the

conditional intensity function:

$$\begin{aligned}\lambda(t | H_t) &= \lambda_0(t) + \int_{-\infty}^t \nu(t-u) dN_u \\ &= \lambda_0(t) + \sum_{t_i < t} \nu(t-t_i)\end{aligned}\tag{4.6}$$

where N is the normal counting process [13]. In Equation (4.7), $\lambda_0(t)$ represents the background rate of events, which is in the simplest model assumed to be constant in time. The second half of the sum describes the self-exciting part of the process. Many choices for the kernel density ν have been proposed [12]. We use an exponential kernel

$$\lambda(t | H_t) = \lambda_0 + \sum_{t_i < t} \alpha e^{-\beta(t-t_i)}\tag{4.7}$$

Here, α is a rate of decay constant for the triggering kernel controlling how quickly the overall rate λ returns to its baseline level λ_0 after an event occurs.

We simulated only six neuron samples and repeated 10 times of the simulation. For the comparison of visualisation between true intensities and simulated neuron, we only simulated this small number of neuron data. The true intensities have the fixed λ_0 set as the 0.5 and the two parameters for the Hawkes process α and β are follows $N(0.3, 0.01)$ and $N(0.7, 0.01)$ respectively. The true intensity functions are shown in Figure 4.7 as the black solid lines. Figure 4.7 displays only six repetitions of simulations. Then we perform the functional principal component analysis, and the estimated intensity function for six neurons for each repetition were displayed as red dotted line in Figure 4.7. It only seems to capture the global sparsity, it does not capture the local sparsity of the neurons. The K-S plot represents all six neurons of first repetition lay on the 95% confidence interval, so it can not be said that it is a bad estimator. However it does not seem to fit the model which has history dependent effect.

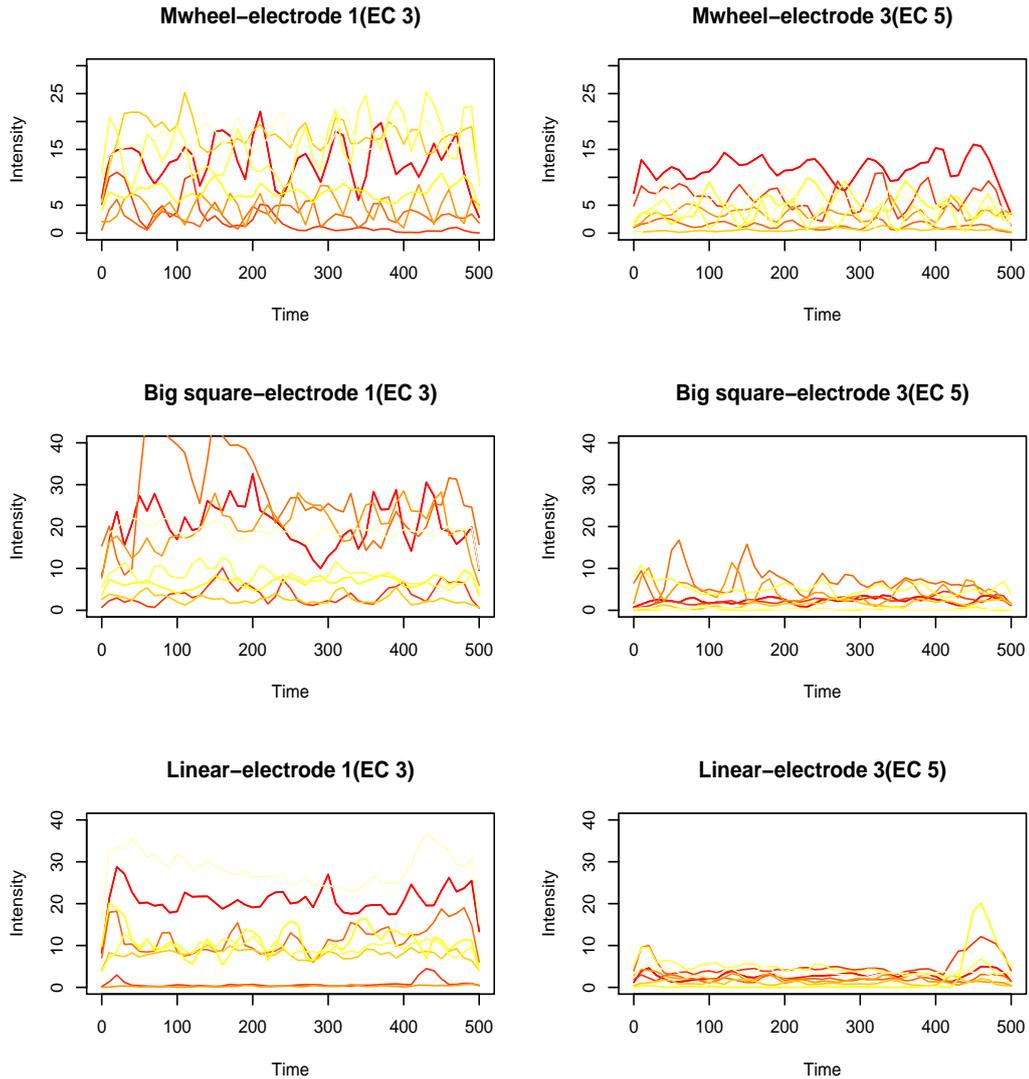


Figure 4.9: The individual neuron firing rate function.

4.4 Application to data

4.4.1 Individual intensity function

One of our purpose of the study is to estimate the density or intensity of the neuron firing event. The simplest way of the non-parametric method is the kernel smoothing. The Figure 4.9 shows the individual intensity function of the neuron firing event when the rat performed the three different tasks. In general, there are larger variabilities on the brain region EC 3 than the region EC 5. The largest variability of intensity function on the brain region EC 3 was shown on the Big-square task. The intensity function from the brain region EC 5 seem to have small variability, the discrepancy intensity functions

between brain region on the task of Big-square was more than five times. The colors indicate the experiment day such as red color denote the first day of the experiment and the yellow means the last day. In addition according to the Figure 4.9, it can be seen that if there is a high firing rate on the region EC 3 than region EC 5 fired the small amount of spike trains. However it is hart to find difference of the patterns between tasks and sessions, the major variability was dominated by the difference number of event times which affects the intensity function.

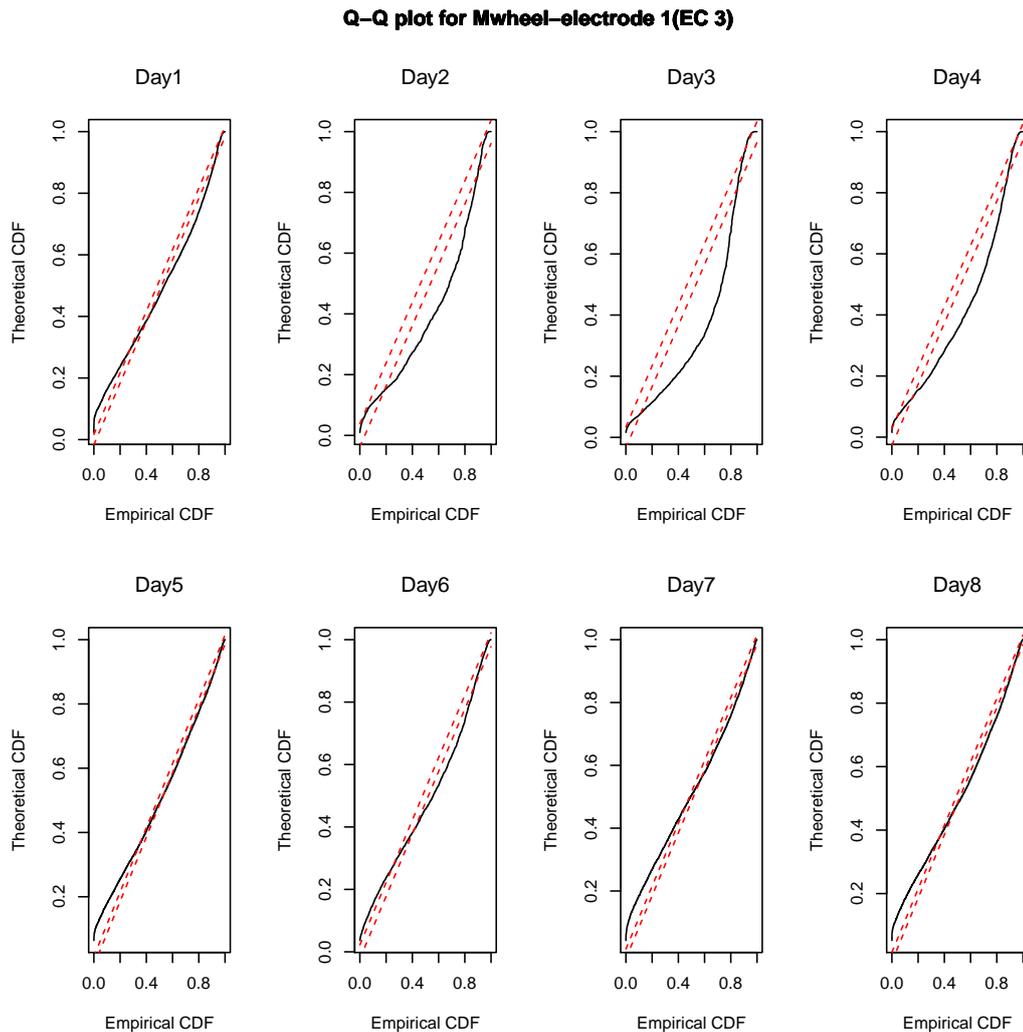


Figure 4.10: The eight days for Kolmogrove-Smirnov plot for the task on the Mwheel (EC 3).

Then we performed the time-rescaling theorem to the individual rate function. Figure 4.10 shows the Kolmogrov-Smirnov plot, the solid black line indicates the KS statistics and the dashed red line indicate the 95% confidence interval for each KS test statistics. Some confidence intervals seem too wide, since few firing events were observed. Most

of KS test statistics does not lay on the confidence interval. Therefore it can be said that the individual rate function is not well estimated, we aim at predicting the random density f_i from observed sparse event data based on the expansion in 4.2.

4.4.2 Functional principal component analysis

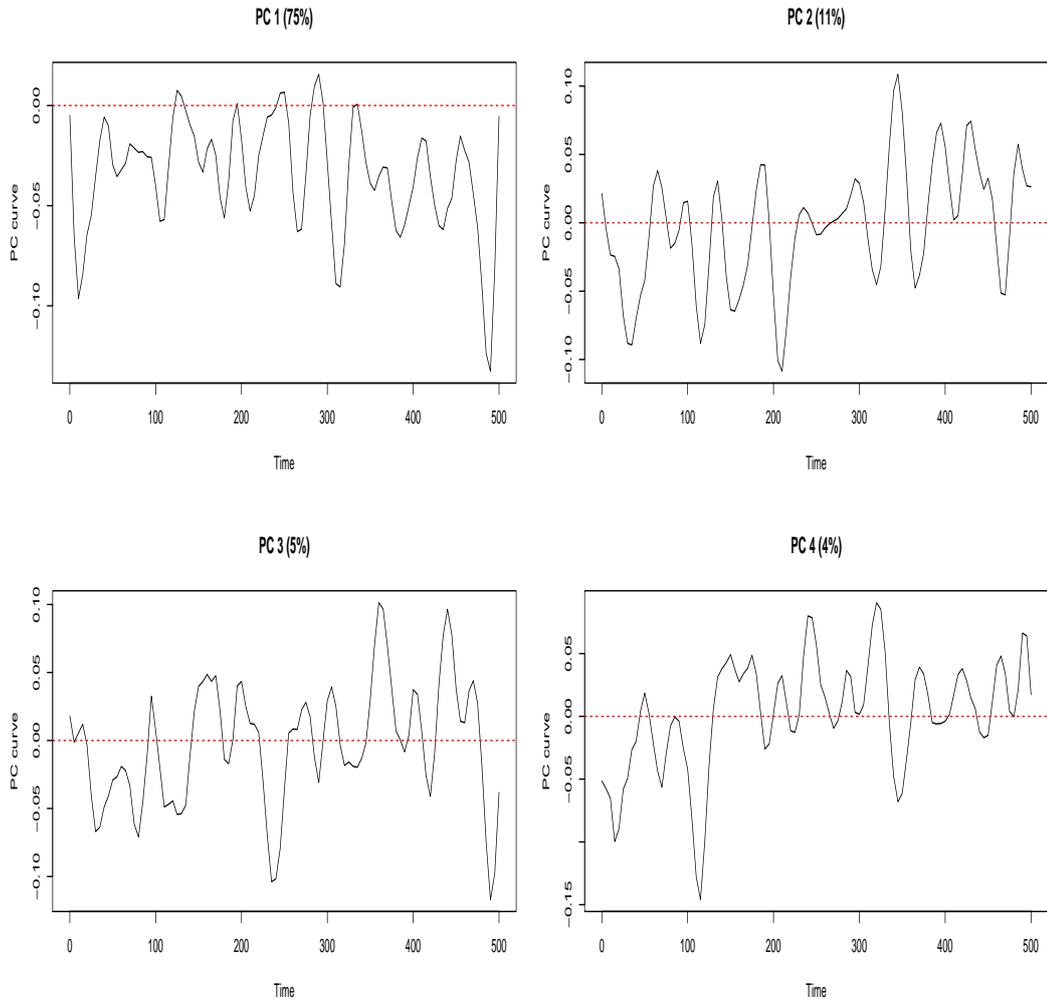


Figure 4.11: The first four principal component curves of the neuron firing data on the Mwheel task.

We discussed about principal component analysis for functional data set in section 4.2.1 and the choice of the number of principal components was discussed in section 4.2.4. The pseudo-Poisson information criterion (PPIC) shows that four principal components. Figure 4.11 displays the weight function ξ_1 to ξ_4 for the neuron firing data on the Mwheel task on the brain region EC 3.

According the ξ_1 , most of the values are negative, whereas only three points have a

positive value. This explains the discrepancy of intensity functions in Figure 4.9. It also shows that there are large variabilities in the beginning and the end of the experiment rather than in the middle of the experiment. In addition, the percentage 75% at the top of the left panel indicates that this type of variation is a dominant variation on the eight functions. The weight function ξ_2 for the neuron firing data is displayed in the upper right panel of Figure 4.11. It cannot be the mode of the variation in the neuron firing data by the set of constraint that it must be orthogonal to ξ_1 . The weight function ξ_2 is the second mode accounts for 11% of the total variation, and it seems to have large contribution at the beginning of the experiment and it gradually decrease to the negative contribution at the end of the experiment. The third component has a similar proportion of the second component, it accounts for 5%. It has a negative contribution at the beginning of the experiment and gradually increased, then it has positive and negative contribution with big fluctuations until the the end of the experiment. The fourth component account for small proportion of the variation, since it has constrains that it must be orthogonal to other components. It is difficult to interpret.

One helpful way to examine the variability of the neuron firing between different days on the same task is to produce plots of the overall mean function and the functions which are obtained by adding and subtracting a suitable multiple of the principal component function. Figure 4.12 shows such a plot for the neuron firing for the Mwheel task on the brain region EC 3. In this plot, the solid line indicates the overall mean functions and the dashed lines shows the adding and subtracting of a multiple of the principal component curves.

The random intensities of the neuron firing activity for the three different tasks are demonstrated in the Fig4.13. These random intensities contain 4 principal components because they explain more than 95 % of the variability. The results are very similar to the kernel density estimation. For the intensities on the Mwheel task, there is a large parallel discrepancy of the neuron firing rate between experiment days, since the principal component ξ_1 explains that difference, as seen on the Figure 4.12 and 4.13. The rest of three principal component helps to explains 95% of the variation. Intensities, on the other region EC 5 on the Mwheel task, have a similar pattern. The majority of the variation is a difference of firing rate between experiment days.

The first principal components ξ_1 of the Big-square on the brain region EC 3 and

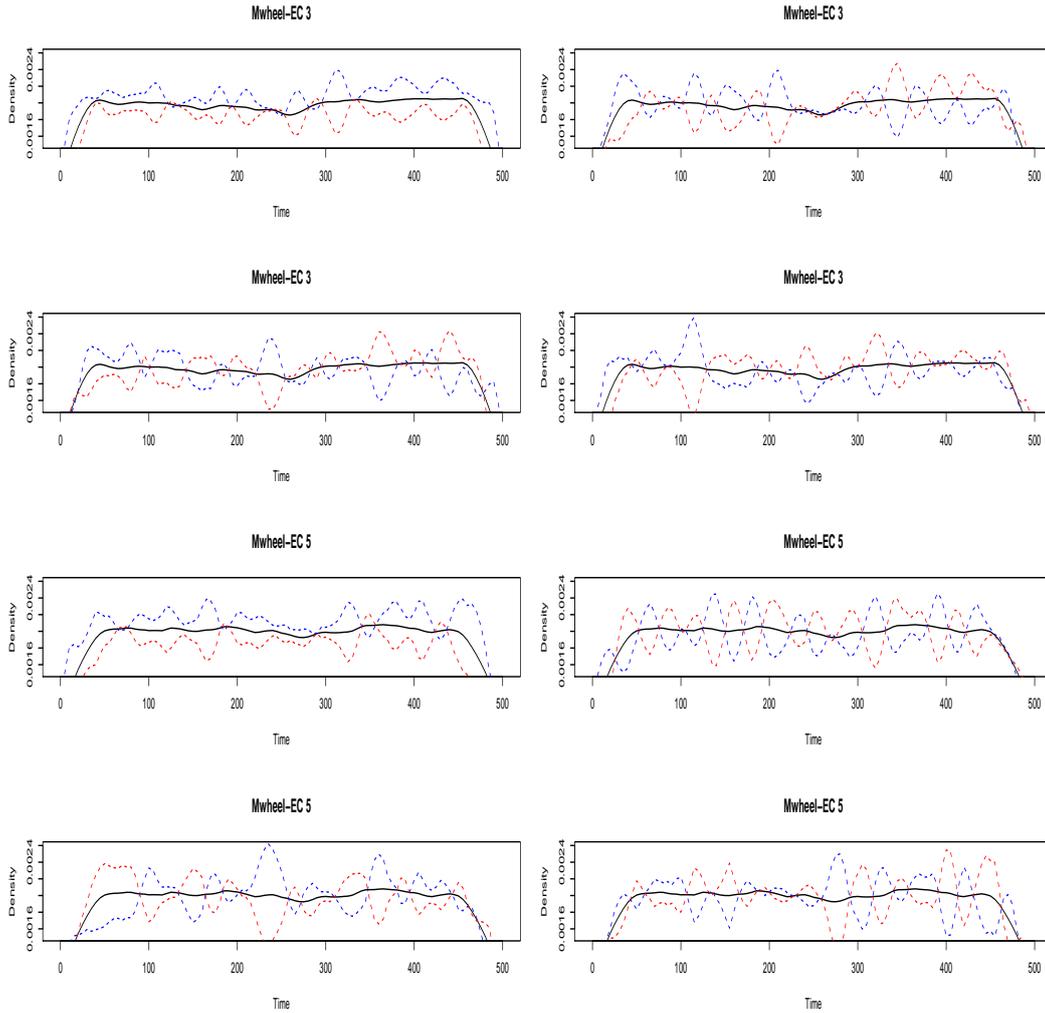


Figure 4.12: The overall mean neuron firing rate function and the effect of adding and subtracting a suitable multiple of each principal component curve.

EC 5 also explain the parallel difference of firing spike trains. However there is a big difference of the firing rate between regions, since the number of spike trains on the region EC 5 is much lower than region EC 3. The intensity functions for the Linear task seem to have a similar trend with the Big-square task. Compare to the individual rate function in the section 4.1, this rate function is less smoother, since it is affected by ξ which play the role of the random effects.

4.4.3 Model checking

In the previous section 4.1 we performed the time-rescaling theorem for the individual intensities, and only a few of them lay on the 95% confidence interval. In this section, we will perform the time-rescaling theorem again to the random intensities then compare

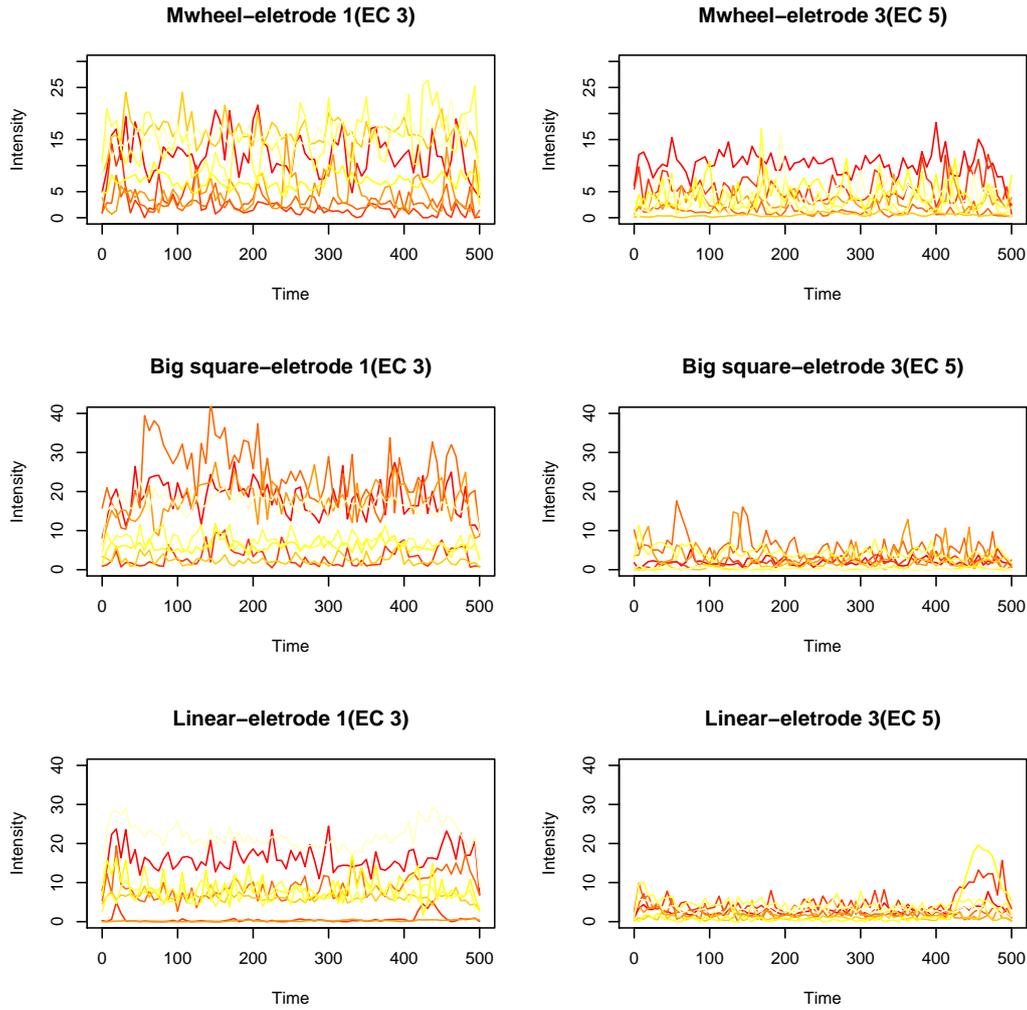


Figure 4.13: The random rate function for the neuron firing data.

with Kolmogrov-Smirnov test statistics. In addition, it will show how many principal component will be suitable. Figure 4.14 shows the Kolmogrov-Smirnov test statistics for the neuron firing rate function, the thick black solid line indicates the most suitable KS test statistics of the random intensity which includes 4 principal components. The dashed red line indicates that other random intensities with rest of principal components and the thick dashed green line displays the KS test statistics of the individual intensities. According to Figure 4.14, most of the KS test statistics of the random intensities which includes 4 principal component have better test statistics than individuals. Still lots of them are laying outside of confidence interval, hence it is hard to say that these random intensities are good estimations. However the random intensities with the selected number of principal components gives a better explanation of the firing rate of the neuron.

Kolmogorov–Smirnov plot for Mwheel

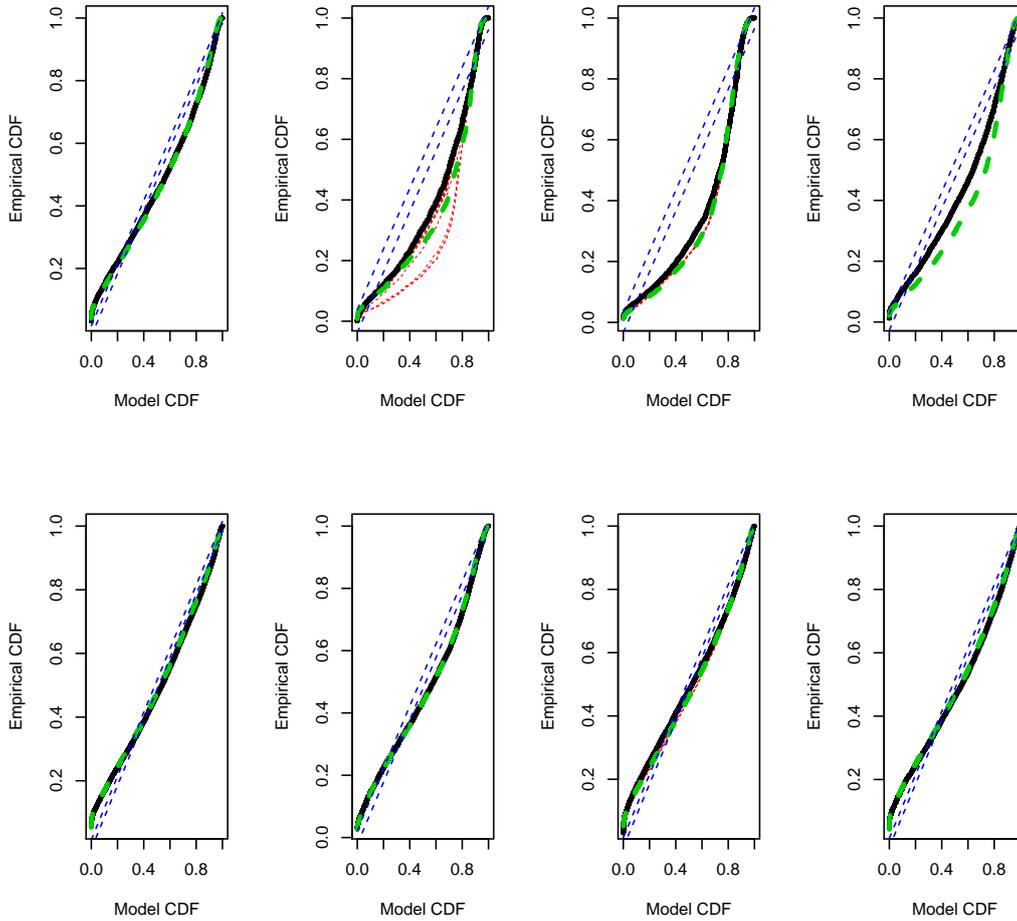


Figure 4.14: Kolmogorov-smirnov plot for the neuron firing on the Mwheel task.

4.5 Discussion and conclusion

We applied the functional data analysis method for the repeated point process data in the simulation study and one may say that it is a useful method to characterise the intensity function for individual objects. In the model checking procedure, all four different scenarios have good Kolmogorov-Smirnov statistics and the comparison of the estimated individual intensity functions with the true intensity function does not seem too different. In the scenario 4, however, the estimated intensity function does not explain the history effects which are represented in the self-exciting process. The FPCA only represents the global sparsity of the data.

The interesting finding from the simulation study is that the estimated intensity function from borrowing the strength from all objects seems more reliable than the

functional data analysis with individual smoothing method. The test statistics indicates that both methods are reliable since most of Kolmogorov-Smirnov plot lay on the 95% confidence interval, but by looking at the intensity function, the approach, which borrow the strength from all object, is more likely to capture the effect of the covariance function. In general, both approach have a good estimates for the characterising the neuron activities in the simulation study.

In the result section, the majority factor for obstruction of comparing intensity function on each neuron is the number of neuron activities. The intensity function was defined as multiply the number of events on their density function 4.3. However the number of neuron activities in real neuron have very big difference between trials. Some trial have more than 20 times larger, hence it may affect on the FPCA, the first principal components explain that difference and the proportion of that is almost 100%. It is quite difficult to analyse the characterised intensity functions, since they are too fluctuated and it is hard to find similar patterns. But they seems to have some different pattern on the region. At this stage, we can only find that there is an opposite trend of firing neuron spike, if region EC3 fire many number of spike train, there are only few number of spike trains are fired.

Chapter 5

Multivariate functional principal component analysis with multiple neurons

5.1 Introduction

In the previous chapter, we introduced functional data analysis and univariate functional principal component analysis with the multivariate neuron spike train data. We characterised each intensity function with principal components on each trials. We found that characterising the intensity functions with FPCA has better performance than their individual smoothing. At this point, the question may arise how to model the multivariate neuron spike train data on the different regions. We tried to study the variation of more than one function. Each neuron on the same task has been collected from two different regions EC3 and EC5. To understand the total system of firing activities, we want to know how the neuron firing activities are characterised in the regions EC3 and EC5 jointly. Assuming that they are measured in the same units and have similar variation, then the classical multivariate functional principal component (mFPCA) can work well. The idea of mFPCA is the extension of the FPCA idea to deal with multivariate functional data. A successful example is shown in [21]. They modelled bivariate PCA for the children's gait cycles.

This study is organised as follows. We first discuss the definition of the classical multivariate principal component analysis. A simulation study is shown in section 3.

Consistency results are the theme of Section 4. The conclusion of the study can be found in Section 5.

5.2 Method

5.2.1 Multivariate principal component analysis

In the previous chapter, we studied how the principal component analysis in multivariate data analysis can be defined by finding eigenvalues and eigenfunctions of the covariance matrix. Then we denoted that $N \times p$ matrix X which contains the values x_{ij} .

In multivariate principal component analysis, let $\{X_r\}_{r=1,\dots,p}$ be a set of random functions where each function X_k lies in the Hilbert space $L^2(\tau)$ of square integrable function for a compact domain τ . Denote by \mathbf{X} the vector $(X_1, \dots, X_k)^T$. Assume that each function X_k has an unknown smooth mean function $\boldsymbol{\mu}(t) = (\mu_1(t), \dots, \mu_k(t))^T$ where $\mu_k(t) = E(X_k(t))$ and the covariance function $\mathbf{v}(s, t) = \{v_{kl}(s, t)\}_{1 \leq k, l \leq p}$, where $v_{kl}(s, t) = cov(X_k(s), X_l(t))$. The covariance function $\mathbf{v}(s, t)$ is symmetric hence $\mathbf{v}(s, t) = \mathbf{v}(t, s)^T$, in the same sense as the covariance function for fixed k and l $v_{kl}(s, t) = v_{lk}(t, s)$. Define the inner product of two functions f and g with respect to a measure t in $L_2(\tau)$ as $\langle f, g \rangle = \int f(t)g(t)dt$ with $\| \cdot \|^2 = \langle \cdot, \cdot \rangle$. The vectors of function f_r and g_r in the Hilbert space \mathbb{H} denote as $\mathbf{f} = (f_1, f_2, \dots, f_p)^T$ and $\mathbf{g} = (g_1, g_2, \dots, g_p)^T$. The inner product of the vectors of two function \mathbf{f} and \mathbf{g} in Hilbert space \mathbb{H} is defined as $\langle \mathbf{f}, \mathbf{g} \rangle_{\mathbb{H}} = \sum_{r=1}^p \langle \mathbf{f}_r, \mathbf{g}_r \rangle$, with norm $\| \cdot \|_{\mathbb{H}}^2 = \langle \cdot, \cdot \rangle_{\mathbb{H}}$. In addition, the inner product can be expressed as $\langle f, g \rangle = \int w f(t)g(t)$, where w is the weight function.

Classical method for multivariate functional principal component analysis

In the previous chapter, we introduced the Karhunen-Loeve representation, then the multivariate random function can be expressed as

$$\mathbf{X}(t) = \boldsymbol{\mu}(t) + \sum_{k=1}^{\infty} \varphi_k \boldsymbol{\xi}_k(t)$$

where $\boldsymbol{\mu} = \mu_1, \mu_2, \dots, \mu_p$ is the mean function of \mathbf{X} , and $\boldsymbol{\xi}_k$ is an orthonormal basis function in Hilbert space \mathbb{H} and it satisfies the orthogonality constraints $\int \xi_k \xi_l = 0, k < l$. The φ_k are random variables, the functional principal component scores (FPCS). The

FPCS are associated with the eigenvalues, since their mean and variance are $E(\xi_k) = 0$, $E(\xi_k^2) = \rho_k$ and $\sum_k \rho_k < \infty$. In the previous chapter, we studied the covariance function $v(s, t)$ defined as

$$v(s, t) = N^{-1} \sum_{i=1}^N x_i(s)x_i(t)$$

$$\int v(s, t)\xi(t)dt = \rho\xi(s)$$

for an appropriate eigenvalue ρ . The left side of 4.1 is an integral transform V of the weight function ξ defined by

$$\Sigma\xi = \int v(\cdot, t)\xi(t)dt$$

This integral transform is called the covariance operator Σ . Hence the eigenequation can be expressed directly as

$$\Sigma\xi = \rho\xi$$

In multivariate principal component case, let $\mathbf{v}_k = (v_{k1}, \dots, v_{kp})^T$ and $\mathbf{v}(s, t) = \{v_{kl}(s, t)\}$.

$$\Sigma\xi(s) = \int \mathbf{v}(s, t)\xi(t)dt = \begin{pmatrix} \langle \mathbf{v}_1(s, t), \xi \rangle_{\mathbb{H}} \\ \cdot \\ \cdot \\ \cdot \\ \langle \mathbf{v}_p(s, t), \xi \rangle_{\mathbb{H}} \end{pmatrix},$$

where $\langle \mathbf{v}_k(s, \cdot), \xi \rangle_{\mathbb{H}} = \sum_{l=1}^p \langle v_{kl}(s, \cdot), \xi_l \rangle$. Since Σ is a linear operator, $\Sigma\xi(s) = \rho\xi(s)$ satisfied. The multivariate covariance function \mathbf{v} is continuous, symmetric and non-negative-definite ($\langle \Sigma\xi, \xi \rangle_{\mathbb{H}} \geq 0$ for any $\xi \in \mathbb{H}$). By the multivariate version of Mercer's theorem, $\langle \mathbf{v}_k(s, \cdot), \xi \rangle_{\mathbb{H}}$ can be written as

$$\langle \mathbf{v}_k(s, \cdot), \xi_r \rangle_{\mathbb{H}} = \sum_{l=1}^p \langle v_{kl}(s, \cdot), \xi_{lr} \rangle = \rho_r \xi_{kr}(s)$$

for $k = 1, \dots, p$ and all r . The covariance of (k, l) elements can be represented as $v_{kl}(s, t) = \sum_{r=1}^{\infty} \rho_r \xi_{kr}(s)\xi_{lr}(t)$, and the multivariate covariance function \mathbf{v} expressed as $\mathbf{v}(s, t) = \sum_{r=1}^{\infty} \rho_r \boldsymbol{\xi}_r(s)\boldsymbol{\xi}_r(t)^T$.

5.3 Simulation study

This simulation study was designed to propose the multivariate functional principal component analysis with two simple simulated neurons. The thinning algorithm was used to simulate the point process data. Two different simulations were designed, the purpose of the first simulation is to check how well mFPCA estimates bivariate functions with similar variation. The second scenario was designed to have two functions with large variation.

5.3.1 Scenario 1

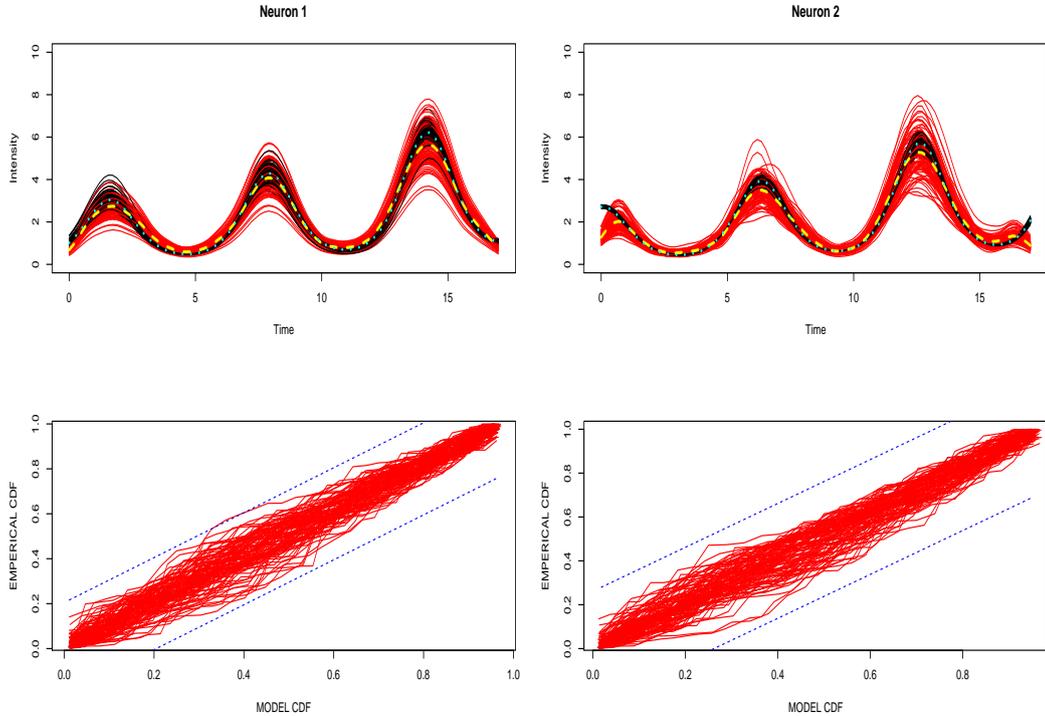


Figure 5.1: Intensity function of the true and simulated neuron and Kolmogorov-smirnov plot.

We construct the 100 true intensity functions $\lambda_i^r(t), i = 1, \dots, 100, r = 1, 2$. The true intensity function has form of the number of events multiplies their density function $\lambda_i^r(t) = N_i^r f_i^r(t)$. The density function was constructed by the exponential form of Karhunen-Loeve representation,

$$f_i^r(t) = \exp(f_\mu^r(t) + \sum_{k=1}^k \varphi_{ik}^r \xi_k^r(t))$$

where the mean density function $f_\mu^1(t) = t/T + \sin(t)$ and $f_\mu^2(t) = t/T + \cos(t)$, two

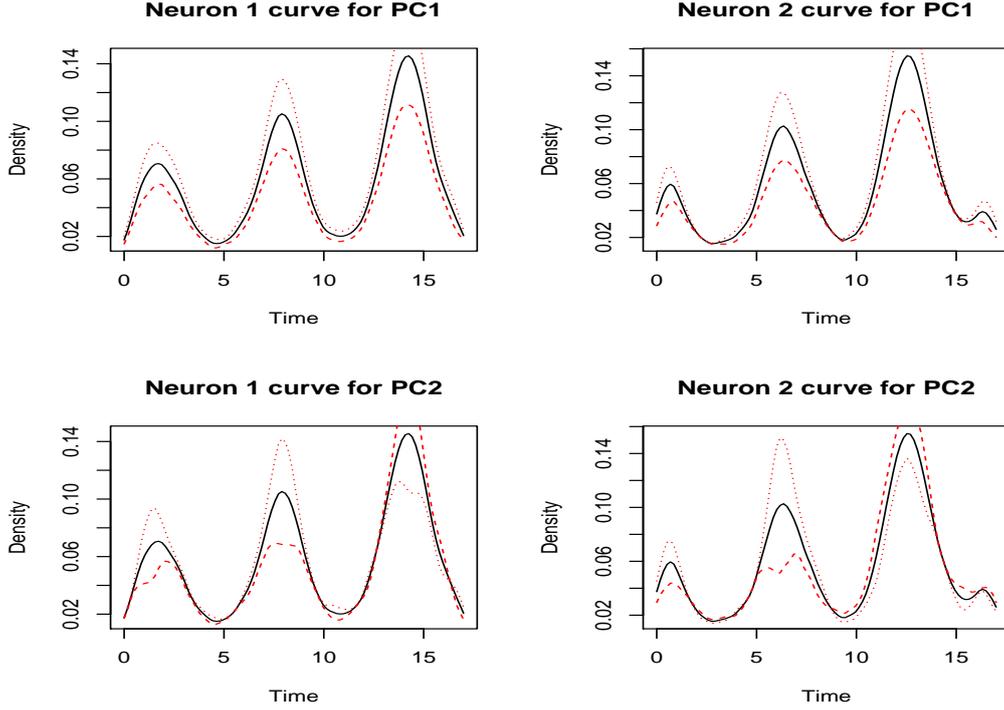


Figure 5.2: Intensity function of the true and simulated neuron and Kolmogrov-smirnov plot.

eigen functions for the first neuron $\xi_1^1 = -\cos(\pi t/10)/\sqrt{5}$, and $\xi_2^1 = \sin(\pi t/10)/\sqrt{5}$, $0 \leq t \leq 20$, and two eigen functions for the second neuron $\xi_1^2 = +\cos(\pi t/5)/\sqrt{3}$, and $\xi_2^2 = -\sin(\pi t/5)/\sqrt{3}$, $0 \leq t \leq 20$. The eigen value were chosen as $\rho_1^1 = 1$ and $\rho_2^1 = 0.8$ for the first neuron and $\rho_1^2 = 0.1$ and $\rho_2^2 = 0.3$ for the second neuron, then the covariance function can be represented in terms of eigenfunctions ξ_k and eigenvalues ρ_k

$$v(s, t) = \sum \rho_k \xi_k(s) \xi_k(t)$$

Then we performed the bivariate functional principal component analysis, the true intensity function and the simulated intensity function are shown on the upper panel in Figure 5.1. To have small variation of two functions, the intensity seems similar. The black solid lines displays true intensities and the red dotted lines show the simulated neuron. The mean of true and simulated intensity functions are displayed as sky-blue dashed line and yellow dashed line respectively. By comparing mean intensity functions, the simulation seems work well. The simulated neuron is more likely to have large variation than the true one, but the kolmogrov-smirnov plot shows that all estimates are significant.

The best way to display the FPCA result is plotting the principal component as perturbations of the mean and it is sufficient to consider individual ξ_i^1 and ξ_i^2 separately. Figure 5.2 displays the first two principal components for neuron 1 and 2 with its mean density function.

Since $\|\xi_i^1\|^2 + \|\xi_i^2\|^2 = 1$ by the definition, calculating $\|\xi_i^1\|^2$ gives the proportion of the variability accounted for by the i -th principal component in the neuron 1 curve. For the first principal components, the contributions of neuron 1 and neuron 2 are 50.01% and 49.99% respectively. Therefore both neuron 1 and 2 are important and so there is very substantial interaction between the two neurons. However for the second principal components, the measure indicates that 77.88% of the variation is due to the neuron 1 curves. The effect on the neuron 1 curves of the first combined principal component of variation is virtually identical to the first principal component curve extracted from the neuron 1 curves considered alone.

5.3.2 Scenario 2

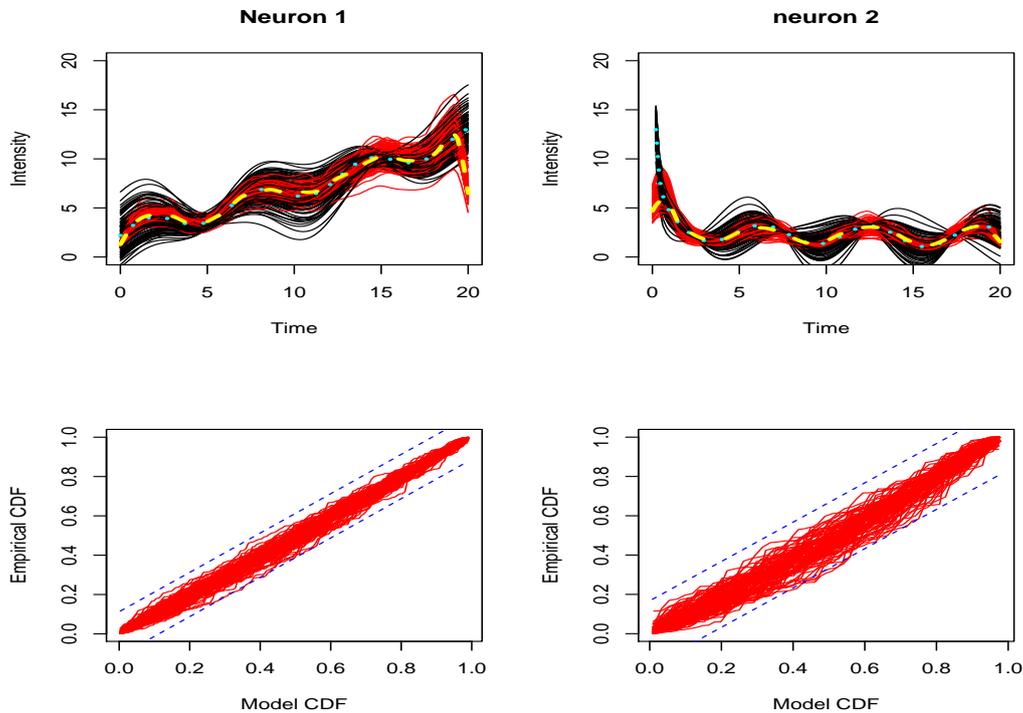


Figure 5.3: Intensity function of the true and simulated neuron and Kolmogorov-smirnov plot.

In this scenario, we designed the simulation to check the performance of multivariate functional principal component analysis with large variabilities on random functions. We

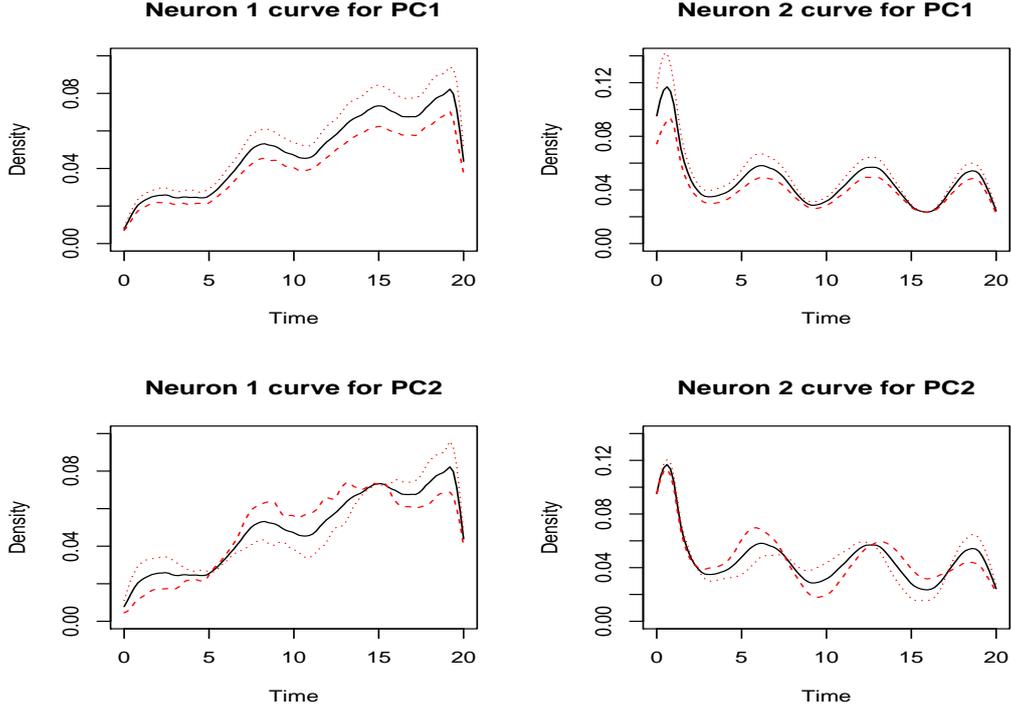


Figure 5.4: Intensity function of the true and simulated neuron and Kolmogrov-smirnov plot.

construct 100 true intensity functions $\lambda_i^r(t), i = 1, \dots, 100, r = 1, 2$. The true intensity function has form of the number of events multiplied by their density function $\lambda_i^r(t) = N_i^r f_i^r(t)$. The density function was constructed by the Karhunen-Loeve representation,

$$f_i(t) = f_\mu^r(t) + \sum_{k=1}^k \varphi_{ik}^r \xi_k^r(t)$$

where the mean density function $f_\mu(t) = 2 + \frac{t}{2} + \sin(t)$, two eigen functions for the first neuron $\xi_1^1 = -\cos(\pi t/10)/\sqrt{5}$, and $\xi_2^1 = \sin(\pi t/10)/\sqrt{5}, 0 \leq t \leq 20$, and two eigen functions for the second neuron $\xi_1^2 = +0.3\cos(\pi t/5)/\sqrt{3}$, and $\xi_2^2 = -0.3\sin(\pi t/5)/\sqrt{3}, 0 \leq t \leq 20$. The eigen values were chosen as $\rho_1^1 = 4$ and $\rho_2^1 = 1$ for the first neuron and $\rho_1^2 = 6$ and $\rho_2^2 = 2$ for the second neuron. Then we performed the bivariate FPCA, the true intensity function and the simulated intensity function are shown on the upper panel in Figure 5.3. The function for neuron 1 is a increasing function with $\sin(t)$ and the function for neuron 2 is a decreasing function of $\cos(t)$. It has a larger variability than scenario 1. The black solid lines displays true intensities and the red dotted lines show the simulated neuron. The mean of true and simulated intensity functions are displayed as skyblue dashed line and yellow dashed line respectively. By comparing mean intensity

functions, the simulation seems work well. The simulated neuron more likely to have large variation than true, but the kolmogrov-smirnov plot shows that all estimates are significant.

Figure 5.4 displays the first two principal components for neuron 1 and 2 with its mean density function. For the first principal components, the contributions of neuron 1 and neuron 2 are 49.19% and 50.81% respectively. The contribution of both neuron 1 and 2 are important and so there is very substantial interaction between the two neurons. In addition, for the second principal components, the measure indicates that 42.38% of the variation is due to the neuron 1 curves and 57.62 due to the neuron 2. Also the second principal component has similar contribution as the first principal component.

5.4 Application to data

The main purpose of the study is to characterise the multiple neuron activities which were collected in the different regions. The neuron data set were collected in the brain regions EC3 and EC5, so we performed the bivariate functional principal component analysis and the intensity functions for each neuron are displayed in Figure 5.5. The estimated intensity functions from the bivariate FPCA seems to have more fluctuation than univariate FPCA, and generally the brain region EC 3 has larger variability than the region EC 5. Although bandwidth for the kernel was selected by optimal bandwidth with plug-in method, the intensity functions seems over-estimated. Compared with the univariate FPCA estimation, the region EC 3 has similar trend but the variability of EC 5 increased with several peaks. Moreover the details about the effect of the principal components are shown in Figure 5.6. These displays the bivariate FPCA results with plotting the principal component as perturbations of the mean density function. The first principal component for the Mwheel task explains 42.75% of the total variability, and the contribution of neurons from the regions EC 3 and EC 5 are 13.53% and 86.47%, respectively. This is the main factor of large variation on the EC 5 in Figure 5.5. The second and third principal component capture 18.42% and 11.72% proportion of total variation respectively. The second principal component may represent on the larger variability for bivariate FPCA than for univariate FPCA. The proportion of the variability and the proportion of contribution are described in Table 5.1

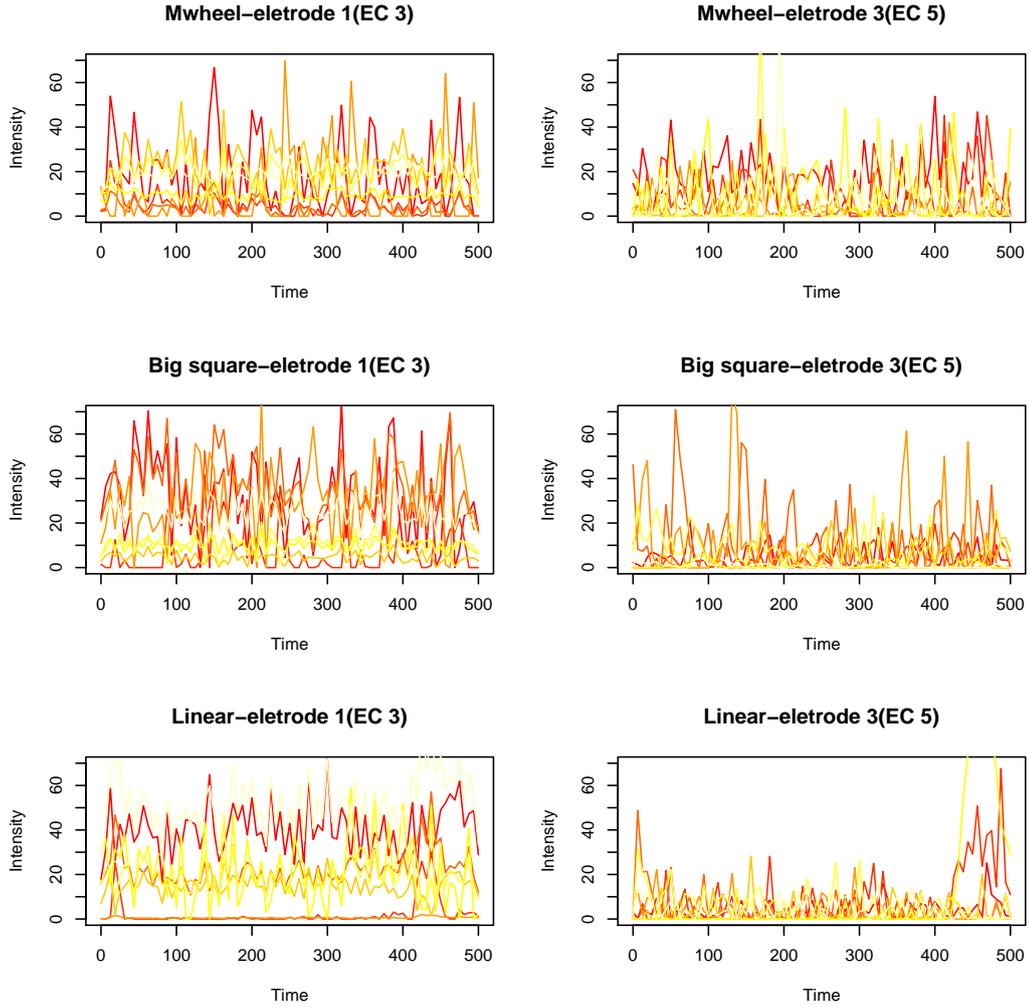


Figure 5.5: The individual neuron firing rate function.

The Kolmogrov-Smirnov plots of the bivariate FPCA for Big-square task are displayed in Figure 5.7. The black solid line indicates the univariate FPCA and the red line show the bivariate FPCA. Most of K-S statistics describes bivariate FPCA works better than univariate one. However it does not work well.

5.5 Discussion and conclusion

We applied the multivariate functional principal component method on the bivariate repeated point process neuron data in the simulation study and one may say that it is a useful method to characterise the intensity functions for individual objects and to understand how two neurons vary jointly in the total system. In the model checking procedure, all two different scenarios have good Kolmogrov-Smirnov statistics.

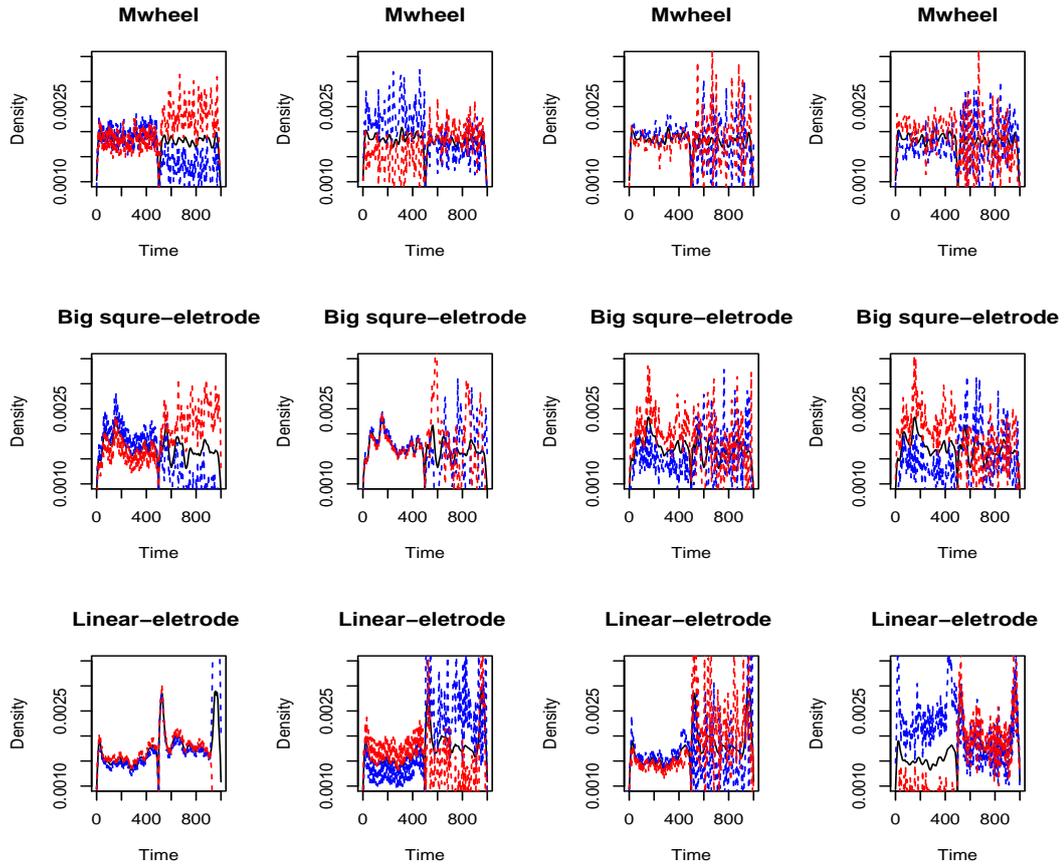


Figure 5.6: The first four principal component curves of the neuron firing.

In the result section, there are the same problems with the discrepancy of the number of neuron spike trains in each trial as in Chapter 4. The intensity function was defined as multiple of the number of events on their density function (4.3). It is quite difficult to analyse the characterised intensity functions, since they are too fluctuated and it is hard to find similar patterns. But there seems to be some different pattern on the region. At this stage, we can only find that there is an opposite trend of firing neuron spike, if region EC3 fire many number of spike train, there are only few number of spike trains are fired. In addition, most of proportion for the contribution of two region was dominated by one region. Only fourth principal component on Big-square task had a similar contribution. According to K-S plot there is a significant improvement from the univariate FPCA, but it still gave a bad estimation.

Table 5.1: Proportion of principal component and the proportion of contribution of each region.

Mwh	PC 1		PC 2		PC 3		PC 4	
	42.75%		18.44%		11.72%		6.51%	
	EC 3	EC 5						
	13.53%	86.47%	78.41%	21.59%	5.74%	94.26%	20.47%	79.53%
B-S	PC 1		PC 2		PC 3		PC 4	
	67.82%		12.25%		4.93%		4.29%	
	EC 3	EC 5						
	11.89%	88.11%	1.12%	98.88%	32.54%	67.46%	50.48%	49.52%
Lin	PC 1		PC 2		PC 3		PC 4	
	99.00%		0.56%		0.01%		0.01%	
	EC 3	EC 5						
	0.72%	99.28%	7.92%	92.08%	2.11%	97.89%	82.97%	17.03%

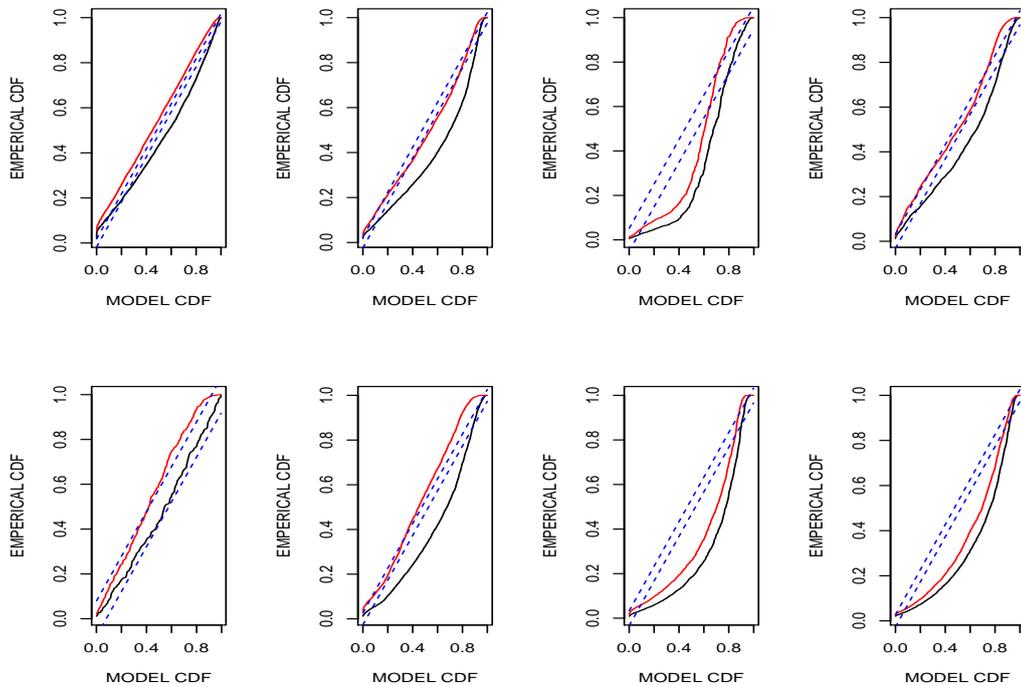


Figure 5.7: The Kolmogorov-Smirnov plot for the task on the Bigsquare.

Chapter 6

Case study - The comparison of three approaches

6.1 Motivation and data description

In the previous chapter, we applied the parametric modelling and the non-parametric modelling to estimate a single intensity function, and the multiple principal component analysis to investigate the influence between the different tasks on each neuron. However the data set had some problems every time when we applied new approaches, hence it may be difficult to compare the improvements of new approaches.

The main aim of this chapter is to compare or check how the analysis improves with different approaches with the same data set. The data set was collected from the rat in the maze session, which is 180 cm x 180 cm size of square platform. The reward was the water and the duration was 90.9 minutes. One of the raw data sets given the time of each spike in 20 MHz and they were rounded to 1 ms bins. Another raw data set is a corresponding file containing cluster information for each spike.

In chapter 2 and 3, the data set contained spike trains where inter-arrival time were less than 2 ms and some spike trains occurred at the same time. This had not met the condition of the spike train which was explained in chapter 1, as the time interval we consider becomes sufficiently small so that no more events occur jointly in any bin. In this data set, the eight spike trains are N_1 , N_2 , N_3 , N_4 , N_5 , N_6 , N_7 and N_8 , there are no spikes with an inter-arrival time of less than 2 ms, and no spike occurred at the same time. Figure 6.1 top left panel shows the raster plot of the eight neurons over 3

seconds on the session EC 582. It can be seen that there are different spike trends, and frequencies are also different. N_6 , N_7 and N_8 seems to have much higher frequency than others. In the top right panel on Figure 6.1, all trials have high frequency whereas the trial EC 430 has a small frequency. It is difficult to find the pattern of spike train by visualisation due to high frequency.

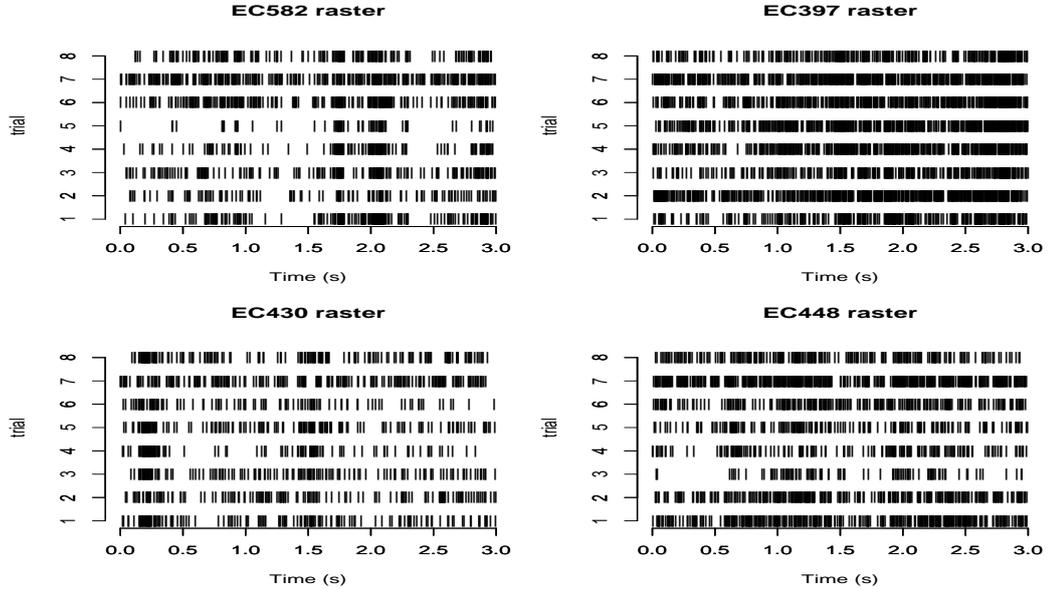


Figure 6.1: Spike of eight neurons within 1 ms bin in 3 seconds .

Figure 6.2 shows the raster plot of the same data set but it increases the bin size to 5ms. In general, the absolute refractory period of the neuron firing is a 1 ms and the relative refractory period is up to 5 ms. To investigate how much different on neuron activity with 5 ms binned data compare to 1 ms binned data, the same analysis was performed twice with different bin size of data. However it is very difficult to find the difference between 1 ms and 5 ms data set.

Figure 6.3 to 6.6 show the histograms of inter-spike times for eight neurons for each sessions and the summary of inter-spike times are described in Table 6.2 to 6.4. The scale of histogram is log scale. Because most of the inter-spike times are less than 0.1 seconds (99%), and they show an exponential decay. All the inter-spike distributions seems similar, only the 8th trial seems to have a heavier tail. In general, it is very difficult to find any different patterns of neuron activities in terms of different sessions and trials. One of the simple approaches to investigate the neuron firing activities is kernel density estimation.

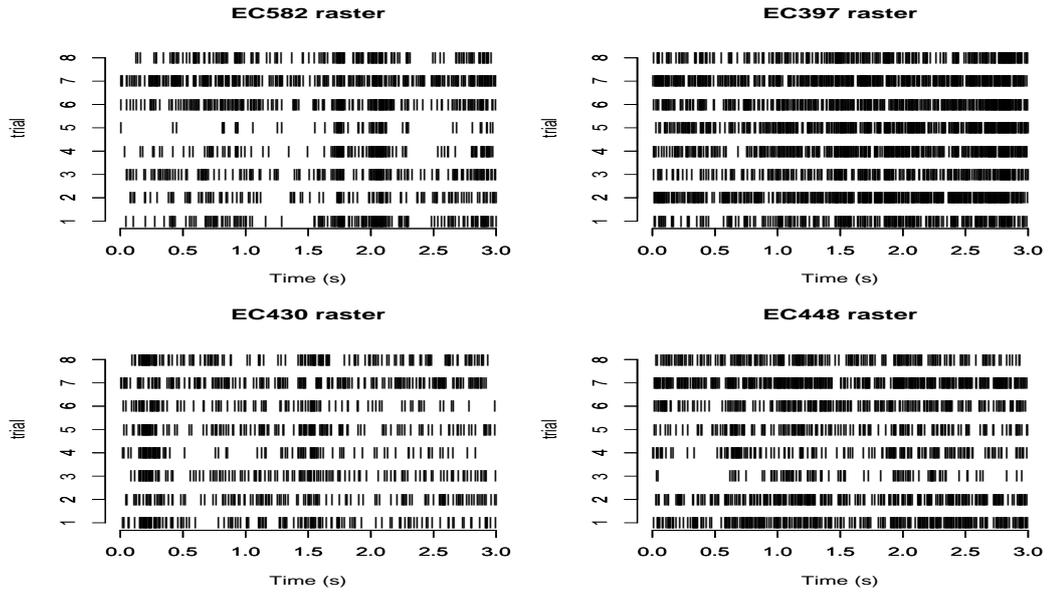


Figure 6.2: Spike of eight neurons within 5 ms bin in 3 seconds .

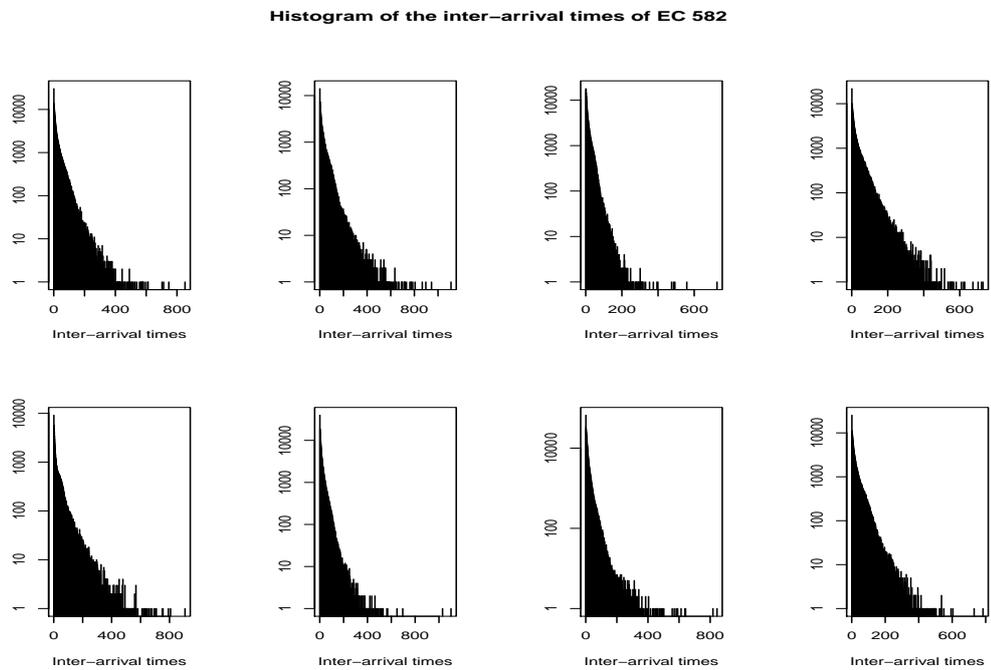


Figure 6.3: Histograms of the inter-spike time for eight neurons of EC582.

Histogram of the inter-arrival times of EC 397

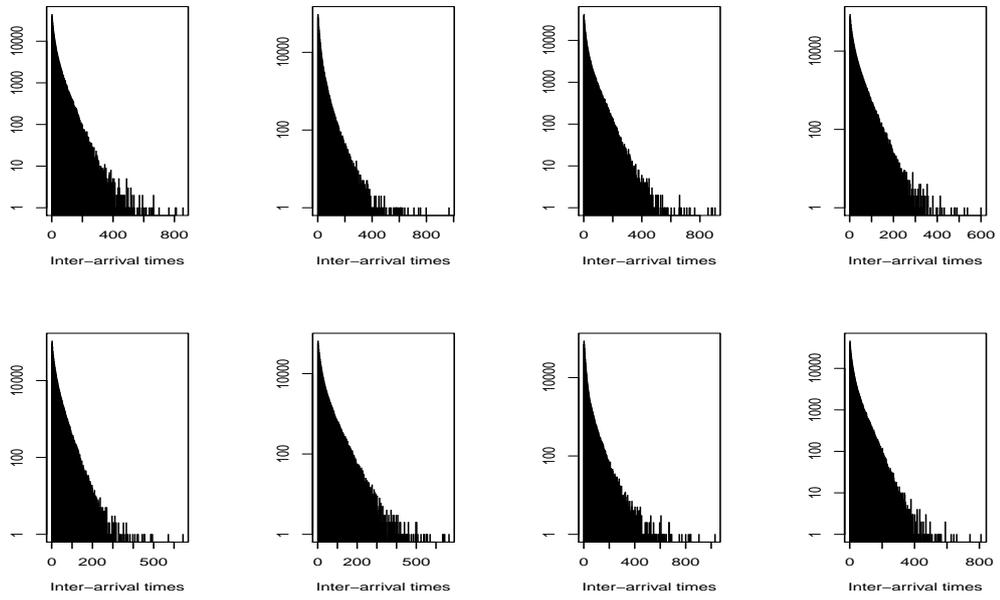


Figure 6.4: Histograms of the inter-spike time for eight neurons of EC397.

Histogram of the inter-arrival times of EC 430

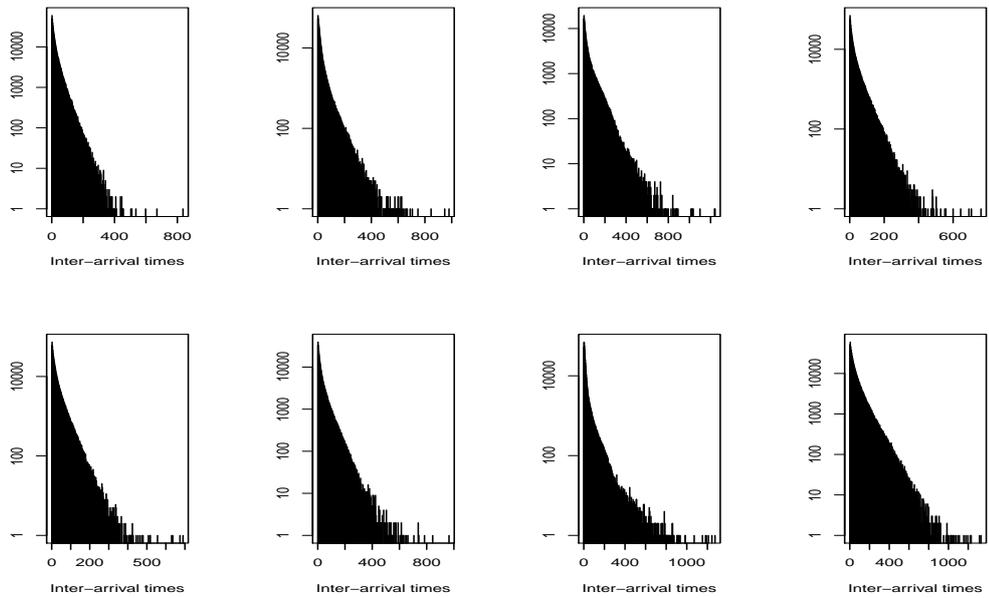


Figure 6.5: Histograms of the inter-spike time for eight neurons of EC430.

Histogram of the inter-arrival times of EC 448

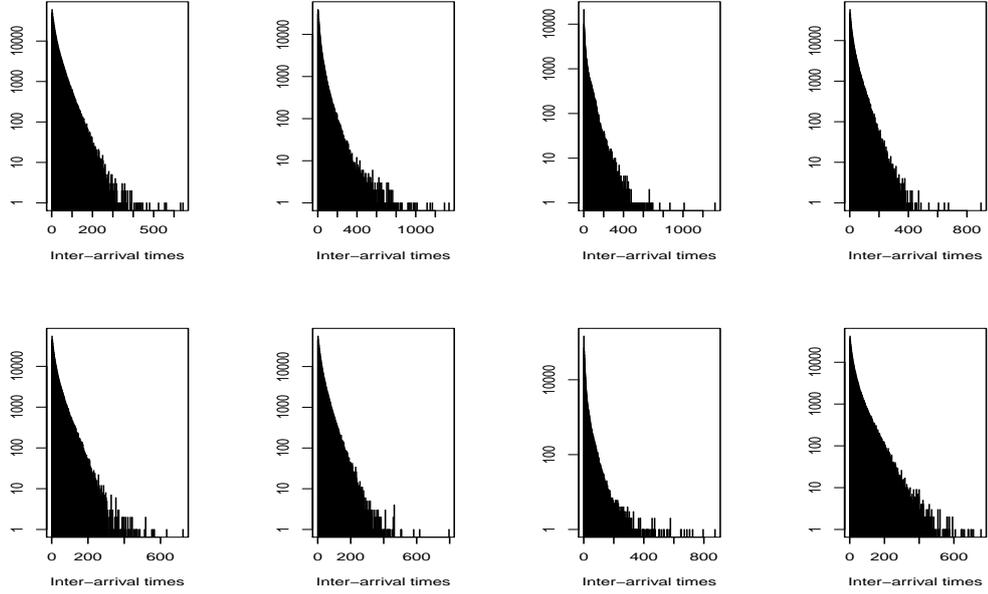


Figure 6.6: Histograms of the inter-spike time for eight neurons of EC448.

Table 6.1: Summary of the inter-spike time for the EC582

	N 1	N 2	N 3	N 4	N 5	N 6	N 7	N 8
0%	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008
1%	0.0009	0.0010	0.0009	0.0009	0.0010	0.0009	0.0009	0.0009
5%	0.0013	0.0017	0.0015	0.0014	0.0014	0.0013	0.0012	0.0014
10%	0.0019	0.0028	0.0025	0.0022	0.0023	0.0019	0.0017	0.0021
25%	0.0044	0.0063	0.0059	0.0051	0.0070	0.0041	0.0034	0.0050
50%	0.0116	0.0174	0.0154	0.0137	0.0250	0.0098	0.0075	0.0129
75%	0.0307	0.0433	0.0388	0.0368	0.0843	0.0241	0.0161	0.0334
90%	0.0642	0.0851	0.0772	0.0757	0.1659	0.0496	0.0322	0.0691
95%	0.0900	0.1153	0.1038	0.1066	0.2480	0.0714	0.0473	0.0954
99%	0.1604	0.2097	0.1744	0.1932	0.4812	0.1224	0.0909	0.1613
mean	0.0246	0.0336	0.0298	0.0292	0.0644	0.0196	0.0138	0.0263
n	221921	162451	182986	186776	84617	278332	395883	207189

Table 6.2: Summary of the inter-spike time for the EC397

	N 1	N 2	N 3	N 4	N 5	N 6	N 7	N 8
0%	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008
1%	0.0009	0.0009	0.0009	0.0009	0.0009	0.0009	0.0009	0.0009
5%	0.0011	0.0010	0.0011	0.0010	0.0010	0.0011	0.0011	0.0011
10%	0.0015	0.0013	0.0014	0.0013	0.0013	0.0014	0.0014	0.0014
25%	0.0029	0.0023	0.0028	0.0023	0.0021	0.0025	0.0024	0.0028
50%	0.0070	0.0047	0.0071	0.0050	0.0045	0.0057	0.0047	0.0069
75%	0.0165	0.0098	0.0177	0.0113	0.0099	0.0128	0.0099	0.0166
90%	0.0337	0.0195	0.0376	0.0224	0.0190	0.0263	0.0206	0.0343
95%	0.0490	0.0294	0.0551	0.0322	0.0272	0.0388	0.0331	0.0503
99%	0.0874	0.0583	0.0974	0.0576	0.0488	0.0722	0.0698	0.0883
mean	0.0136	0.0085	0.0147	0.0092	0.0080	0.0108	0.0092	0.0136
n	401378	638156	371794	589203	677431	502968	594696	399565

Table 6.3: Summary of the cleaned inter-spike time for the EC582

	N 1	N 2	N 3	N 4	N 5	N 6	N 7	N 8
0%	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010
10%	0.0020	0.0030	0.0030	0.0020	0.0020	0.0020	0.0020	0.0020
25%	0.0040	0.0060	0.0060	0.0050	0.0070	0.0040	0.0030	0.0050
50%	0.0120	0.0170	0.0150	0.0140	0.0250	0.0100	0.0080	0.0130
75%	0.0310	0.0430	0.0390	0.0370	0.0840	0.0240	0.0160	0.0330
90%	0.0640	0.0850	0.0770	0.0760	0.1660	0.0500	0.0320	0.0690
95%	0.0910	0.1150	0.1040	0.1070	0.2480	0.0720	0.0470	0.0950
99%	0.1600	0.2100	0.1740	0.1930	0.4810	0.1230	0.0910	0.1610
mean	0.0246	0.0336	0.0298	0.0292	0.0645	0.0196	0.0138	0.0263
n	221564	162269	182796	186552	84525	277862	395033	206910

Table 6.4: Summary of the cleaned inter-spike time for the EC397

	N 1	N 2	N 3	N 4	N 5	N 6	N 7	N 8
0%	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010
10%	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010
25%	0.0030	0.0020	0.0030	0.0020	0.0020	0.0030	0.0020	0.0030
50%	0.0070	0.0050	0.0070	0.0050	0.0050	0.0060	0.0050	0.0070
75%	0.0170	0.0100	0.0180	0.0110	0.0100	0.0130	0.0100	0.0170
90%	0.0340	0.0200	0.0380	0.0220	0.0190	0.0260	0.0210	0.0340
95%	0.0490	0.0290	0.0550	0.0320	0.0270	0.0390	0.0330	0.0500
99%	0.0870	0.0580	0.0970	0.0580	0.0490	0.0720	0.0700	0.0880
mean	0.0136	0.0086	0.0147	0.0093	0.0081	0.0109	0.0092	0.0137
n	400322	635970	370760	587328	675067	501512	592767	398576

The individual firing rate for the eight neurons on each sessions are shown in Figure 6.7. These intensity functions are estimated by Epanechnikov kernel estimation. As shown in the raster plot 6.1, the lowest frequency was on the session EC 582, so the intensity function on EC 582 has the lowest value. The intensity function is highest for the session EC 397, and for EC430 and EC448 it is similar. According to the individual kernel density estimation, it is hard to find any similarity and dissimilarity of neuron activities on the session. One of the reasons is that the simple kernel density estimation shows the individual pattern of the activity and the other possible reason is that the time scale is too big. So this intensity function shows only the global effect of the single neuron firing rate.

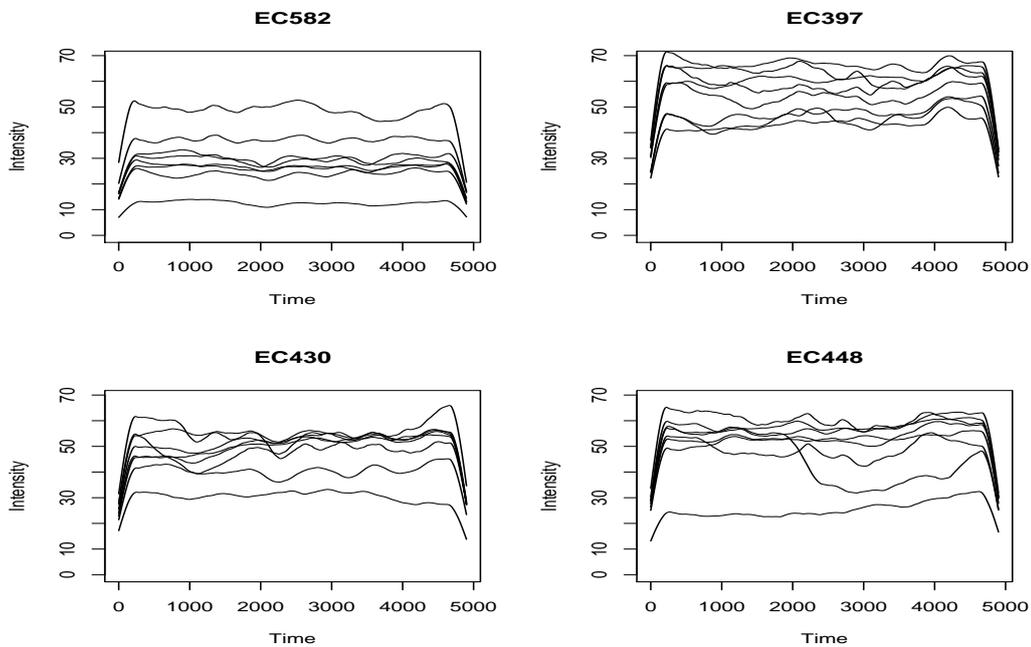


Figure 6.7: The individual intensity functions of four sessions by kernel density estimation.

6.2 Application to data.

The main purpose of this section is to compare the estimation on the same data set with different approaches such as parametric point processes, non-parametric point processes and functional principal component analysis. In the previous section, we performed the kernel estimation, it gives the simple idea of characterising the single neuron firing rates. It was difficult to find any connection between sessions or trials. The aim of the parametric point process to characterising single neuron activities and self-exciting process gives the effect of history dependence.

6.2.1 Characterising single neuron activities by non-parametric modelling

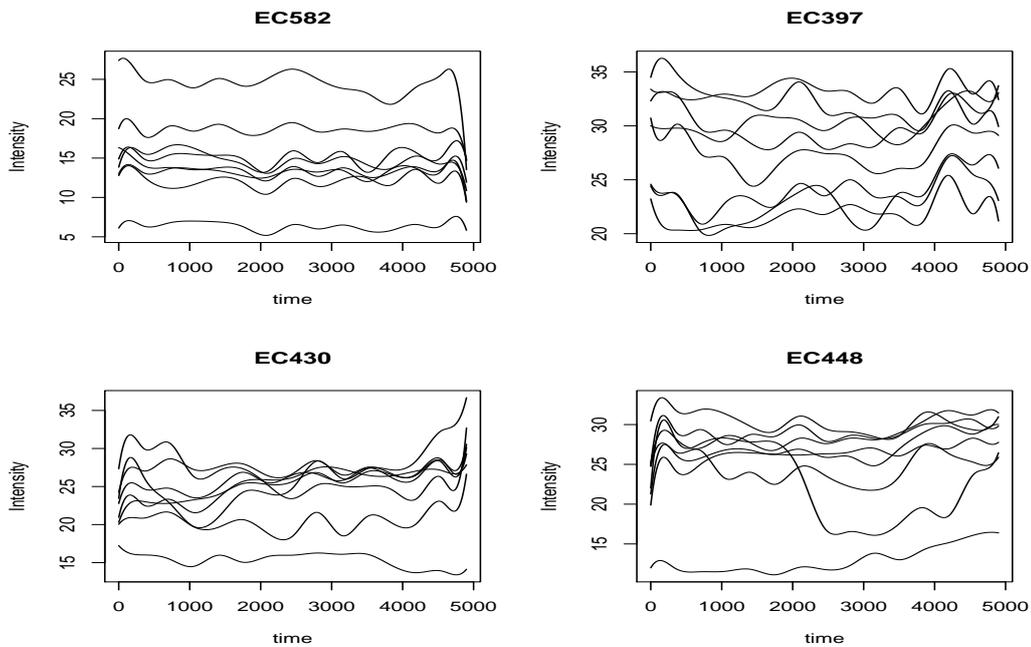


Figure 6.8: The individual intensity functions of four sessions by bspline basis function.

The smoothing spline has been applied with the B-spline basis function which was introduced in chapter 3. The knot sequence of the B-spline basis is determined by Bayesian information criterion. Figure 6.8 shows the intensity function of single neuron activity by trials. Its pattern of intensity function is quite similar compared to the intensity function in Figure 6.7, but the intensity function by smoothing spline using B-spline seems to be more clear. In the session EC 582, the main variation of the intensity

function seems the number of firings. Most intensity functions have a similar trend which fluctuates regularly at the same time, only the different intensity is outstanding. There is also variation in the different number of firing spike trains in session EC 397. Most of trials have a decreasing trend at the beginning of the experiment, from 0 to 1000, then the intensity stay stable with some fluctuations then increase. However it is not clear in session EC 582, since the time points which have high intensity are all different. There seems to be an outlier in the session EC 430 and EC 448, most of the intensities lie in between 20 to 30, some trial's intensities are lower than 20. They also have variation in the different number of neuron firing spike trains, but those variations seems to be lower than for EC 582 and EC 397. Some of the intensities have very different pattern.

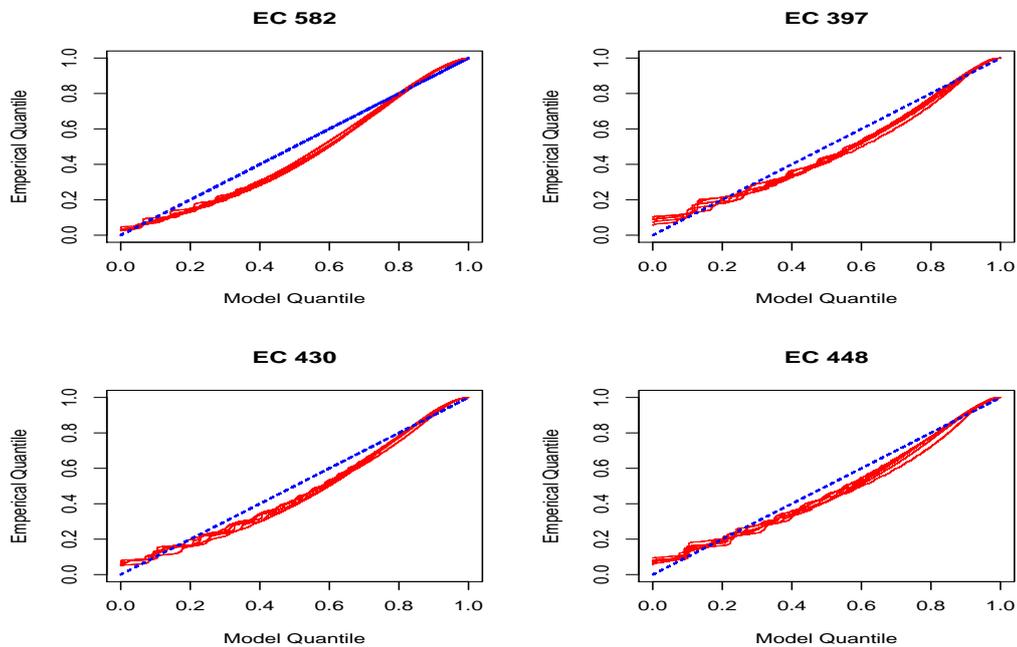


Figure 6.9: The Kolmogorov-smirnov plot of four sessions by bspline basis function.

The results of the Kolmogorov-Smirnov plot in Figure 6.9 tells characterising the single neuron activities are not very successful. Most of the K-S statistics are below 45 degree line. One of the possible reasons is that the intensity functions do not contain the history dependence and the correlation between other neurons. It might be very difficult to have a good estimation for characterising the neuron intensity functions without those effects. In addition, the limitation of this approach is that it can only represent the single neuron firing rate and one of the important mechanism of a neuron network, which is the communication, was omitted.

6.2.2 Characterising neuron activities using functional data analysis

In previous section, we have seen the intensity function in Figure 6.8 and we tried to find the similarity or dissimilarity of the intensity function between the session. The only thing we found was the variation in the number of neuron firings, because that was too obvious. In this section, we will find the other variation and how it affects the model by principal component analysis. First of all, Figure 6.10 shows the principal component curves of the session EC 582. The first principal component curve ξ is dominant which percentage of variability being more than 99%. The first principal component curve ξ_1 is positive throughout the times, and time at 500, 2000 to 3000 and 4500 seconds have a large variability. Moreover the percentage 99.1% indicates that this type of variation strongly dominates all other type of variations. The second principal component curve ξ_2 account for 0.4% of the total variation so we cannot expect ξ_2 to be as important as the curve ξ_1 . It has a negative contribution at where the ξ_1 had low variability. The third and fourth component curves also represent very small percentages of variability and they are difficult to interpret.

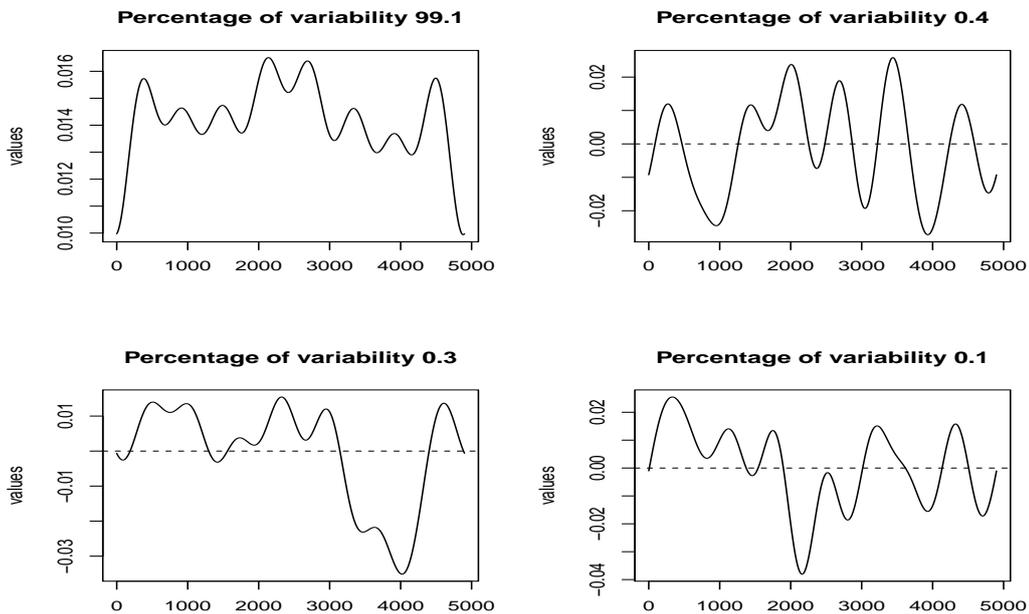


Figure 6.10: The first four principal component curves of the EC582.

The principal component curve ξ_1 on the session EC 397 also strongly dominates all other curves with percentage 94.8%. It has all positive contribution with decreasing

trend, which reflects that at the beginning of the experiments there are large variations, then it decreases. Second component curve ξ_2 contributes very little, it has 2.7% of the variability. In the period between 0 and 1000 it is positive, then most of the rest is negative, it may emphasize the explanation of ξ_1 . The ξ_3 and ξ_4 also account for very small proportions of the variation since they have to be orthogonal to ξ_1 and ξ_2 . The third ξ_3 shows large variation between 0 to 3500 and the fourth ξ_4 represents low variation at the middle of the experiments.

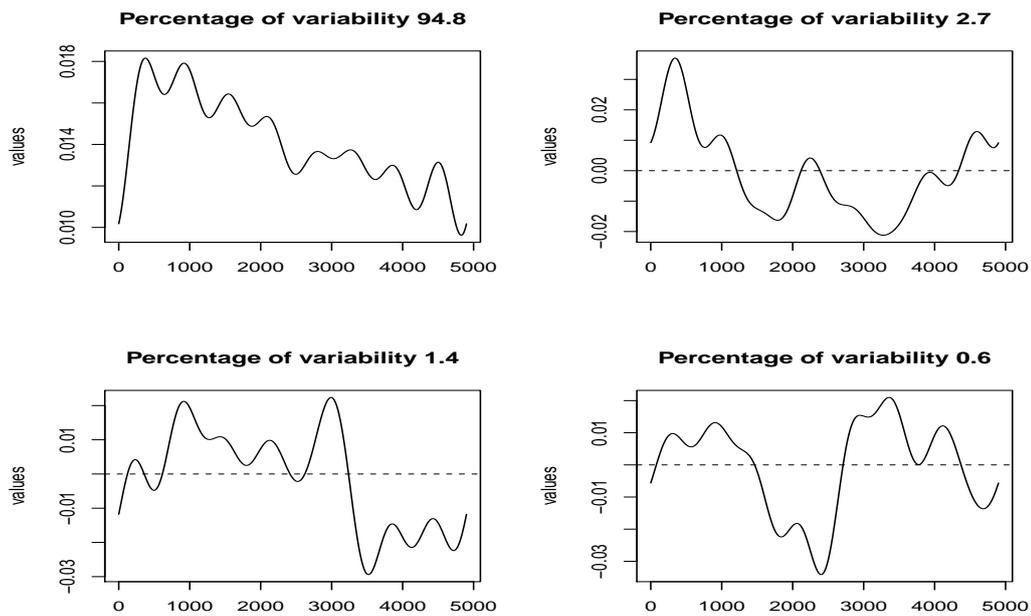


Figure 6.11: The first four principal component curves of the EC397.

The principal component curve for the session EC 430 has similar pattern as for EC 397. The top left panel Figure ?? in Appendix shows that ξ_1 has an increasing trend with big fluctuation at the middle of experiments explaining 85.6%. The second PC curve accounts for 9.9% which is a higher proportion than for the previous two sessions and it describes that contribution to variation between first half and second half are opposite. There are similar patterns in the session EC 448. At the beginning, the middle and the end of the experiments, there seem to be large variation in the first PC which contributes 88.6%. And the effect of ξ_2 is very close to ξ_2 for EC 430. Figure ?? is placed in the Appendix.

According to principal component curves, all in cases the first PC curve ξ_1 is dominant with more than 85% proportions of explained variability. In addition, all ξ_1 consist

of a positive contribution, which supports that there are huge variations in the shape of each ξ_1 for all the sessions. Only two ξ_2 show an effective contribution to the model which are EC 430 and EC 448 with 9.9% and 9.8%, respectively. Those effect could be seen for other sessions too.

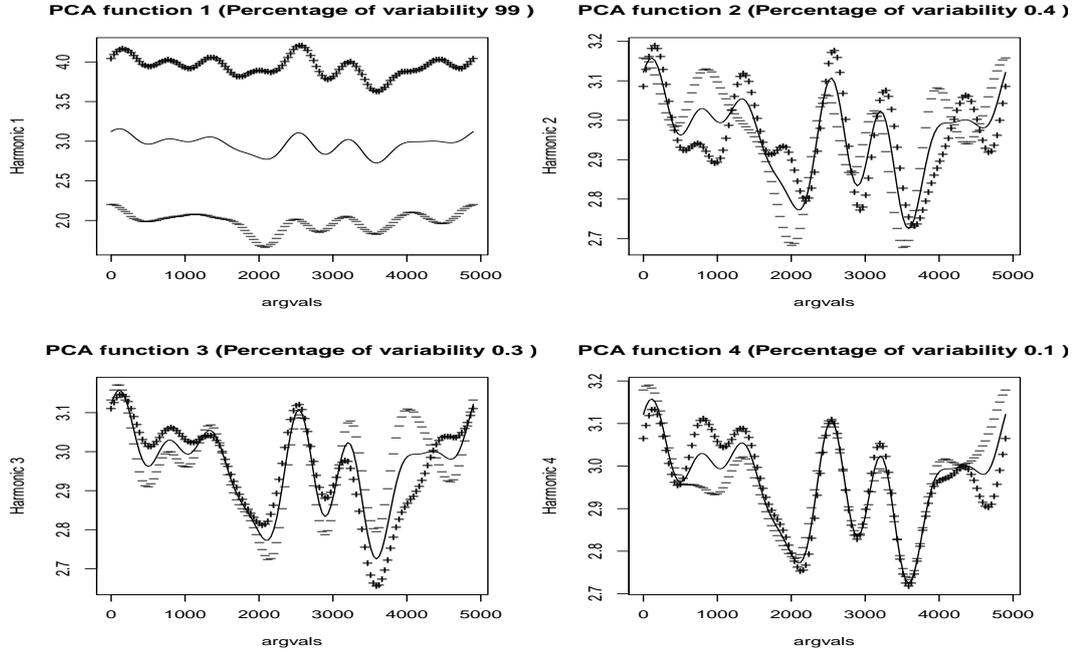


Figure 6.12: The mean functions of the EC582 with four principal component curves.

The effects of the principal component curves are shown in Figure 6.12 by adding and subtracting each PC curve on the mean function. Figure for other session were plotted in the Appendix. As we saw the first PC curve ξ_1 , a positive contribution which indicates the different number of neuron firing spike trains. It is reflected in the individual intensity functions which contain the principal component effects.

The individual intensities of the neuron firing in the session EC 582 are demonstrated in Figure 6.13. These individual intensities contains four principal components, because of the dominance of mode ξ_1 there is an obvious difference in the number of neuron firings in the trials. The intensity functions for EC 582 seem to have a lower firing rate rather than the other sessions and have only gentle fluctuations. Likewise the session EC 397, the proportion of the first PC curve ξ_1 was about 95% hence the major mode is the variation of the number of events. Other effects seem hard to find. The session EC 430 has similar pattern as the session EC 582 but a higher firing rate but there seem to have simple fluctuation. In addition, the variation of the number of events seems less than

other sessions. Only one trial has lower intensity. There are also outliers in the session for EC 448. Except two intensity function, all of them have a very similar pattern.

One of the important aspects of the principal component analysis is the examination of the scores of each PC curve on each component. Figure 6.14 displays the first two PC scores, four different colours were used such as black, red, green and blue which indicate four different sessions EC 582, EC 397, EC 430 and EC 448 respectively. According to the individual intensities in Figure 6.13, we suspect that there are some outliers in the session EC 430 and EC 448. There are obvious outliers which numbers are 3 and 8 in the session EC 448. They might be considered as the two lower intensity functions in the bottom right of Figure 6.13. In addition, there are four more potential outliers, which is not very clear, the 5 and 7 with black colour and 3 and 7 with green colour. It may be consider as the lowest and the highest intensity function on the EC 582 and two lowest intensity on the EC 430. There is no outlier on the EC 397. Moreover there is no cluster effects on the sessions. It means that there is no significant difference between the session, one of the assumption at the beginning of the study we had considered there might be a discrepancy between the sessions. However the scatter plot in Figure 6.14 tells there is no significant discrepancy between the sessions.

The intensity functions were estimated again without outliers as we mentioned. The general patterns are the same as previous one, because the principal components did not have many changes. According to Figure 6.15 the intensity functions from the session EC 582 shows significant difference compared to other sessions. The mean of the intensity function for other sessions stays about 100, but it fluctuates in a range of 40.

While there is some improvement compare to Kolmogorov-Smirnov plot 6.9, it can not be said that the estimation is good enough. However after considering the correlation between other neurons using principal component analysis the K-S statistics is closer to 45 degree line.

6.2.3 Characterising single neuron activities by parametric modelling

We discussed about parametric modelling strategies in Chapter 2, such as Poisson processes, or self exciting processes. In Chapter 2, we applied the Epidemic Type Aftershock Sequence model which was introduced by Ogata[18], [17] on the earthquake data set. In

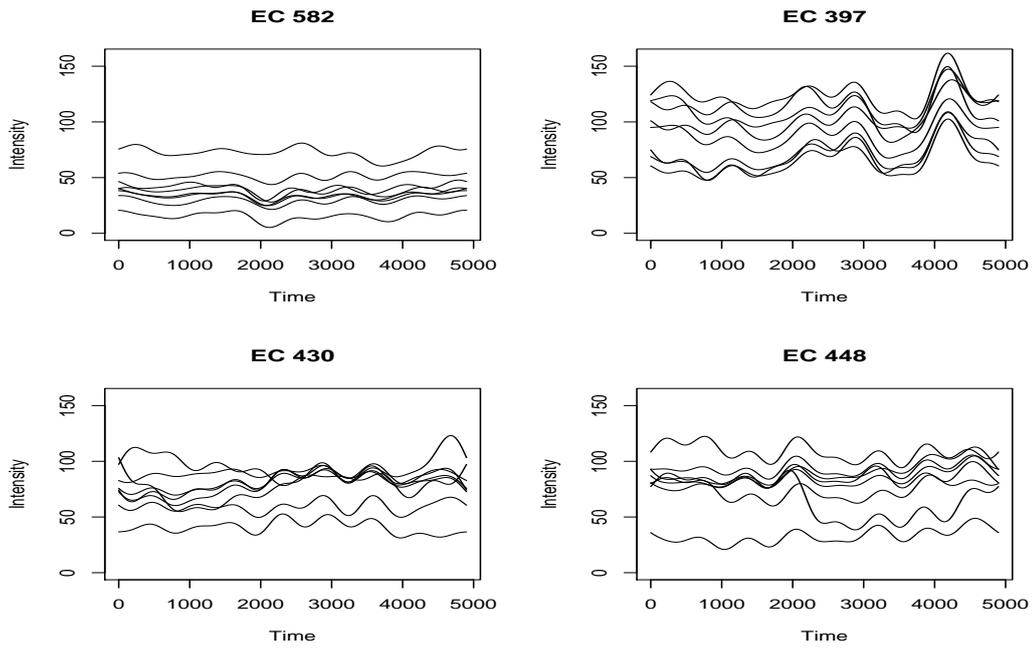


Figure 6.13: The individual intensity functions of the EC582 with four principal component curves.

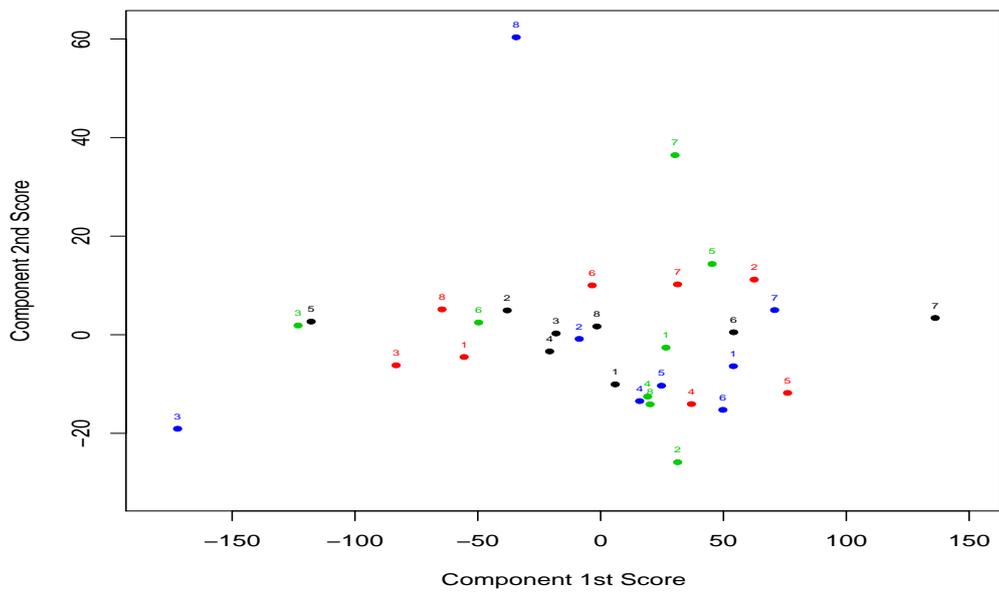


Figure 6.14: The score of the all session and trials on the first two principal components.

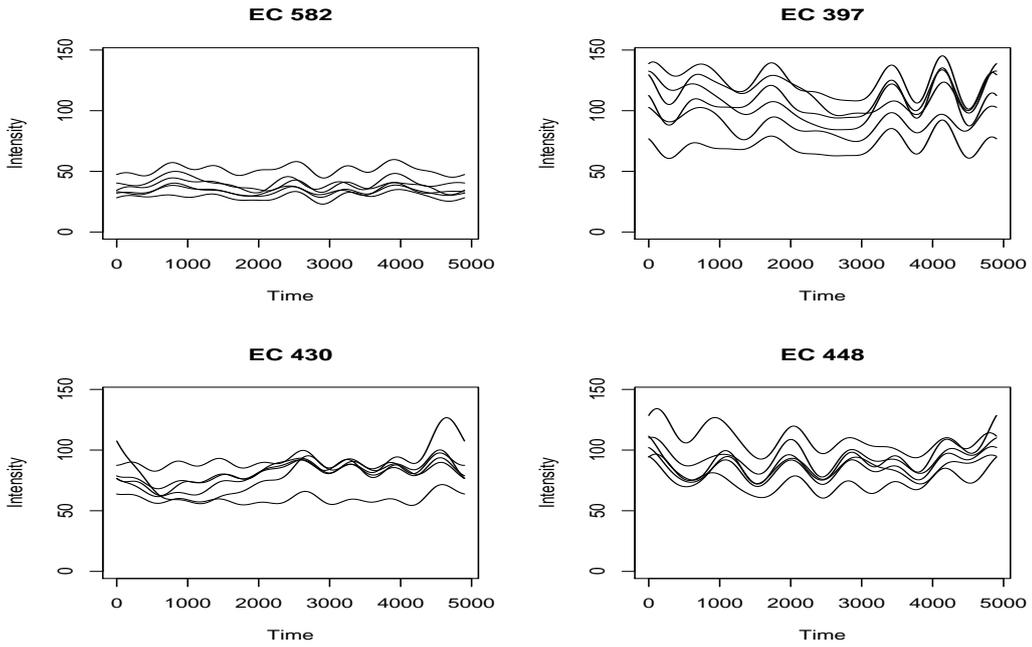


Figure 6.15: The individual intensity functions of the EC582 with four principal component curves after deleting outliers.

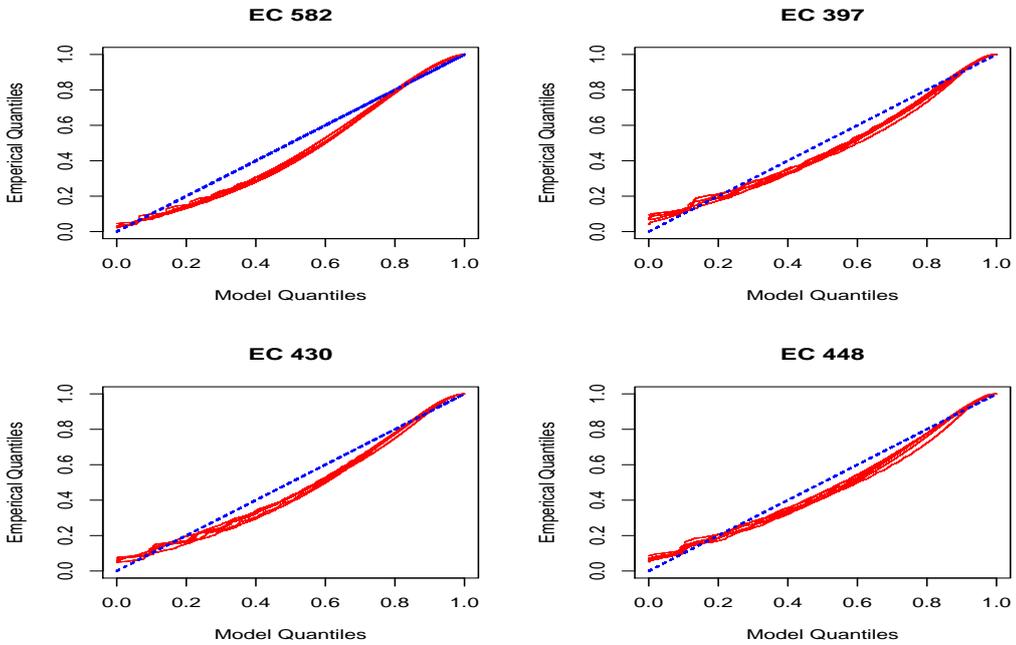


Figure 6.16: The Kolmogorov-smirnov plot of four sessions by bspline basis function.

this section, the self exciting process will be applied which is defined as

$$\lambda(t | H_t) = \lambda_0 + \sum_{t_i < t} \alpha e^{-\beta(t-t_i)}$$

Here, λ_0 is the constant baseline intensity, α is a factor increasing the neuron firing intensity, and β influences the decay function which controls how quickly the overall rate returns to its baseline level λ_0 after an event occurs.

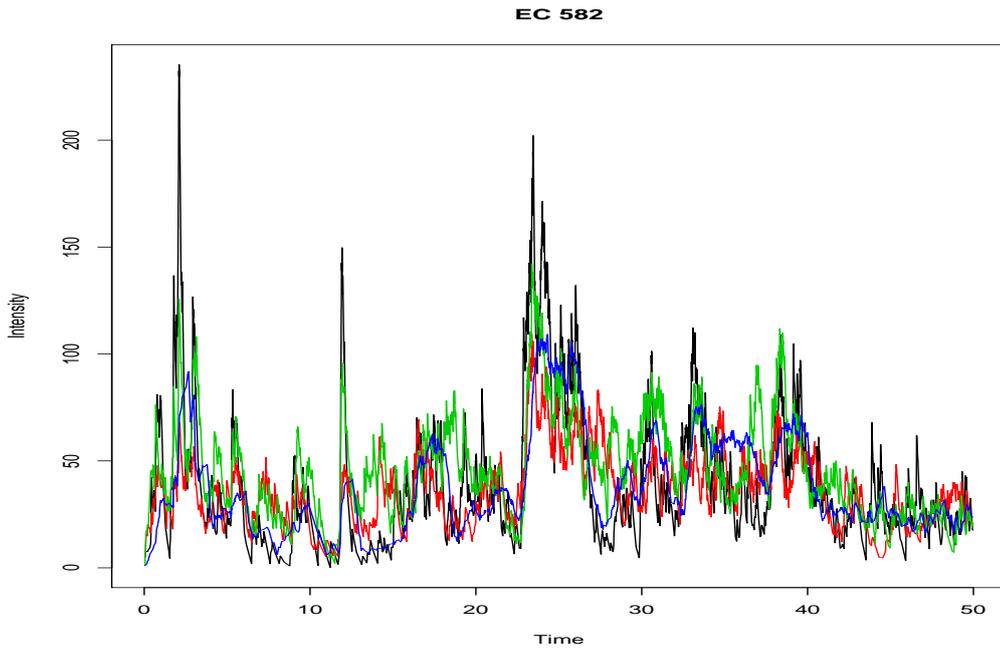


Figure 6.17: The individual intensity functions by Hawkes process.

Before performing the self exciting process on the new data set, we cut the data so that the range of time is 0 to 50 seconds and consider history dependent as time since 5 seconds. The estimated intensity function of the self exciting process is shown in Figure 6.17, each colour represents the trials. The general trend of neuron firing rate seems similar, there are two large peaks in common at times around 2 and 25. After the large peaks it has decreasing trend with small fluctuation. The trial 1 and 3 seem to have more peaks and compared to the result of the functional principal component analysis, the pattern is not exactly the same but it can be said it may have similar results. The reason for the difference may be caused by the history effects because during 1 second there are more than 50 neuron spike train firings.

The comparisons of K-S plots are shown in Figure 6.18. The thick black line indi-

cates the K-S plot of Hawkes process and the red line denote K-S statistics from the FPCA. These K-S plots show that the estimation from the Hawkes process had a better estimation rather than non-parametric method of FPCA. It means that considering a history dependence effect on the modelling might lead to the good estimation.

6.2.4 Characterising single neuron activities by Semi-parametric modelling

According to results from the self-exciting process, the history dependence is very crucial feature in analysis of neuron activities. Parametric modelling can consider the history dependence and it leads to the better estimation of neuron activities. However the computations are too expensive so it was too difficult to estimate the patterns for whole period of data set. On the other hand, non-parametric modelling can take account of whole period of data set but it was hard to add history effect on the model. Hence the model estimation which the history dependence was omitted does not seem to have good estimation. To supplement each limitation of both modellings, the semi-parametric model will be applied.

The main method of this approach is the same of the Hawke's process in self-exciting process. In previous work I use the base intensity as a constance λ_0 , but I will use the base intensity as the intensity function which was estimated by Functional principal component analysis. The background of main assumption are: firstly, adding time varying base intensity function, it may lead to better results than using constant base intensity function. Secondly, it might be reduce the time consuming on the computation. Because in previous work, the optimisation was estimated three parameters λ_0 , α and β . In current approach we only need to estimate α and β so the computation needs will be reduced. The self exciting process for semi-parametric modelling will be applied which is defined as

$$\lambda(t | H_t) = \lambda_t + \sum_{t_i < t} \alpha e^{-\beta(t-t_i)}$$

Here, λ_t is the baseline intensity which was estimated from the functional principal component analysis, α is a factor increasing the neuron firing intensity, and β influences the decay function which controls how quickly the overall rate returns to its baseline level λ_t after an event occurs.

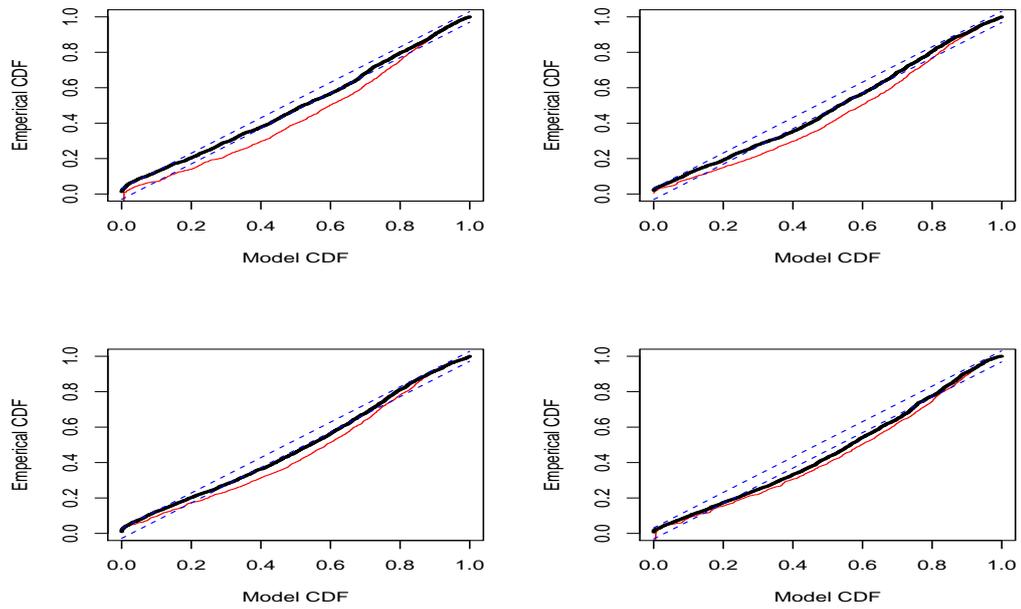


Figure 6.18: The Kolmogorov-smirnov plot of session EC 582 by Hawkes process and FPCA.

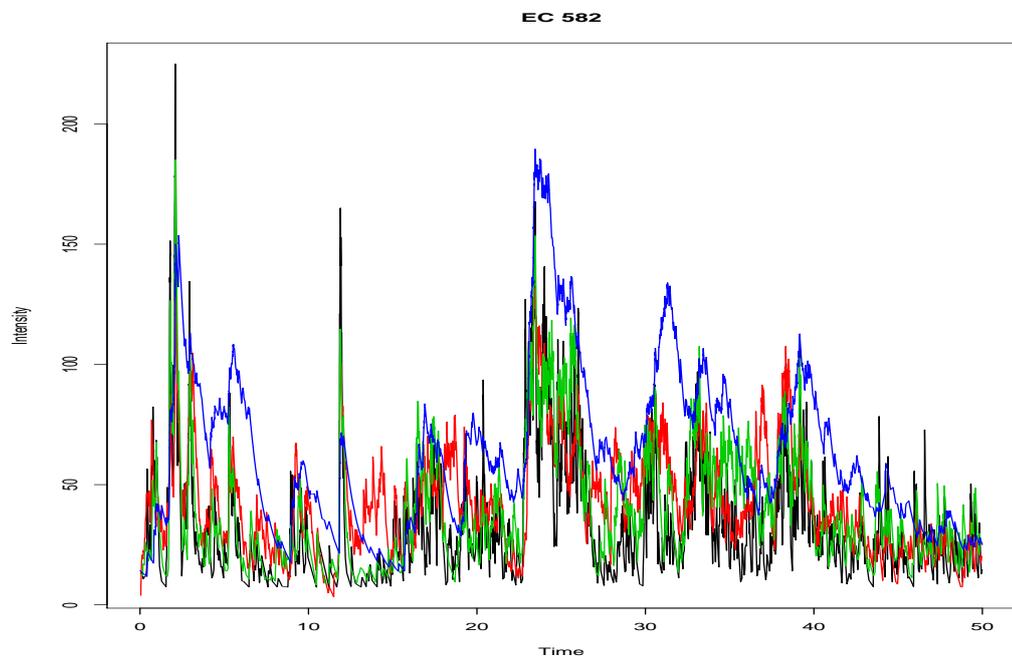


Figure 6.19: The individual intensity functions by Hawkes process with time varying base intensity function.

The Figure 6.19 shows the intensity functions from the semi-parametric modelling. It seems to have very similar results compare to Hawke's process with constant base intensity, there are some discrepancy on the base line but it can not be a major. Therefore it can be said that the history dependence is the most important in analysing neuron activities. According to Figure 6.20, there is no big difference between Hawke's process

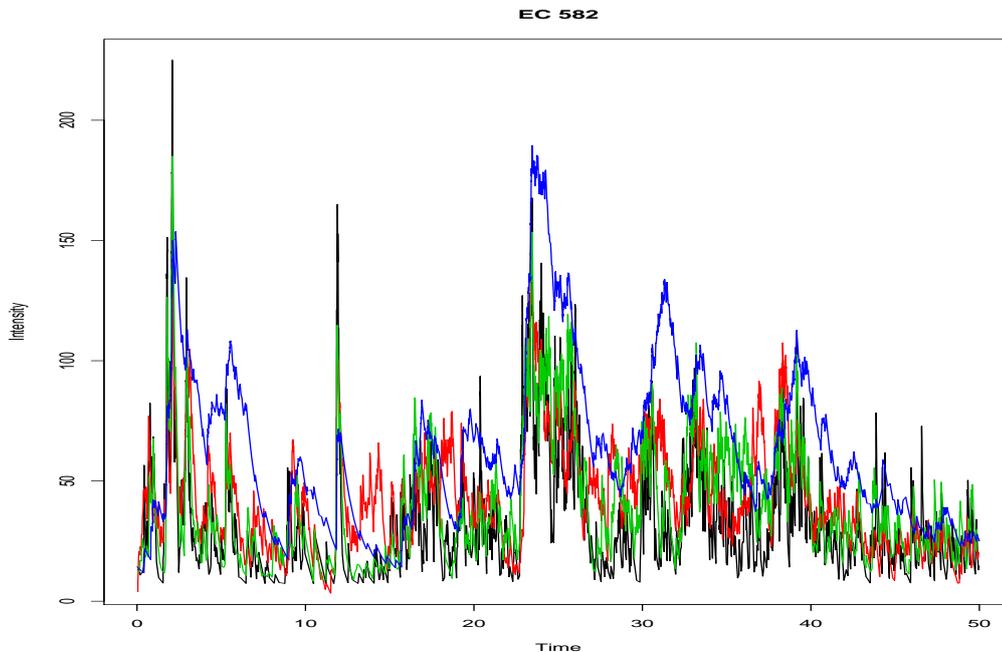


Figure 6.20: The individual intensity functions by Hawkes process with time varying base intensity function.

with constant base intensity and time varying base intensity. Trial 2 and 4 seem to be the same and Hawke's process with constant base intensity function gives better results on trial 1 but Hawke's process with time varying base intensity lead to better results on trial 3.

6.3 Discussion and conclusion

In this chapter, we discussed the modelling of neuron firing activities by parametric and non-parametric approaches which we have already applied in Chapters 2, 3 and 4. The main purpose of this chapter was comparing those modelling strategies and finding which approach is the most suitable for characterising the neuron activities.

Two different approaches of non-parametric modelling were applied in this chapter, one is the smoothing spline with B-spline basis function and another one is the functional

principal component analysis. There was no significant discrepancy from the two methods in the non-parametric modelling, intensity functions and the K-S plot had similar results. However the intensity function for the later approach contains other behaviour of the neighbouring neurons whereas the spline smoothing can only show the single neuron activities. One limitation of these approaches is that it is difficult to add history dependent effects. The history dependence is a very important factor for characterising neuron spike trains, hence ignoring this effect may cause a weakness of the study.

The history dependence model in the parametric modelling was not very suitable for this neuron data set. One of the reasons is the computational time consumption, the data set contains too many firing events and adding the history effects made it more difficult for computation. Non-parametric modelling did not have serious problems of time consumption for computing whereas the parametric approach took much more than non-parametric modelling. Therefore the first two approaches estimated intensity function for the whole data set, on the other hand the Hawkes process only estimated 1% of data set. If more time had been given it might estimate intensity function up to 10% or 15% which might represent 1 experiment of the trials because one trial on each session had ten repetitions of experiments, nevertheless it does not seem to suit large scale data sets. In addition, without smoothing the rate function seemed to have too many spikes, hence it made the interpretation difficult even if its Q-Q plot showed the better estimates. To improve the parametric modelling approach, time varying base intensity functions, that were estimated in functional principal component analysis, were applied as the semi-parametric modelling. It seem to reduce the computation time consuming, but the results did not seem to improve. It means that the most important feature of neuron activity is the history dependence effect.

To conclude this case study which compared of three different approaches for modelling the high frequency neuron data set, non-parametric approaches have the big advantage of the computing. Optimising parameters by `optim` in R took too much time therefore it does not seem to suit for the large data set. Considering the neighbour effect by applying functional principal component analysis, the estimation was improved however without history dependence which is a very important factor in neuron spike trains, there was a limitation. Although the Hawkes process took too much time, it had a better estimation.

Appendix A

Graphics

A.1 Graphics for Chapter 3

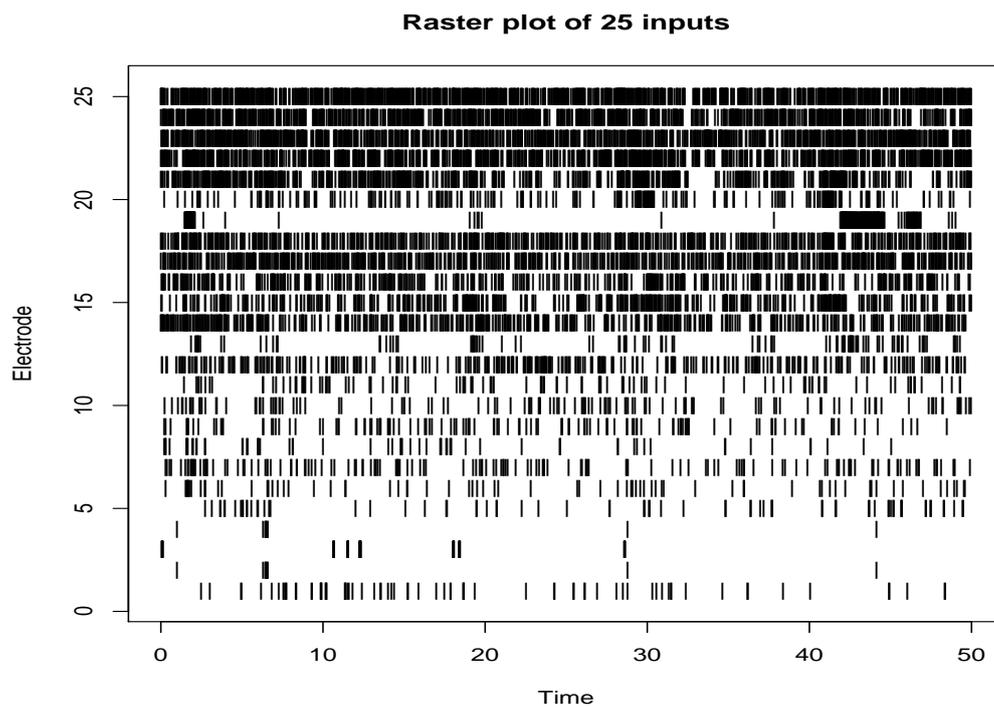


Figure A.1: Raster plot of 25 input data

Raster plot of 18 outputs

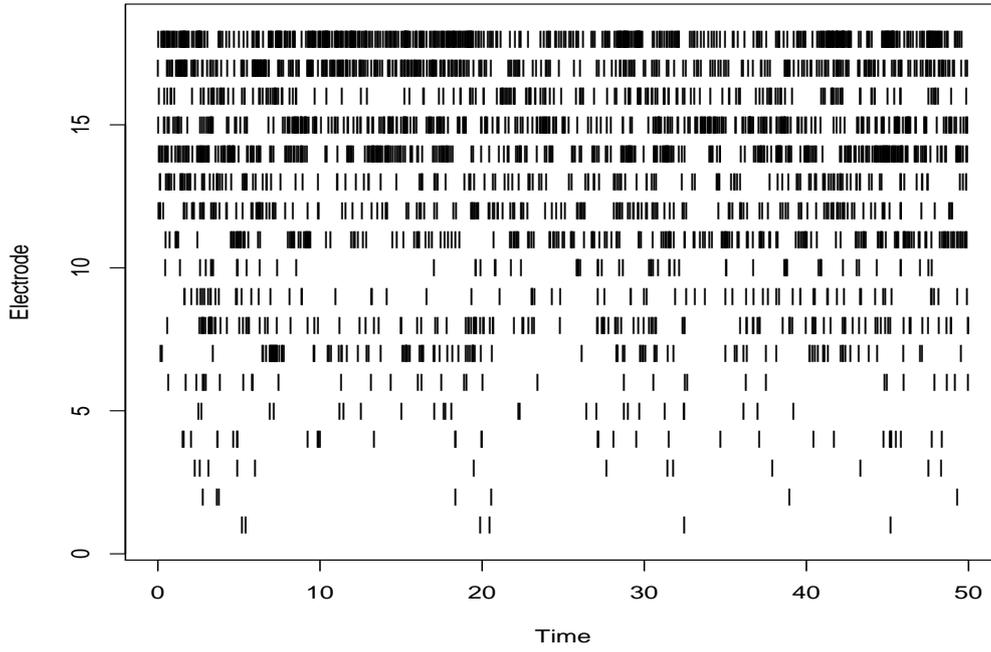


Figure A.2: Raster plot of 18 output data

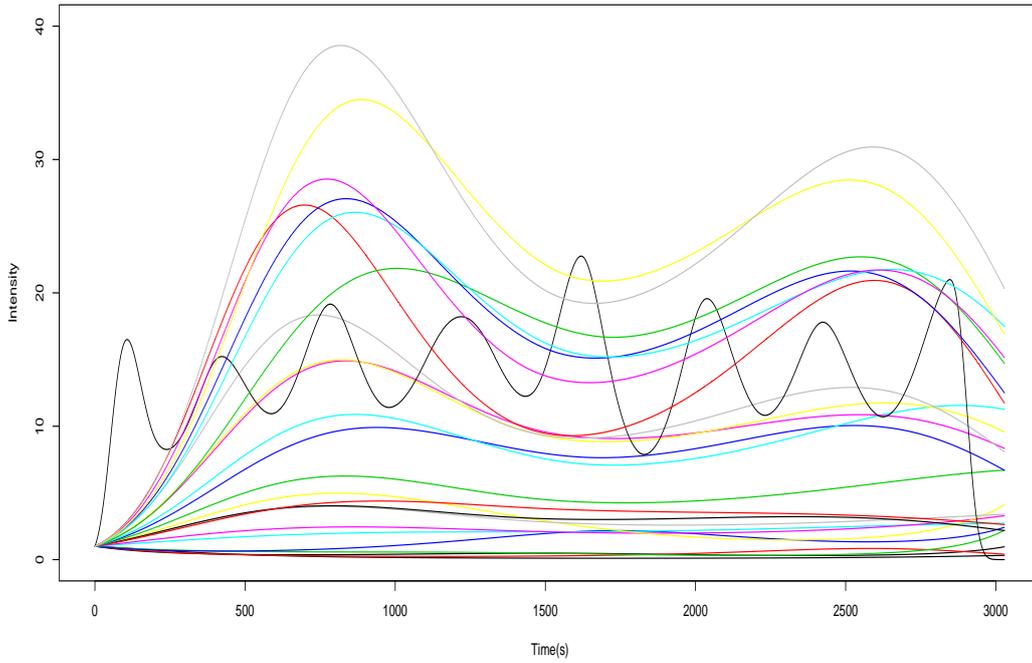


Figure A.3: Kernel estimation using B spline basis function

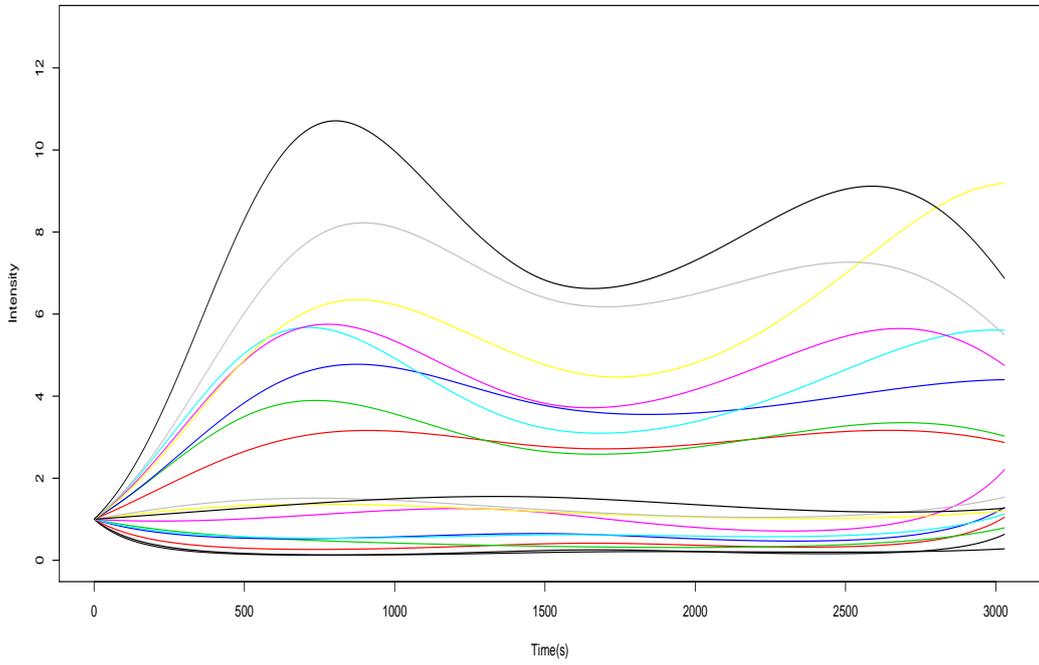


Figure A.4: Kernel estimation using B-spline basis function

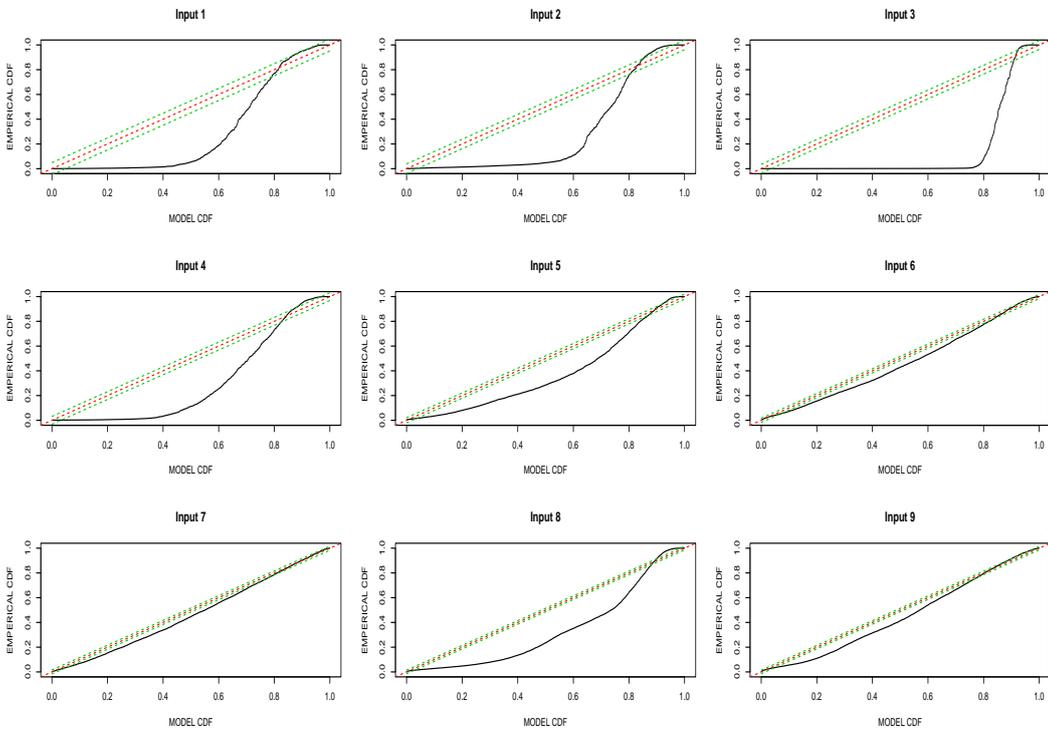


Figure A.5: kormogrov-smirnov plot, (black line-original data, red line-modified data)

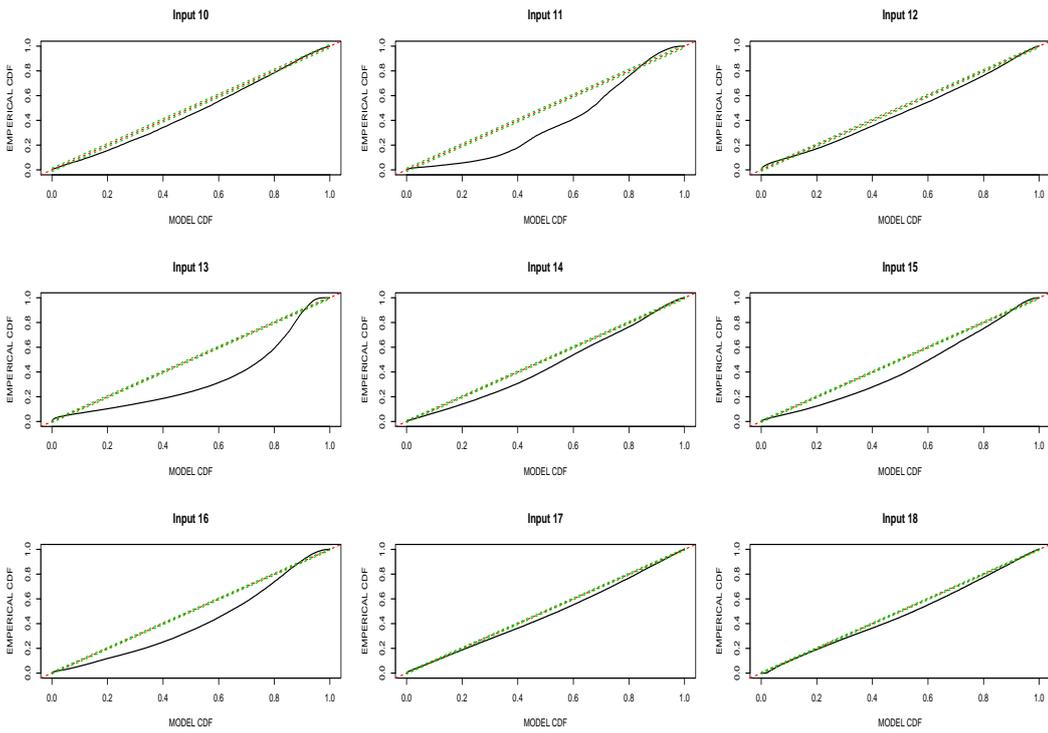


Figure A.6: kormogrov-smirnov plot, (black line-original data, red line-modified data)

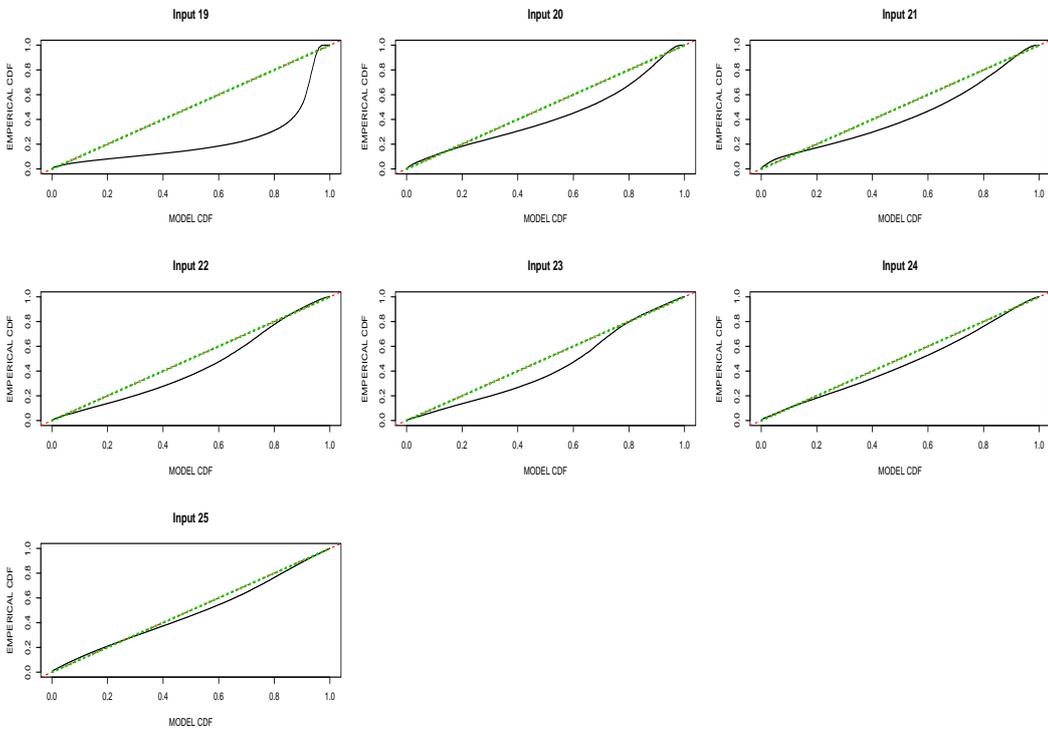


Figure A.7: kormogrov-smirnov plot, (black line-original data, red line-modified data)

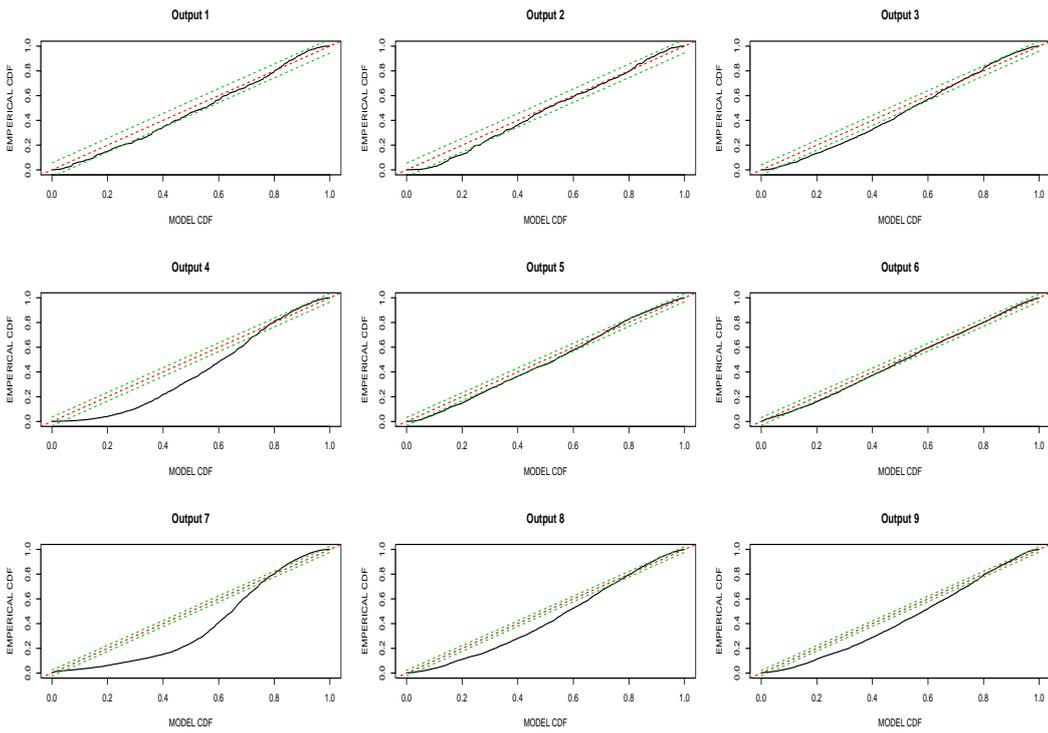


Figure A.8: kormogrov-smirnov plot, (black line-original data, red line-modified data)

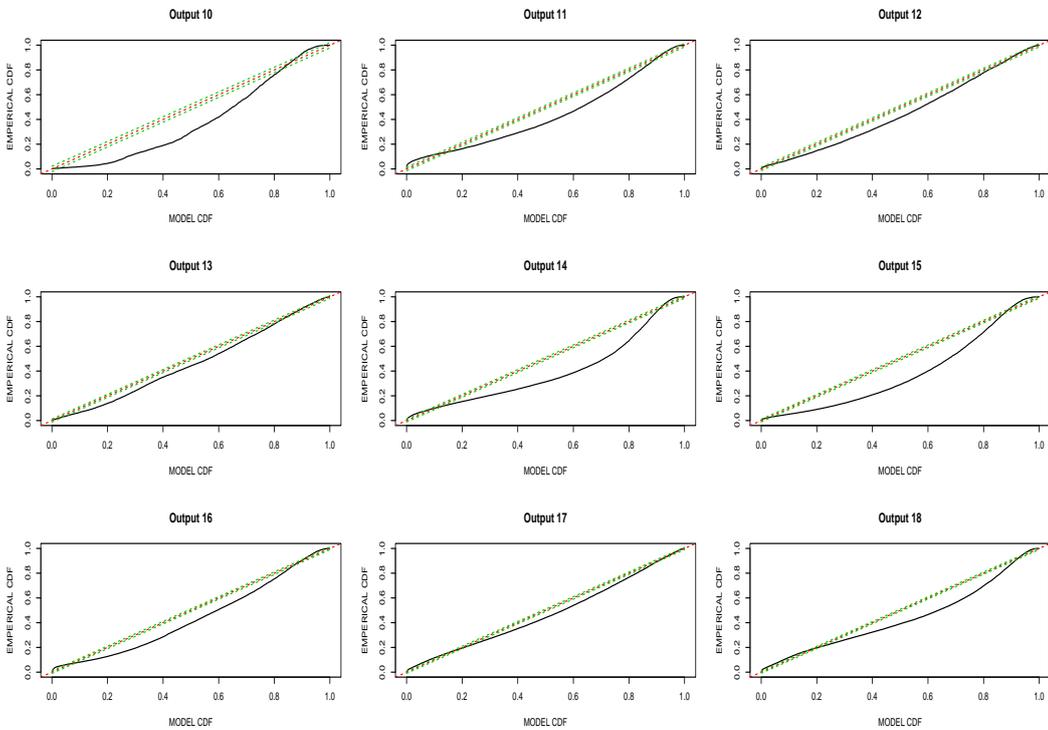


Figure A.9: kormogrov-smirnov plot, (black line-original data, red line-modified data)

A.2 Graphics for chapter 4

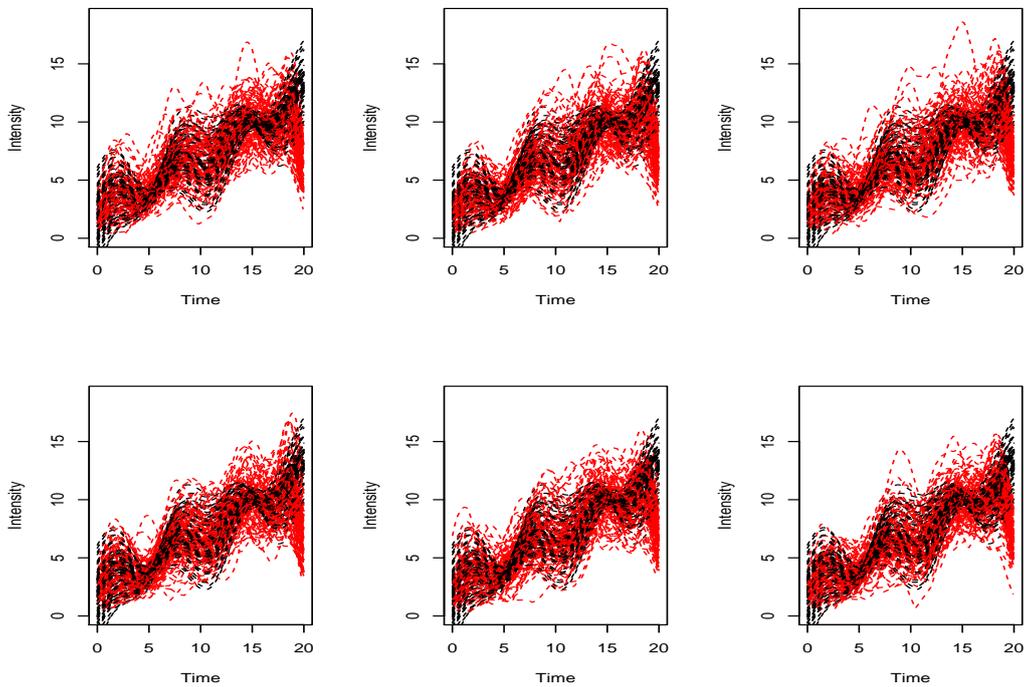


Figure A.10: The comparison of true intensity function and the individual intensity function from functional data analysis without PCA.

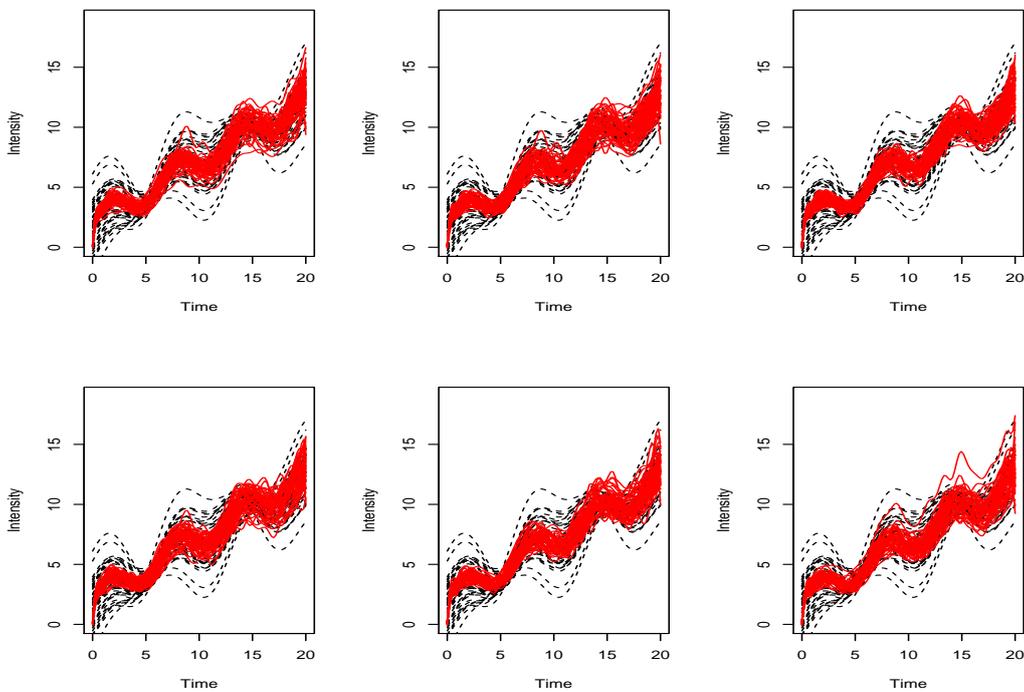


Figure A.11: The comparison of true intensity function and the intensity function for the first approach of principal component analysis with point process.

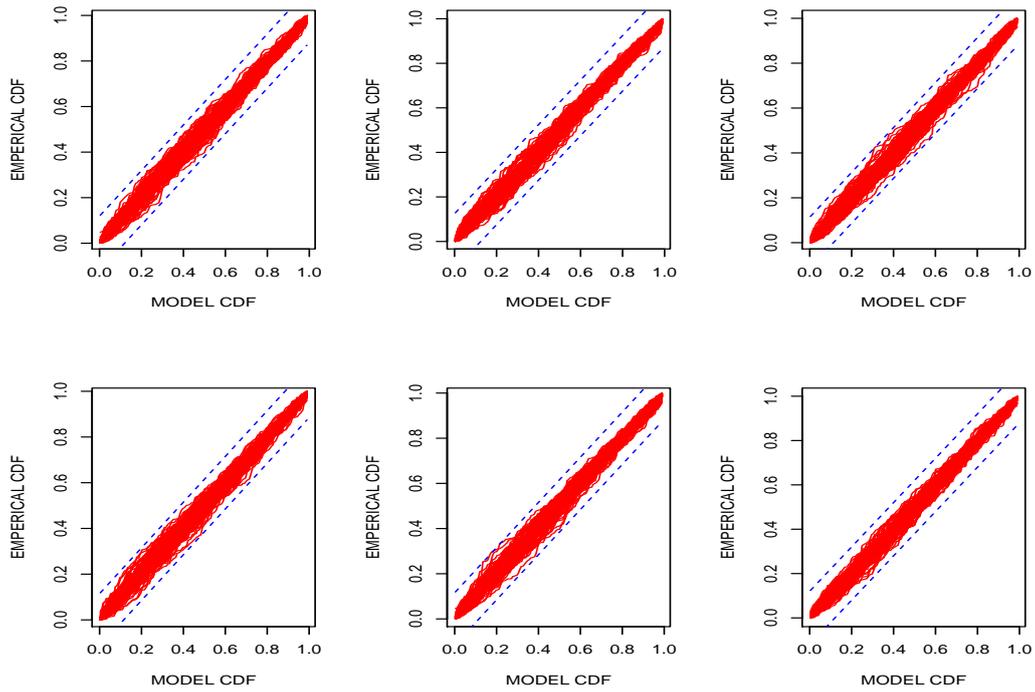


Figure A.12: The kolmogrov-smirnov plot for the simulated data.

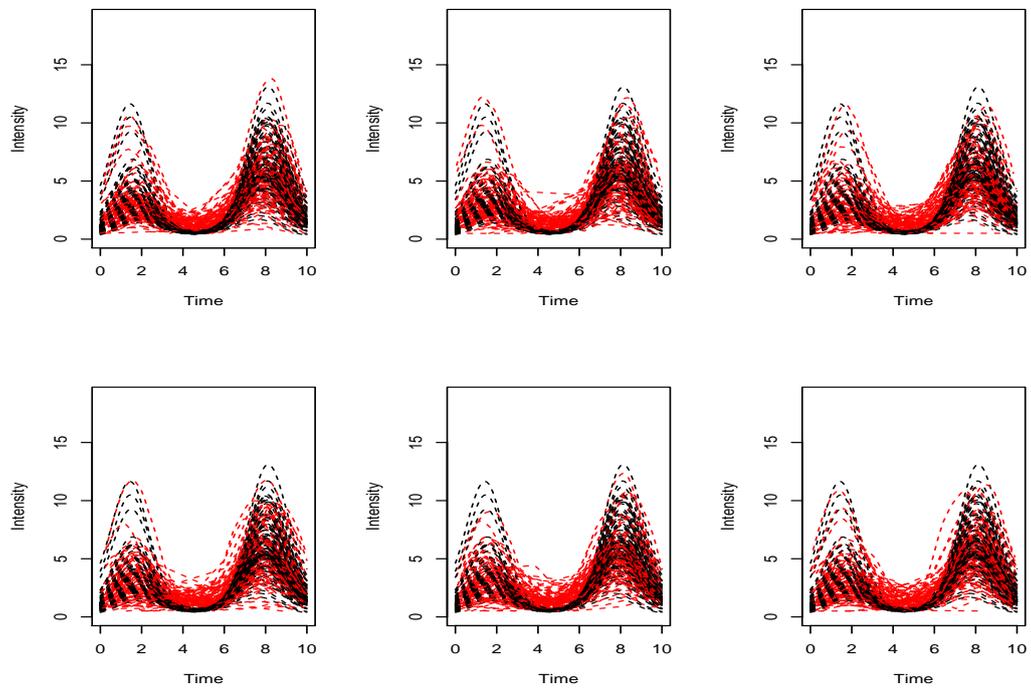


Figure A.13: The intensity function of the true and simulated data.

A.3 Graphics for chapter 6

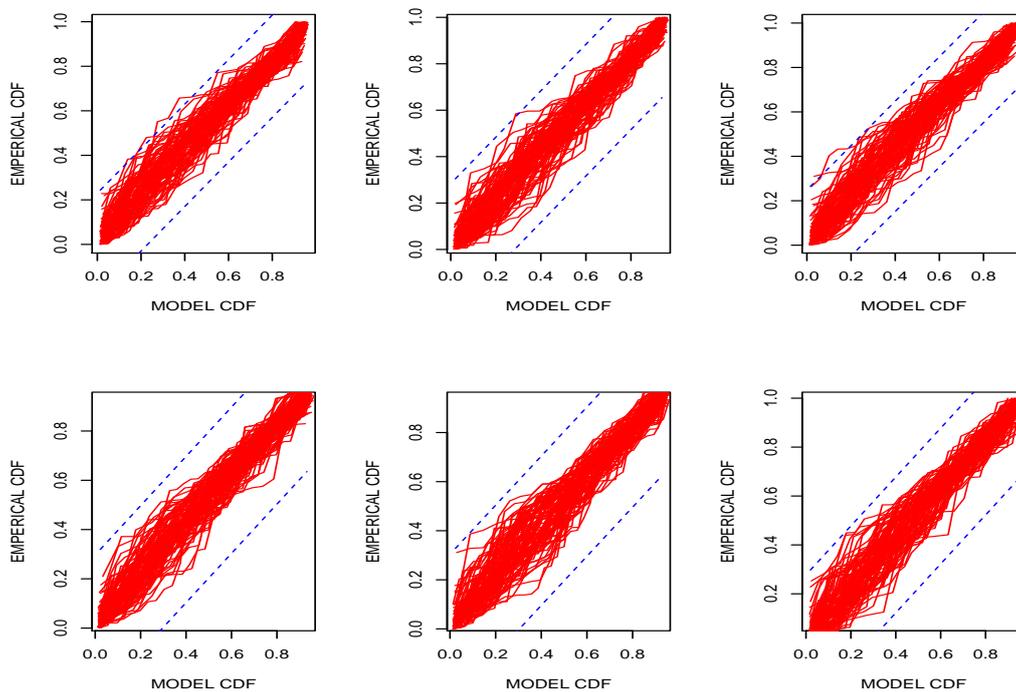


Figure A.14: The kolmogrov-smirnov plot for the simulated data.

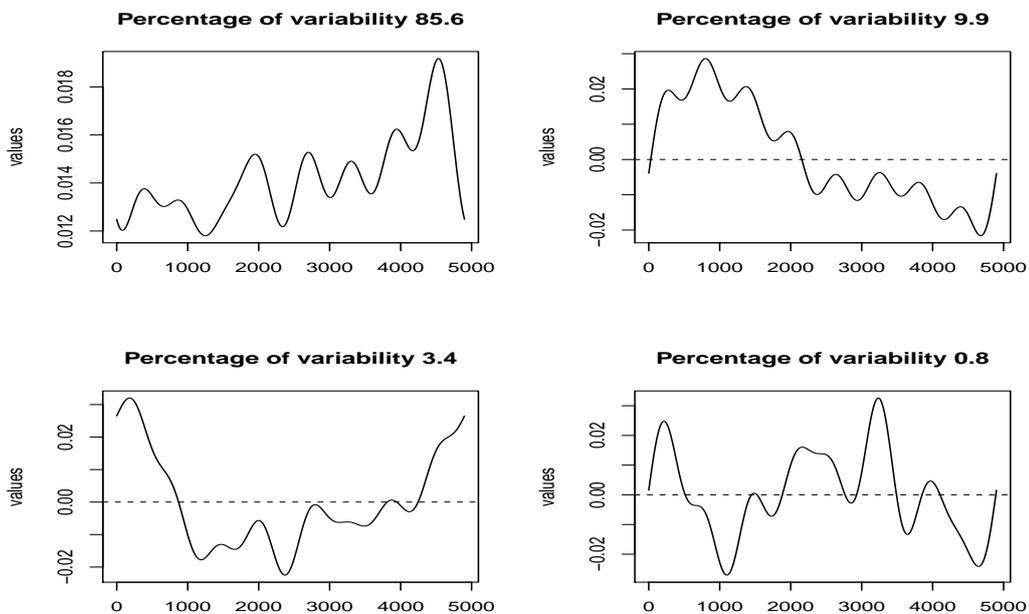


Figure A.15: The first four principal component curves of the EC430.

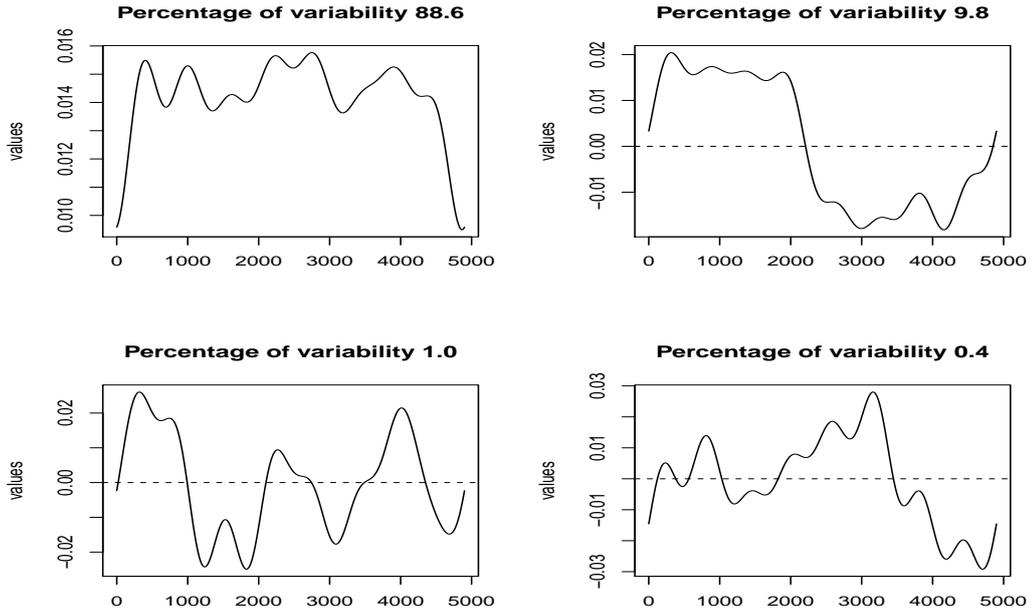


Figure A.16: The first four principal component curves of the EC448.

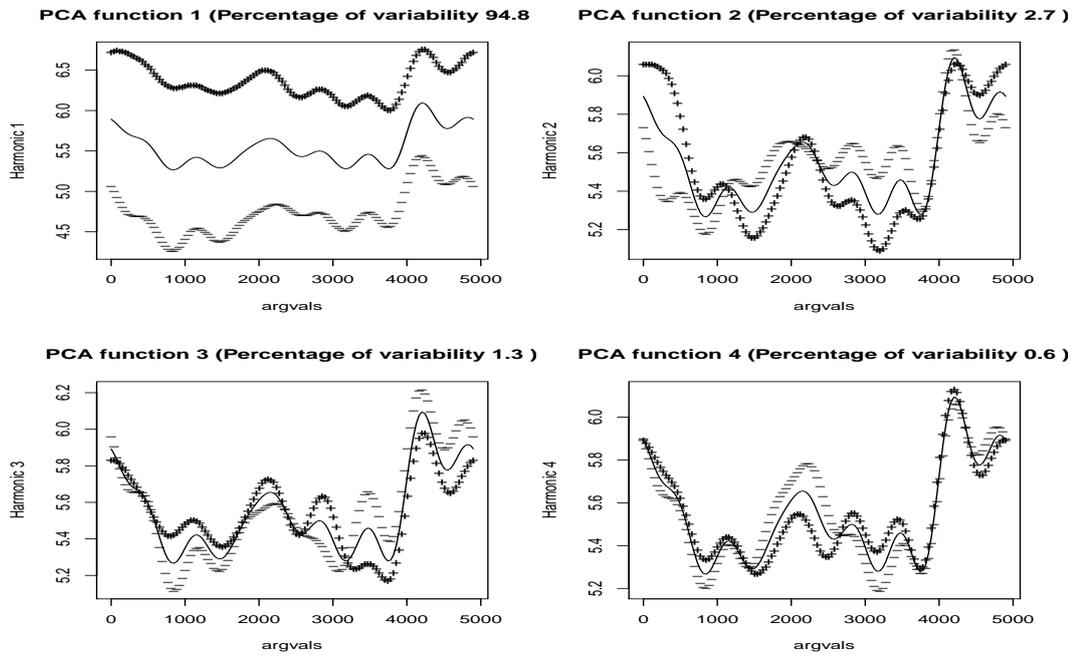


Figure A.17: The mean functions of the EC397 with four principal component curves.

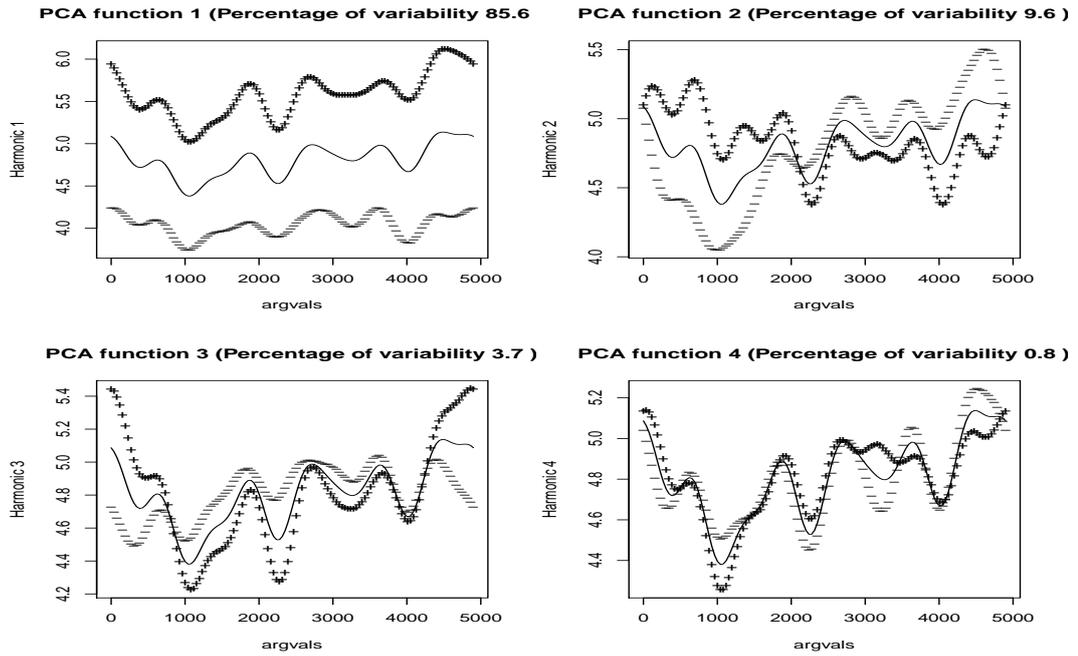


Figure A.18: The mean functions of the EC430 with four principal component curves.

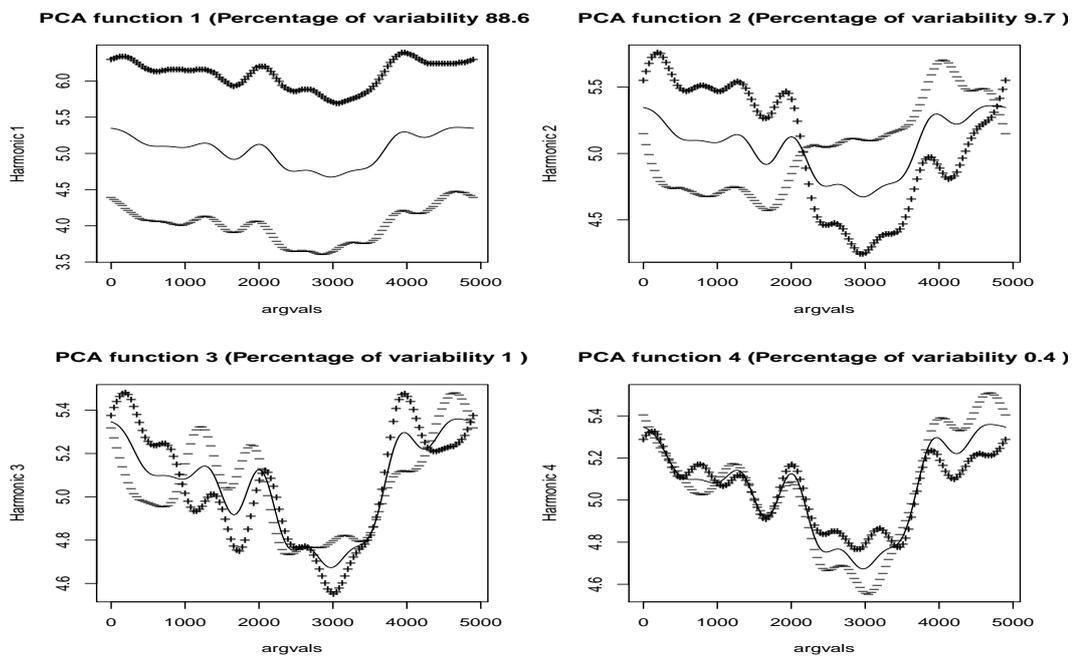


Figure A.19: The mean functions of the EC448 with four principal component curves.

Appendix B

R codes for computation

```
#####  
#####  
#####                               Code for Hawkes process  
#####  
#####  
#####  
#####  
load( 'bintimesec11.Rdata' )  
library(Rcpp); library( inline ); library( splines ); library( cubature )  
library( splines )  
CUT<-50  
#####  
t11 <- rep(NA,8); t12 <- rep(NA,8); t13 <- rep(NA,8); t14 <- rep(NA,8)  
for(i in 1:8) {  
  t11[i] <- list((T11[[i]][ which(T11[[i]] < CUT)]) )  
  t12[i] <- list((T12[[i]][ which(T12[[i]] < CUT)]) )  
  t13[i] <- list((T13[[i]][ which(T13[[i]] < CUT)]) )  
  t14[i] <- list((T14[[i]][ which(T14[[i]] < CUT)]) )  
  
}  
  
t1 <- rep(NA,6); t2 <- rep(NA,6); t3 <- rep(NA,6); t4 <- rep(NA,6)  
t1 <- list( t11[[1]], t11[[2]], t11[[3]], t11[[4]], t11[[6]], t11[[8]] )
```

```

t2 <- list(t12[[1]], t12[[2]], t12[[4]], t12[[5]], t12[[6]], t12[[7]])
t3 <- list(t13[[1]], t13[[2]], t13[[4]], t13[[5]], t13[[6]], t13[[8]])
t4 <- list(t14[[1]], t14[[2]], t14[[4]], t14[[5]], t14[[6]], t14[[7]])

```

```

Len1 <- unlist(lapply(t1, length))
Len2 <- unlist(lapply(t2, length))
Len3 <- unlist(lapply(t3, length))
Len4 <- unlist(lapply(t4, length))

```

```

lm1 <- length(t1)
lm2 <- length(t2)
lm3 <- length(t3)
lm4 <- length(t4)

```

```

#####

```

```

rang <- c(0,50)
h1 <- rep(NA,6); h2 <- rep(NA,6); h3 <- rep(NA,6); h4 <- rep(NA,6)
for(i in 1:lm1){
h1[i] <- list(c(0, t11[[i]])[-(Len1[i]+1)])
h2[i] <- list(c(0, t12[[i]])[-(Len2[i]+1)])
h3[i] <- list(c(0, t13[[i]])[-(Len3[i]+1)])
h4[i] <- list(c(0, t14[[i]])[-(Len4[i]+1)])
}

```

```

#####

```

```

lx2 <-function(p, x, h, t0=0.1, tmax=50, D=1) {
  n <- length(x)
  lambda <- rep(0, n)
  for(i in 1:n){
    lambda[i] <- p[1] + sum(p[2]*exp(-p[3]*(x[i] - h[which(x[i]>h)]))
      [which(x[i]-h[which(x[i]>h)]<D)])])
  }
}

```

```

}
for(i in 1:length(x)){
  if(lambda[i]<=0){
    lambda[i] <- 0
  }
}
int.lam <- function(t,b,H){
  b[1]+sum(b[2]*exp(-b[3]*(t-H[which(t>H)]))
  [which(t-H[which(t>H)]<D)])
}
integral <- adaptIntegrate(int.lam,lowerLimit=t0
,upperLimit= tmax,b=p,H=h)$integral
return (-sum(log(lambda))+ integral )
}
#####

sxh <-rep(0.001,3) # initial values
ypar1 <- rep(NA,6); ypar2 <- rep(NA,6)
ypar3 <- rep(NA,6); ypar4 <- rep(NA,6)
for(i in 1:lm1){
ypar1[i] <- list(optim(p=sxh,lx2,x=t11[[i]],control=list(maxit=100000
,trace=TRUE),h=h1[[i]])$par)
ypar2[i] <- list(optim(p=sxh,lx2,x=t12[[i]],control=list(maxit=100000
,trace=TRUE),h=h2[[i]])$par)
ypar3[i] <- list(optim(p=sxh,lx2,x=t13[[i]],control=list(maxit=100000
,trace=TRUE),h=h3[[i]])$par)
ypar4[i] <- list(optim(p=sxh,lx2,x=t14[[i]],control=list(maxit=100000
,trace=TRUE),h=h4[[i]])$par)
}

lamx1 <- rep(NA,6); lamx2 <- rep(NA,6)
lamx3 <- rep(NA,6); lamx4 <- rep(NA,6)

```

```

for(j in 1:lm1){
  lamx1[j] <- list(rep(0,Len1[j]))
  lamx2[j] <- list(rep(0,Len2[j]))
  lamx3[j] <- list(rep(0,Len3[j]))
  lamx4[j] <- list(rep(0,Len4[j]))
  for(i in 1: Len1[j]){
    lamx1[[j]][i] <- (ypar1[[j]][1]+sum(ypar1[[j]][2]
*exp(-ypar1[[j]][3]*(t11[[j]][i]-h1[[j]][which(t11[[j]]
[i]>h1[[j]])])[which(t11[[j]][i]-h1[[j]][which(t11[[j]]
[i]>h1[[j]])] <1)))))
    lamx2[[j]][i] <- (ypar2[[j]][1]+sum(ypar2[[j]][2]
*exp(-ypar2[[j]][3]*(t12[[j]][i]-h2[[j]][which(t12[[j]]
[i]>h2[[j]])])[which(t12[[j]][i]-h2[[j]][which(t12[[j]]
[i]>h2[[j]])] <1)))))
    lamx3[[j]][i] <- (ypar3[[j]][1]+sum(ypar3[[j]][2]
*exp(-ypar3[[j]][3]*(t13[[j]][i]-h3[[j]][which(t13[[j]]
[i]>h3[[j]])])[which(t13[[j]][i]-h3[[j]][which(t13[[j]]
[i]>h3[[j]])] <1)))))
    lamx4[[j]][i] <- (ypar4[[j]][1]+sum(ypar4[[j]][2]
*exp(-ypar4[[j]][3]*(t14[[j]][i]-h4[[j]][which(t14[[j]]
[i]>h4[[j]])])[which(t14[[j]][i]-h4[[j]][which(t14[[j]]
[i]>h4[[j]])] <1)))))
  }
} # intensity
#####
#####
#####                               Code for B-Spline basis function
#####
#####
#####
#####
load('c5secnew.Rdata');load('c6data.Rdata')

```

```

load( 'bintimesec11.Rdata' )
CUT<-4900
#####
t11 <- rep(NA,8);t12 <- rep(NA,8);t13 <- rep(NA,8);t14 <- rep(NA,8)
for(i in 1:8) {
  t11[i] <- list((T11[[i]][which(T11[[i]] < CUT)]))
  t12[i] <- list((T12[[i]][which(T12[[i]] < CUT)]))
  t13[i] <- list((T13[[i]][which(T13[[i]] < CUT)]))
  t14[i] <- list((T14[[i]][which(T14[[i]] < CUT)]))
}

t1 <- rep(NA,6);t2 <- rep(NA,6);t3 <- rep(NA,6);t4 <- rep(NA,6)

t1 <- list(t11[[1]],t11[[2]],t11[[3]],t11[[4]],t11[[6]],t11[[8]])
t2 <- list(t12[[1]],t12[[2]],t12[[4]],t12[[5]],t12[[6]],t12[[7]])
t3 <- list(t13[[1]],t13[[2]],t13[[4]],t13[[5]],t13[[6]],t13[[8]])
t4 <- list(t14[[1]],t14[[2]],t14[[4]],t14[[5]],t14[[6]],t14[[7]])

Len1 <- unlist(lapply(t1,length));Len2 <- unlist(lapply(t2,length))
Len3 <- unlist(lapply(t3,length));Len4 <- unlist(lapply(t4,length))
lm1 <- length(t1);lm2 <- length(t2);lm3 <- length(t3);lm4 <- length(t4)
R1 <- range(0,4900);R2 <- range(0,4900)
R3 <- range(0,4900);R4 <- range(0,4900)
S1 <- seq(0,4900,by=10);S2 <- seq(0,4900,by=10)
S3 <- seq(0,4900,by=10);S4 <- seq(0,4900,by=10)
Sn1 <- length(S1);Sn2 <- length(S2);Sn3 <- length(S3);Sn4 <- length(S4)

#####
E1 <- matrix(rep(0,lm1*Sn1),ncol=lm1)
E2 <- matrix(rep(0,lm1*Sn2),ncol=lm2)
E3 <- matrix(rep(0,lm1*Sn3),ncol=lm3)

```

```

E4 <- matrix(rep(0,lm1*Sn4),ncol=lm4)

oc <- function(x,n,l){
  o1 <- rep(0,l)
  for(i in 1:l){
    o1[i] <- list(rep(0,n[i]))
  }
  return(o1)
}

o1 <- oc(t1,Len1,lm1);o2 <- oc(t2,Len2,lm2)
o3 <- oc(t3,Len3,lm3);o4 <- oc(t4,Len4,lm4)

fl <- function(x,o,b,n,y,l){
  for(k in 1:l){
    for(i in 1:n[k]){
      o[[k]][i] <- min(which(x[[k]][i]<b))
    }
    for(i in 1:length(b)){
      y[i,k] <- length(which(o[[k]]==i))
    }
  }
  return(y/10)
}

y1 <- fl(t1,o1,S1,Len1,E1,lm1);y2 <- fl(t2,o2,S2,Len2,E2,lm2)
y3 <- fl(t3,o3,S3,Len3,E3,lm3);y4 <- fl(t4,o4,S4,Len4,E4,lm4)

library(fda)
n1 <- length(S1);n2 <- length(S2);n3 <- length(S3);n4 <- length(S4)

basis1 <- create.bspline.basis(range(S1),17)
basis2 <- create.bspline.basis(range(S2),17)

```

```

basis3 <- create.bspline.basis(range(S3),17)
basis4 <- create.bspline.basis(range(S4),17)

yfd1 <- smooth.basis(S1,y1,basis1)$fd
#functional data object containing a smooth of the data
yfd2 <- smooth.basis(S2,y2,basis2)$fd
yfd3 <- smooth.basis(S3,y3,basis3)$fd
yfd4 <- smooth.basis(S4,y4,basis4)$fd

if1 <- eval.basis(S1,basis1)%*%yfd1$coefs
if2 <- eval.basis(S2,basis2)%*%yfd2$coefs
if3 <- eval.basis(S3,basis3)%*%yfd3$coefs
if4 <- eval.basis(S4,basis4)%*%yfd4$coefs

#####
#####
# model checking
#####

Denf1 <- function(o,s,n,g){
  DY <- rep(NA,3)
  dy <- rep(NA,n)
  for(i in 1:n){
    dy[i] <- list((approx(x=s,y=o[,i],xo=g[[i]])$y))
  }
  return(dy)
}

DM1 <- Denf1(o=if1,s=S1,n=lm1,g=t1)
DM2 <- Denf1(o=if2,s=S2,n=lm2,g=t2)
DM3 <- Denf1(o=if3,s=S3,n=lm3,g=t3)
DM4 <- Denf1(o=if4,s=S4,n=lm4,g=t4)

```

```

plot(t1[[1]],DM1[[1]],type='l',ylim=c(0,150))
for(i in 2:6){
  lines(t1[[i]],DM1[[i]])
}
plot(t2[[1]],DM2[[1]],type='l',ylim=c(0,150))
for(i in 2:6){
  lines(t2[[i]],DM2[[i]])
}
QQ1 <- function(x,y,n,l){
  u1 <- rep(NA,n);cu1 <- rep(NA,n);cl1 <- rep(NA,n)
  q1 <- rep(NA,n);q11 <- rep(NA,n)
  for(i in 1:n){
    u1[i] <- list(1/l[i]*(1:(l[i]-1)-1/2))
    cu1[i] <- list(u1[[i]]+1.36/((l[i]-1)^(1/2)))
    cl1[i] <- list(u1[[i]]-1.36/((l[i]-1)^(1/2)))
    q11[i] <- list(rep(NA,l[i]-1))
    for(j in 1:(l[i]-1)){
      q11[[i]][j] <- ((x[[i]][j+1]-x[[i]][j])
/2*(y[[i]][j+1]+y[[i]][j]))
    }
    q1[i] <- list(sort(1-exp(-q11[[i]]),decreasing=F))
  }

  return(list(u=u1,cu=cu1,cl=cl1,q=q1))
}
MQ1 <- QQ1(x=t1,y=DM1,n=lm1,l=Len1) # K-S plot
MQ2 <- QQ1(x=t2,y=DM2,n=lm2,l=Len2)
MQ3 <- QQ1(x=t3,y=DM3,n=lm3,l=Len3)
MQ4 <- QQ1(x=t4,y=DM4,n=lm4,l=Len4)

```

```

#####
#####

```

```

#####          Code for Functional principal component analysis
#####
#####
#####
#####
pca1 <- pca.fd(yfd1,4)
pca2 <- pca.fd(yfd2,4)
pca3 <- pca.fd(yfd3,4)
pca4 <- pca.fd(yfd4,4)
#####

DT1 <- rep(NA,lm1);DT2 <- rep(NA,lm2)
DT3 <- rep(NA,lm3);DT4 <- rep(NA,lm4)
for(i in 1:lm1){
    DT1[i] <- list(density(t1[[i]],kernel='epanechnikov',
    ,from=0,to=4900,n=Sn1))
    DT2[i] <- list(density(t2[[i]],kernel='epanechnikov',
    ,from=0,to=4900,n=Sn2))
    DT3[i] <- list(density(t3[[i]],kernel='epanechnikov',
    ,from=0,to=4900,n=Sn3))
    DT4[i] <- list(density(t4[[i]],kernel='epanechnikov',
    ,from=0,to=4900,n=Sn4))
}

DY1 <- matrix(rep(NA,lm1*Sn1),nrow=lm1)
DY2 <- matrix(rep(NA,lm2*Sn2),nrow=lm2)
DY3 <- matrix(rep(NA,lm3*Sn3),nrow=lm3)
DY4 <- matrix(rep(NA,lm4*Sn4),nrow=lm4)

for(i in 1:lm1){
    DY1[i,] <- DT1[[i]]$y
    DY2[i,] <- DT2[[i]]$y

```

```

        DY3[i ,] <- DT3[[ i ]] $y
        DY4[i ,] <- DT4[[ i ]] $y

    }
DX <- DT1[[1]] $x
Dmu1 <- colMeans(DY1)
Dmu2 <- colMeans(DY2)
Dmu3 <- colMeans(DY3)
Dmu4 <- colMeans(DY4)

#pdf('pdaind.pdf')
par(mfrow=c(2,2))
plot(DX,DY1[1 ,]*Len1[1] , type='l' ,ylim=c(0,69) ,main='EC582'
, xlab='Time' ,ylab='Intensity')
for(i in 2:lm1){
    lines(DX,DY1[i ,]*Len1[i])
}
plot(DX,DY2[1 ,]*Len2[1] , type='l' ,ylim=c(0,69) ,main='EC397'
, xlab='Time' ,ylab='Intensity')
for(i in 2:lm2){
    lines(DX,DY2[i ,]*Len2[i])
}
plot(DX,DY3[1 ,]*Len3[1] , type='l' ,ylim=c(0,69) ,main='EC430'
, xlab='Time' ,ylab='Intensity')
for(i in 2:lm3){
    lines(DX,DY3[i ,]*Len3[i])
}
plot(DX,DY4[1 ,]*Len4[1] , type='l' ,ylim=c(0,69) ,main='EC448'
, xlab='Time' ,ylab='Intensity')
for(i in 2:lm4){
    lines(DX,DY4[i ,]*Len4[i])
}

```

```

}

#dev.off()

#####

score1 <- pca1$scores
score2 <- pca2$scores
score3 <- pca3$scores
score4 <- pca4$scores

#Plot of score function

plot(score1[,1], score1[,2], pch=20, xlim=c(-180,150), ylim=c(-32,62)
, ylab='Component_2nd_Score', xlab='Component_1st_Score')
text(score1[,1], score1[,2], labels=1:8, cex=0.5, col=1, pos=3)
points(score2[,1], score2[,2], pch=20, col=2)
text(score2[,1], score2[,2], labels=1:8, cex=0.5, col=2, pos=3)
points(score3[,1], score3[,2], pch=20, col=3)
text(score3[,1], score3[,2], labels=1:8, cex=0.5, col=3, pos=3)
points(score4[,1], score4[,2], pch=20, col=4)
text(score4[,1], score4[,2], labels=1:8, cex=0.5, col=4, pos=3)

#####

ef1 <- eval.basis(S1, basis1)%*%pca1$harmonics$coefs
ef2 <- eval.basis(S2, basis2)%*%pca2$harmonics$coefs
ef3 <- eval.basis(S3, basis3)%*%pca3$harmonics$coefs
ef4 <- eval.basis(S4, basis4)%*%pca4$harmonics$coefs
mean1 <- eval.basis(S1, basis1)%*%pca1$meanfd$coefs
mean2 <- eval.basis(S2, basis2)%*%pca2$meanfd$coefs
mean3 <- eval.basis(S3, basis3)%*%pca3$meanfd$coefs
mean4 <- eval.basis(S4, basis4)%*%pca4$meanfd$coefs
mxij1 <- rep(NA, lm1)
mk <- function(x, y, l, z){
mk1 <- rep(NA, 3)
for(k in 1:3){

```

```

for(j in 1:l){
  emx1 <- rep(NA,(k+1))
  for(i in 1:(k+1)){
    emx1[i] <- list(approx(x=x,y=y[,i],xo=z[[j]])$y)
  }
  mxij1[j] <- list(emx1)
}
mk1[k] <- list(mxij1)
}
return(mk1)
}
mk1 <- mk(x=S1,y=ef1,l=lm1,z=t1)
mk2 <- mk(x=S2,y=ef2,l=lm2,z=t2)
mk3 <- mk(x=S3,y=ef3,l=lm3,z=t3)
mk4 <- mk(x=S4,y=ef4,l=lm4,z=t4)

#####
h=20
MP <- function(x,y,z,l,ll){
MP1 <- rep(NA,3)
Mp1 <- rep(NA,lm1)
for(k in 1:3){
  for(i in 1:l){
    Mp11 <- rep(NA,(k+1))
    for(j in 1:(k+1)){
      Mp11[j] <- (1/length(x[[k]][[i]][[j]])
        *sum(x[[k]][[i]][[j]])*ll[i] - sum(y*z[,j])*h)
    }
    Mp1[i] <- list(Mp11)
  }
  MP1[k] <- list(Mp1)
}
}

```

```

return(MP1)
}
MP1 <- MP(x=mk1 ,y=mean1 ,z=ef1 , l=lm1 , ll=Len1)
MP2 <- MP(x=mk2 ,y=mean2 ,z=ef2 , l=lm2 , ll=Len2)
MP3 <- MP(x=mk3 ,y=mean3 ,z=ef3 , l=lm3 , ll=Len3)
MP4 <- MP(x=mk4 ,y=mean4 ,z=ef4 , l=lm4 , ll=Len4)
#####
Denf <- function(x ,y ,z ,l ,n){
Mden <- rep(NA,3);Mlam <- rep(NA,3)
for(k in 1:3){
  mden <- rep(NA,lm1);mden1 <- rep(NA,lm1);mlam <- rep(NA,lm1)
  for(i in 1:l){
    mden[i] <- list(z + y[,1:(k+1)]%*%x[[k]][[i]])
    mden1[i] <- list(z + y[,1:(k+1)]%*%x[[k]][[i]])

    mden[[i]][which(mden1[[i]]<0)] <- 0
    mlam[i] <- list(mden1[[i]]*n[i])
  }
  Mden[k] <- list(mden1)
  Mlam[k] <- list(mlam)
}
return(list(den=Mden, int=Mlam))
}
Mden1 <- Denf(x=MP1,y=ef1 ,z=mean1 ,l=lm1 ,n=Len1)$den
# individual intensity function of FPCA
Mden2 <- Denf(x=MP2,y=ef2 ,z=mean2 ,l=lm2 ,n=Len2)$den
Mden3 <- Denf(x=MP3,y=ef3 ,z=mean3 ,l=lm3 ,n=Len3)$den
Mden4 <- Denf(x=MP4,y=ef4 ,z=mean4 ,l=lm4 ,n=Len4)$den
par(mfrow=c(2,2))
plot(S1,Mden1[[3]][[1]], type='l',ylim=c(0,159),col=1,xlab='Time',
,ylab='Intensity',main='EC_582')
for(i in 2:lm1){

```

```

    lines(S1,Mden1[[3]][[i]],col=1)

}

plot(S2,Mden2[[3]][[1]],type='l',ylim=c(0,159),col=1,xlab='Time',
,ylab='Intensity',main='EC_397')
for(i in 2:lm2){
    lines(S2,Mden2[[3]][[i]],col=1)

}

plot(S3,Mden3[[3]][[1]],type='l',ylim=c(0,159),col=1,xlab='Time',
,ylab='Intensity',main='EC_430')
for(i in 2:lm1){
    lines(S3,Mden3[[3]][[i]],col=1)

}

plot(S4,Mden4[[3]][[1]],type='l',ylim=c(0,159),col=1,xlab='Time',
,ylab='Intensity',main='EC_448')
for(i in 2:lm4){
    lines(S4,Mden4[[3]][[i]],col=1)

}

#####
# model checking
#####

Denf1 <- function(o,s,n,g){
  DY <- rep(NA,3)
  for(k in 1:3){
    dy <- rep(NA,n)
    for(i in 1:n){
      dy[i] <- list((approx(x=s,y=o[[k]][[i]],xo=g[[i]])$y))
    }
  }
}

```

```

    }
    DY[k] <- list(dy)
  }
  return(DY)
}

DM1 <- Denf1(o=Mden1, s=S1, n=lm1, g=t1)
DM2 <- Denf1(o=Mden2, s=S2, n=lm2, g=t2)
DM3 <- Denf1(o=Mden3, s=S3, n=lm3, g=t3)
DM4 <- Denf1(o=Mden4, s=S4, n=lm4, g=t4)

QQ1 <- function(x, y, n, l){
  u <- rep(NA, 3); cl <- rep(NA, 3); cu <- rep(NA, 3); q <- rep(NA, 3)
  for(k in 1:3){
    u1 <- rep(NA, n); cu1 <- rep(NA, n); cl1 <- rep(NA, n)
    q1 <- rep(NA, n); q11 <- rep(NA, n)
    for(i in 1:n){
      u1[i] <- list(1/l[i] * (1:(l[i]-1)-1/2))
      cu1[i] <- list(u1[[i]]+1.36/((l[i]-1)^(1/2)) )
      cl1[i] <- list(u1[[i]]-1.36/((l[i]-1)^(1/2)) )
      q11[i] <- list(rep(NA, l[i]-1))
      for(j in 1:(l[i]-1)){
        q11[[i]][j] <- ((x[[i]][j+1]-x[[i]][j])
          /2*(y[[3]][[i]][j+1]+y[[3]][[i]][j]))
      }
      q1[i] <- list(sort(1-exp(-q11[[i]]), decreasing=F))
    }
    u[k] <- list(u1)
    cu[k] <- list(cu1)
    cl[k] <- list(cl1)
    q[k] <- list(q1)
  }
}

```

```

    return( list (u=u , cu=cu , cl=cl , q=q))
}
MQ1 <- QQ1(x=t1 , y=DM1, n=lm1 , l=Len1) # K-S plot
MQ2 <- QQ1(x=t2 , y=DM2, n=lm2 , l=Len2)
MQ3 <- QQ1(x=t3 , y=DM3, n=lm3 , l=Len3)
MQ4 <- QQ1(x=t4 , y=DM4, n=lm4 , l=Len4)

#####

library(fda.usc)
unlis <- function(x, l){
    m <- matrix(unlist(x[[3]]), ncol=1)
    return(m)
}

md1 <- unlis(Mden1, lm1)
md2 <- unlis(Mden2, lm2)
md3 <- unlis(Mden3, lm3)
md4 <- unlis(Mden4, lm4)
D1 <- rep(NA, lm1); D2 <- rep(NA, lm2); D3 <- rep(NA, lm3); D4 <- rep(NA, lm4)
for(i in 1:lm1){
    D1[i] <- list(density(t1[[i]], bw=bw1[i]
/sqrt(66), kernel="epanechnikov", from=0, to=R1[2], n=Sn1))
    D2[i] <- list(density(t2[[i]], bw=bw2[i]
/sqrt(66), kernel="epanechnikov", from=0, to=R2[2], n=Sn2))
    D3[i] <- list(density(t3[[i]], bw=bw3[i]
/sqrt(66), kernel="epanechnikov", from=0, to=R3[2], n=Sn3))
    D4[i] <- list(density(t4[[i]], bw=bw4[i]
/sqrt(66), kernel="epanechnikov", from=0, to=R4[2], n=Sn4))
}
Dy1 <- (matrix(rep(NA, lm1*Sn1), nrow=lm1))
Dy2 <- (matrix(rep(NA, lm2*Sn2), nrow=lm2))
Dy3 <- (matrix(rep(NA, lm3*Sn3), nrow=lm3))

```

```

Dy4 <- (matrix(rep(NA, lm4*Sn4), nrow=lm4))
for(i in 1:lm1){
  Dy1[i,] <- D1[[i]]$y
  Dy2[i,] <- D2[[i]]$y
  Dy3[i,] <- D3[[i]]$y
  Dy4[i,] <- D4[[i]]$y
}

mdatos12 <- cbind(md1,md2)
mdatos13 <- cbind(md1,md3)
mdatos14 <- cbind(md1,md4)
mgroup12 <- c(rep(1,lm1),rep(2,lm2))
mgroup13 <- c(rep(1,lm1),rep(2,lm3))
mgroup14 <- c(rep(1,lm1),rep(2,lm4))

tdatos2 <- cbind(md1,md2,md3,md4)

tgroup <- c(rep(1,lm1),rep(2,lm2),rep(3,lm3),rep(4,lm4))

mfd12 <- fdata(t(mdatos12), argvals=S1)
mfd13 <- fdata(t(mdatos13), argvals=S1)
mfd14 <- fdata(t(mdatos14), argvals=S1)

tfd2 <- fdata(t(tdatos2), argvals=S1)

mres12=anova.onefactor(mfd12, mgroup12, nboot=20, plot=TRUE)
mres13=anova.onefactor(mfd13, mgroup13, nboot=20, plot=TRUE)
mres14=anova.onefactor(mfd14, mgroup14, nboot=20, plot=TRUE)

mres2=anova.onefactor(mfd2, tgroup, nboot=20, plot=TRUE) # functional anova

```

```
mres3=anova.onefactor(mfl2 , tgroup , nboot=20,plot=TRUE)
```

Bibliography

- [1] David R. Brillinger. Some statistical methods for random process data from seismology and neurophysiology. *The Annals of Statistics*, 16(1,):1–54, 1988.
- [2] David R. Brillinger. Nerve cell spike train data analysis: A progression of technique. *Journal of the American Statistical Association*, 87(418,):260–271, 1992.
- [3] Emery N Brown, Riccardo Barbieri, Valérie Ventura, Robert E Kass, and Loren M Frank. The time-rescaling theorem and its application to neural spike train data analysis. *Neural computation*, 14(2):325–346, 2002.
- [4] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media, 2007.
- [5] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*, volume 2. Springer, 2007.
- [6] Sam A Deadwyler, Terence Bunn, and Robert E Hampson. Hippocampal ensemble activity during spatial delayed-nonmatch-to-sample performance in rats. *The Journal of neuroscience*, 16(1):354–372, 1996.
- [7] Gerstein. Donald H. Perkel, George L. Neuronal spike trains and stochastic point process. *Biophysical journal*, 7(1,):391–418, 1967.
- [8] Gerstein. Donald H. Perkel, George L. Neuronal spike trains and stochastic point process. *Biophysical journal*, 7(2,):419–440, 1967.
- [9] Uri T. Eden. Point process models for neuron spike trains. Department of Mathematics and Statistics, Boston University Boston, Massachusetts, 2008.

- [10] Gianluca Fusai and Andrea Roncoroni. *Implementing models in quantitative finance: methods and cases*. Springer-Verlag Berlin Heidelberg, 2008.
- [11] Christian Gourieroux, Alain Monfort, and Alain Trognon. Pseudo maximum likelihood methods: Applications to poisson models. *Econometrica: Journal of the Econometric Society*, pages 701–720, 1984.
- [12] Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.
- [13] Alan G Hawkes and David Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, pages 493–503, 1974.
- [14] Clifford M Hurvich, Jeffrey S Simonoff, and Chih-Ling Tsai. Smoothing parameter selection in nonparametric regression using an improved akaike information criterion. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(2):271–293, 1998.
- [15] John J Leonard and Hans Jacob S Feder. A computationally efficient method for large-scale concurrent mapping and localization. In *ROBOTICS RESEARCH-INTERNATIONAL SYMPOSIUM-*, volume 9, pages 169–178. Citeseer, 2000.
- [16] Kenji Mizuseki, Anton Sirota, Eva Pastalkova, and György Buzsáki. Theta oscillations provide temporal windows for local circuit computation in the entorhinal-hippocampal loop. *Neuron*, 64(2):267–280, 2009.
- [17] Y Ogata. Statistical model for standard seismicity and detection of anomalies by residual analysis. *Tectonophysics*, 169(1):159–174, 1989.
- [18] Yoshiko Ogata. Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association*, 83(401):9–27, 1988.
- [19] Fusakichi Ōmori. *On the after-shocks of earthquakes*, volume 7. The University, 1894.
- [20] Alexandre Pouget, Peter Dayan, and Richard S Zemel. Inference and computation with population codes. *Annual review of neuroscience*, 26(1):381–410, 2003.

- [21] James O Ramsay. *Functional data analysis*. Wiley Online Library, 2006.
- [22] James O Ramsay, Giles Hooker, and Spencer Graves. *Functional data analysis with R and MATLAB*. Springer Science & Business Media, 2009.
- [23] Jerald F. Lawless Richard J. Cook. *The statistical analysis of recurrent events*. Springer, 2007.
- [24] Jerald F. Lawless Richard J. Cook. *The The statistical analysis of recurrent events*. Springer, 2007.
- [25] Sheldon M. Ross. *Simulation*. Amsterdam : Elsevier Academic Press, 2006.
- [26] David W Scott. Multivariate density estimation and visualization. In *Handbook of Computational Statistics*, pages 549–569. Springer, 2012.
- [27] Ela Sienkiewicz, Dong Song, F. Jay Breidt, and Haonan Wang. Sparse functional dynamical models — a big data approach. *Journal of Computational and Graphical Statistics*, 0(ja):0–0, 0.
- [28] Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.
- [29] Donald L Snyder and Michael I Miller. *Random point processes in time and space*. Springer Science & Business Media, 2012.
- [30] Dong Song, Rosa HM Chan, Vasilis Z Marmarelis, Robert E Hampson, Sam A Deadwyler, and Theodore W Berger. Nonlinear dynamic modeling of spike train transformations for hippocampal-cortical prostheses. *IEEE Transactions on Biomedical Engineering*, 54(6):1053–1066, 2007.
- [31] Dong Song, Rosa HM Chan, Vasilis Z Marmarelis, Robert E Hampson, Sam A Deadwyler, and Theodore W Berger. Nonlinear modeling of neural population dynamics for hippocampal prostheses. *Neural Networks*, 22(9):1340–1351, 2009.
- [32] Dong Song, Haonan Wang, Catherine Y Tu, Vasilis Z Marmarelis, Robert E Hampson, Sam A Deadwyler, and Theodore W Berger. Identification of sparse neural functional connectivity using penalized likelihood estimation and basis functions. *Journal of computational neuroscience*, 35(3):335–357, 2013.

- [33] Wilson Truccolo, Uri T Eden, Matthew R Fellows, John P Donoghue, and Emery N Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089, 2005.
- [34] Berwin A Turlach et al. *Bandwidth selection in kernel density estimation: A review*. Université catholique de Louvain, 1993.
- [35] Shuang Wu, Hans-Georg Müller, and Zhen Zhang. Functional data analysis for point processes with rare events. *Statistica Sinica*, pages 1–23, 2013.
- [36] Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional data analysis for sparse longitudinal data. *Journal of the American Statistical Association*, 100(470):577–590, 2005.