Imperial College London

Imperial College London
Department of Mathematics

Statistical Inference for Hawkes Processes with Deep Learning

Tom Keane

CID: 01788365

Supervised by Dr. Seth Flaxman

4th September 2020

Submitted in partial fulfilment of the requirements for the MSc in Statistics of Imperial College London

The work contained in this thesis is my own work unless otherwise stated.

Signed: J Keane Date: 04/09/2020

Acknowledgements

I would like to take this opportunity to thank those without whom this project would not have been successful. This work was made possible by the support and guidance of my supervisor, Dr. Seth Flaxman, and I would like to thank him for giving so much of his time to advising and assisting me despite the challenges posed by remote work. I would also like to thank Dr. Swapnil Mishra for his efforts and expertise which guided me through the unique challenges of my area of study. I would like to thank my family for their unwavering support of me throughout my time at Imperial, and the tough circumstances which arose due to Covid-19.

Finally, I would like to dedicate my work to my late Godmother, Christine Walsh, who passed away after a short battle with cancer in the week preceding the submission of this thesis. May she rest in peace.

Abstract

Hawkes processes and their applications are a fast-growing area of study. A key difficulty for their application to certain areas is imprecise observation leading to censored event times. The resulting processes have been termed as aggregated Hawkes processes. This thesis aims to address statistical inference and parameter estimation for aggregated Hawkes processes. This is achieved using deep learning frameworks such as Variational Auto-Encoders (VAEs) and Neural Networks. Two methodologies are developed in this thesis; one regarding the successful encoding of the aggregated data using a Variational Auto-Encoder and a Poisson likelihood, and the other uses a Multilayer Perceptron to solve the problem of parameter estimation. The successful application of VAEs to aggregated Hawkes processes allowed for Bayesian inference to be performed on the branching ratio, η and the baseline intensity, μ. The potential of this method is demonstrated using four test processes with a range of underlying parameters. The problem of parameter estimation was successfully approached using a blended supervised learning technique. This method was tested comprehensively using a range of parameter values to establish an understanding of its performance. It was concluded that the method developed in this thesis provides similar performance to other solutions but with a significantly reduced computational time.

Table of Contents

1. Introduction **1**

2. Background **2**
2.1. Hawkes Processes . . . 2
2.1.1. Stochastic Point Processes . . . 2
2.1.2. The Hawkes Process . . . 5
2.2. Variational Auto-Encoders . . . 7
2.2.1. Neural Networks . . . 8
2.2.2. General Structure . . . 9
2.2.3. Usage . . . 11
2.2.4. Loss Function . . . 11
2.2.5. Key Points on VAE Training . . . 12

3. Literature Review **15**
3.1. Parameter Estimation for Continuous Data . . . 15
3.1.1. Parametric Estimation . . . 15
3.1.2. Non-parametric Estimation . . . 16
3.2. Parameter Estimation for Discrete Data . . . 17
3.3. Variational Auto-Encoders and Hawkes Processes . . . 20
3.3.1. VAEs for Self Excited Discrete data . . . 20
3.3.2. Bayesian Inference in VAE Frameworks . . . 20

4. Methodology **22**
4.1. Data . . . 22
4.1.1. Data Generation . . . 22
4.2. Bayesian Inference and Variational Auto-Encoders . . . 25
4.2.1. Training . . . 25
4.2.2. Bayesian Inference . . . 30
4.2.3. Testing VAE Performance . . . 31
4.2.4. Super Resolution . . . 32
4.3. Supervised Learning . . . 32
4.3.1. Estimation of Baseline Intensity and Branching Ratio . . . 32
4.3.2. Estimation of Self Excitation and Intensity Decay . . . 33
4.3.3. Testing and Comparison . . . 35
4.3.4. Comparison of Parameter Estimation Methods . . . 36

5. Results & Discussion **37**
5.1. Evaluation of Variational Auto-Encoders . . . 37

5.2. Evaluation of Supervised Learning Method 46
5.2.1. Test of Effect of Parameter Values/Ranges 46
5.2.2. Comparison with MLE and the MC-EM Algorithm 53
5.2.3. Discussion of General Performance and Comparison 55

6. Conclusion **57**

A. Appendix **62**
A.1. Code Repository 62

1. Introduction

The study of Hawkes processes has shown a wealth of potential applications in seismology (Ogata, 1998), criminology (Mohler et al., 2011), and most recently finance (Bacry et al., 2015). The application of Hawkes processes to finance has been a rapidly growing area of study, with this growth expected to continue for many years (Hawkes, 2018). However, since the conception of algorithmic trading, the time scales at which events occur in financial markets have been reducing to near microsecond levels (Perez, 2011). As these time scales have plummeted, the ability to precisely observe event times has diminished due to technological constraints. These short time scales have become inhibitory to the application of Hawkes processes, as it is a self-exciting stochastic point process. This means that two events cannot be observed to have occurred at the same time. This problem of imprecise observation is not limited to finance, and is seen in other areas such as cyber-security (Turcotte et al., 2017), where the monetary cost of precise observation outweighs the benefits it brings. Therefore, this has led to the requirement of parameter estimation techniques for the resulting "aggregated" Hawkes processes. These consist of the number of events over regular disjoint intervals, rather than the precise event times. The work seen in Shlomovich et al. (2020) is a good example of a solution to this problem. Of interest is to define the underlying features of a Hawkes process realisation such as the expected number of events caused by the self-exciting property. Accurate parameter estimation is a key component in understanding these features, and the censoring of event times complicates parameter estimation.

The work in this thesis aims to use deep learning to address the problem of parameter estimation when event times have been censored. This shall first be approached using Variational Auto-Encoders, as the statistical inference framework they provide may allow for the distributions of the conditional intensity function parameters to be understood. Variational Auto-Encoders shall also be applied for process reconstruction with the aim of using frameworks such as π-VAE (Mishra et al., 2020) to interpolate censored event times. The problem shall also be approached as a regression problem using supervised learning techniques such as neural networks.

The background knowledge on Hawkes processes and Variational Auto-Encoders necessary for subsequent chapters shall be provided in the next chapter. This shall be followed by a comprehensive literature review of the Hawkes process parameter estimation problem when the events times are censored as well as when they are not. Then, the novel methodologies derived in this paper which apply deep learning to this problem shall be presented. This thesis is then concluded with the results of these methodologies and the subsequent discussion regarding their performance.

2. Background

The aim of this chapter is to establish the prerequisite knowledge regarding Hawkes processes, neural networks, and Variational Auto-Encoders (VAEs). First, the necessary properties and definitions relating to the Hawkes process are established. This is followed by the necessary definitions and properties of VAEs and neural networks. This knowledge is required to comprehend the methodologies presented in Section 4.

2.1. Hawkes Processes

The Hawkes process is a self-exciting point process which was defined in Hawkes (1971). Before defining the Hawkes process, some elementary knowledge of stochastic processes is required and some motivation for the self-exciting property is presented.

2.1.1. Stochastic Point Processes

The Hawkes process is a point process which is a form of stochastic process. Temporal point processes are often used to describe the times at which random events occur, with t_i indicating the time at which the ith random event occurred. For example, define a vehicle passing a point on a road as a random event. The sequence of times at which a vehicle passes this point, form a realisation of a temporal point process. The random events are sometimes referred to as arrivals and the time at which they occurred are referred to as arrival times. A formal definition of a general point process can be seen in Chapter 2 of Last and Penrose (2017), however it is sufficient for this thesis to understand the form of a temporal point process realisation.

Let $T_1, T_2, \dots$ be random variables occurring according to a point process. Let $t_1, t_2, \dots$ denote realisations from that point process. Realisations of a point process are commonly represented in three ways:

1. A series of event times; $\{t_i\}_{i=1,\dots,n}$
2. A series of waiting times; $\{\tau_i\}_{i=1,\dots,n}$ where $\tau_i = t_i - t_{i-1}$ and $t_0 = 0$
3. A series of event counts over the interval $(0, t]$; $\{N(t)\}_{t\geq 0}$

Note that given the event times, the waiting times can be calculated and vice versa. The third form presented is referred to as a counting process and the formal definition is presented here:

Definition 2.1.1 (Counting Process)**.** *A counting process is a stochastic point process, $\{N(t)\}_{t\geq 0}$, which has the following properties:*

1. $N(t) \geq 0$ *(non negative)*

2. $N(t) \in \mathbb{Z}$ *(integer valued)*

3. $N(t) \leq N(s) \quad \forall\, t \leq s$ *(non-decreasing)*

It should also be noted that for $t < s$, the number of random events occurring over the interval $(t, s]$, is equal to $N(s) - N(t)$. Therefore, a counting process can be represented by a chosen number of increments. An increment is the value of the counting process over disjoint subsets of the interval.

For example, let $\{N(t)\}_{t\geq 0}$ denote a counting process on the interval $(0, T]$, with an underlying point process. This process can be represented by the desired number of increments, n, using an increment length of $\Delta = T/n$. The increments can be defined as follows:

$$N_i^{(\Delta)} = N(\Delta i) - N(\Delta(i-1)) \quad i = 1, \ldots, n$$

Point processes are often observed as increments of a counting process when the observation of event times is impeded. This observation may be impeded by limits of technology or costs of observation and therefore is common when events occur on very small time scales such as in finance or cyber-security (Turcotte et al., 2017). There is an inherent loss of information in this case. One of the focuses of this thesis is to overcome this loss of information. However, it should be noted that this loss of information is directly linked to the size of increment length Δ. As $\Delta \downarrow 0$, the counting process increments will allow the underlying point process to be recovered.

The most widely known example of a counting process is a Poisson process. The fundamental property of a Poisson process is that the number of events over an interval of length t, follows a Poisson random variable with rate λt. This can be expressed as:

$$\mathbb{P}(N(t) = n) = \frac{(\lambda t)^n}{n!} \exp(-\lambda t) \tag{2.1}$$

Eq. (2.1) defines a **homogeneous** Poisson process due to its constant rate parameter $\lambda > 0$. This rate parameter is also known as the intensity of the Poisson process. This leads to the formal definition:

Definition 2.1.2 (Homogeneous Poisson Process)**.** *A counting process is a homogeneous Poisson process if the following properties hold:*

1. $N(0) = 0$

2. $N(t)$ *has independent increments*

3. *The number of events in an interval of length t follows a Poisson random variable with intensity* λt*.*

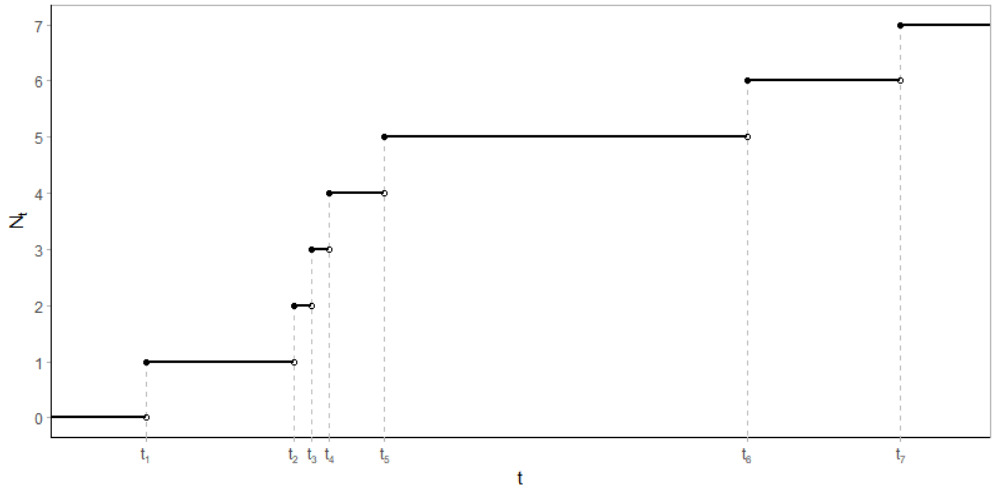


Figure 2.1.: Plot of a homogeneous Poisson process

A homogeneous Poisson process is a limiting assumption in many cases. Building on the earlier example where a random event was defined as a vehicle passing a point on a road. This would be said to follow a homogeneous Poisson process if the number of vehicles which pass that point over the course of a minute/hour/day, is a Poisson random variable with constant intensity. The homogeneity property means the expected number of vehicles which shall pass the point at rush hour between 08:00 and 09:00 and during the night between 23:00 and 00:00, will be equal. This would be a reasonable assumption if the road was not regularly used. However, this would be an unreasonable assumption given a main road into a busy city.

This leads to the **inhomogeneous** Poisson process which has a variable intensity defined by a function of time $\lambda(t)$. For an inhomogeneous Poisson process, the number of events over an interval $(0, t]$, follows a Poisson random variable with rate equal to the integral of the intensity function over the interval: $\int_0^t \lambda(s)\,\mathrm{d}s$. Therefore, the constant intensity λt in Eq. (2.1), is replaced by this integral to give:

$$\mathbb{P}(N(t) = n) = \frac{(\int_0^t \lambda(s)\,\mathrm{d}s)^n}{n!} \exp\left(- \int_0^t \lambda(s)\,\mathrm{d}s \right) \tag{2.2}$$

The formal definition is as follows:

Definition 2.1.3 (Inhomogeneous Poisson Process)**.** *A counting process is an inhomogeneous Poisson process with intensity $\lambda(t)$ if the following properties hold:*

1. *$\lambda(t)$ is an integrable function*
2. *$N(0) = 0$*
3. *$N(t)$ has independent increments*

4. for any $t \in [0, \infty)$

$$\begin{aligned}
\mathbb{P}(N(t+\Delta) - N(t) = 0) &= 1 - \lambda(t)\Delta + \mathrm{o}(\Delta) \\
\mathbb{P}(N(t+\Delta) - N(t) = 1) &= \lambda(t)\Delta + \mathrm{o}(\Delta) \\
\mathbb{P}(N(t+\Delta) - N(t) \geq 2) &= \mathrm{o}(\Delta)
\end{aligned}$$

Remark 2.1.4. *The notation seen above,* $\mathrm{o}(\cdot)$*, is commonly referred to as 'little o' notation. It can be defined as follows:*

$$f(n) = \mathrm{o}(g(n)) \quad \iff \quad \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

2.1.2. The Hawkes Process

The Hawkes process is known as a self-exciting point process. This means the Hawkes process falls into a category of point processes which are defined by their conditional intensity. The conditional intensity can intuitively be known as the expected arrivals conditional on a process' history, $\mathcal{H}_t$. Formally, $\mathcal{H}_t$ is a filtration, however intuitively it is the representation of arrivals up to, but not including, time t. The conditional intensity can be defined on a combination of the conditional density $f(t|\mathcal{H}_t)$ and conditional distribution $F(t|\mathcal{H}_t)$ or directly on the conditional expectation. This leads to the formal definition of conditional intensity:

Definition 2.1.5 (Conditional Intensity function)**.** *Consider a count process,* $N(t)$*, with history,* $\mathcal{H}_t$*. Given the conditional density,* $f(t|\mathcal{H}_t)$*, and the conditional distribution,* $F(t|\mathcal{H}_t)$*, the conditional intensity function of the underlying point process can be defined as:*

$$\lambda^*(t) = \frac{f(t|\mathcal{H}_t)}{1 - F(t|\mathcal{H}_t)}$$

Alternatively, the conditional intensity can also be defined as:

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

Here, $\lambda^*(t)$ is a non-negative function which defines the intensity of the point process based on the history of points seen up to but not including the present. Point processes which are defined on their conditional intensity can fall into two categories; self-exciting or self-correcting. The former relates to a process which has an increase in intensity given an arrival and the latter relates to a process which has a decrease in intensity given an arrival. As mentioned previously, the Hawkes processes is a self-exciting process. This leads us to the definition of the Hawkes process:

Definition 2.1.6 (Hawkes Process)**.** *Consider a counting process,* $N(t)$*, with associated history,* $\mathcal{H}_t$*, and conditional intensity function,* $\lambda^*(t)$*. Suppose that for any* $t \in [0, \infty)$

the following holds:

$$\begin{aligned}
\mathbb{P}(N(t+\Delta) - N(t) = 0) &= 1 - \lambda^*(t)\Delta + \mathrm{o}(\Delta) \\
\mathbb{P}(N(t+\Delta) - N(t) = 1) &= \lambda^*(t)\Delta + \mathrm{o}(\Delta) \\
\mathbb{P}(N(t+\Delta) - N(t) \geq 2) &= \mathrm{o}(\Delta)
\end{aligned}$$

If the conditional intensity function, $\lambda^*(t)$, is of the form:

$$\lambda^*(t) = \mu + \int_0^t k(t-u)\,\mathrm{d}N(u) \tag{2.3}$$

where $\mu \geq 0$ and $k : (0,\infty) \mapsto [0,\infty)$, then $N(t)$ is said to be a Hawkes process

From the above definition of the Hawkes process, it can be seen that the form of the conditional intensity function is its defining feature. The conditional intensity of a Hawkes process can be split into two terms. The first term, μ, is known as the background intensity, and the second term, $k(\cdot)$, is known as the excitation function. In recent literature, this excitation function often takes the form of a shift invariant kernel and this form is assumed in Eq. (2.3), however this is not a requirement.

Comparing the definition of the Hawkes process to that of the inhomogeneous Poisson process, a link between the two can be seen. In particular, the incremental counts of a Hawkes process on the interval $(s,t]$ follows a Poisson random variable with intensity as given in Eq. (2.4).

$$\int_s^t \left[\mu + \int_0^v k(v-u)\,\mathrm{d}N(u)\right] \mathrm{d}v \tag{2.4}$$

This leads to the conditional intensity function being considered in two parts, the homogeneous baseline intensity and the inhomogeneous intensity from self-excitation. Setting the excitation function to 0, leads to $\lambda^*(t) = \mu$. This represents a constant conditional intensity function which is equivalent to a homogeneous Poisson process. This allows for a Hawkes process to be viewed as an immigrant-birth process. In this framework, the points in a Hawkes process can be classified as either an immigrant or a descendant. Immigrants are events which are attributed to the background intensity. Descendants occur due to the self-exciting property, these are considered "births". For example in the case of seismology, the initial earthquake would be considered an immigrant into the point process, however any aftershocks are considered descendants of the initial earthquake. The form of the kernel function strictly defines the properties of the Hawkes process; the base form proposed in Hawkes (1971) is as follows:

$$k(t-u) = \alpha \exp(-\beta(t-u)) \tag{2.5}$$

Other forms of kernel functions are regularly used but this is application dependent and unless specified otherwise, the kernel function shall be assumed to be of the form seen in Eq. (2.5). A realisation of a Hawkes process is seen in Figure 2.2.

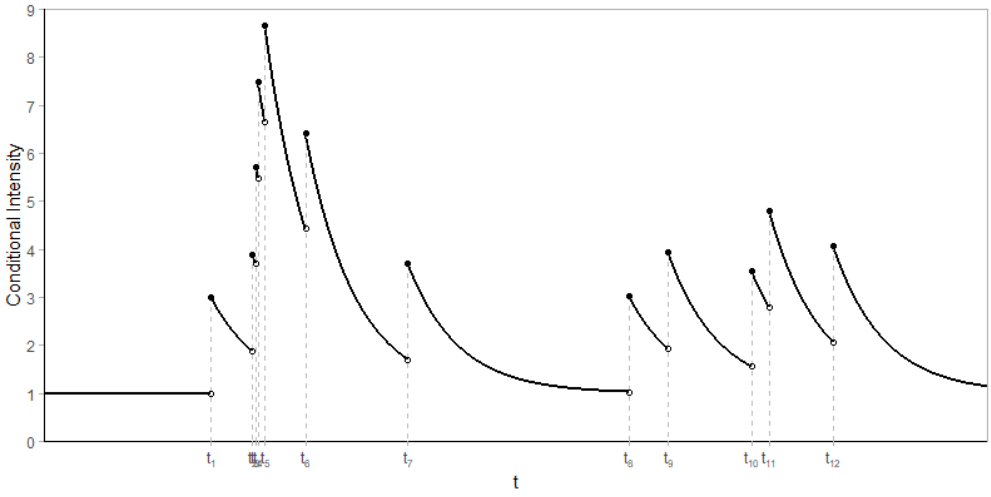


Figure 2.2.: Plot of a Hawkes process with parameter $\{\alpha, \beta, \mu\} = \{2, 2.5, 1\}$

Understanding the self-exciting properties which are implied by the kernel function choice is an important area of study. Of particular interest in later sections in the branching ratio, η. This branching ratio is defined as the expected number of first generation descendants caused by an event. This is equal to the integral of the kernel function over the interval $[0, \infty)$ which is as follows:

$$\eta = \int_0^\infty \alpha \exp(-\beta(s))\, \mathrm{d}s = \frac{\alpha}{\beta}$$

As discussed in Laub et al. (2015), the branching ratio is very important for the simulation of Hawkes processes. It is required for simulation of Hawkes processes that $\eta < 1$. The reason for which can be seen by considering the total expected number of descendants for an event. As shown in Laub et al. (2015), this is $\frac{\eta}{1-\eta}$ in the case where $\eta < 1$ and ∞ when $\eta \geq 1$. Henceforth, η will be assumed to be less than 1.

In conjunction with defining the number of expected descendants, the total number of expected events is given in Laub et al. (2015) as

$$\mathbb{E}(\lambda^*(t)) = \frac{\mu}{1-\eta} \tag{2.6}$$

This relationship will be heavily relied upon in the data generation process seen in later sections.

2.2. Variational Auto-Encoders

Auto-encoder Variational Bayes (AEBV) was first proposed in Kingma and Welling (2013) as a variational Bayesian approach for inference when certain distributions are intractable. Variational Auto-Encoders (VAE) are a special case of the AEBV, where deep learning is combined with the proposed inference framework. This method grew

in popularity with applications in image compression, image de-noising, and Natural Language Processing. In particular, a generative model which is provided by VAEs is of interest as it allows for variations of observed data to be generated. In the case of de-noising images, the variations of an image generated aim to contain less noise that the original observed image. This is achieved by compressing the input into a lower dimensional space. When the image is being compressed to a lower dimension, some information is inherently lost. As the noise in the image does not contribute to the reconstruction performance, it is among the first information to be lost during compression. This complex task is accomplished using deep learning and some pre-requisite knowledge on neural networks is required. The required background for neural networks is presented next, followed by an explanation of the mathematics of VAEs.

2.2.1. Neural Networks

Neural networks have become synonymous with deep learning in recent years and its variations such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) are regularly used for complicated problems such as computer vision. However unless otherwise stated the term neural network in this thesis refers to its simplest sequential form known as a multilayer perceptron.

Multilayer Perceptrons

Multi-Layer Perceptrons (MLP) are a sequential feed forward network. An MLP is designed to loosely replicated a brain's neural structure. This is achieved by constructing layers of "neurons". There are three types of layers in an MLP. The first is an input layer and is used to enter data into the network. The second type of layer is known as a hidden layer. As the name indicates the input into these layers is usually unseen. Data is passed from the input layer to a user specified number of hidden layers, each with a user specified number of neurons. Every neuron in a given layer is connected to every neuron in the previous layer. The input is passed through the hidden layers to the output layer which has the same number of neurons as the desired dimension of the output.

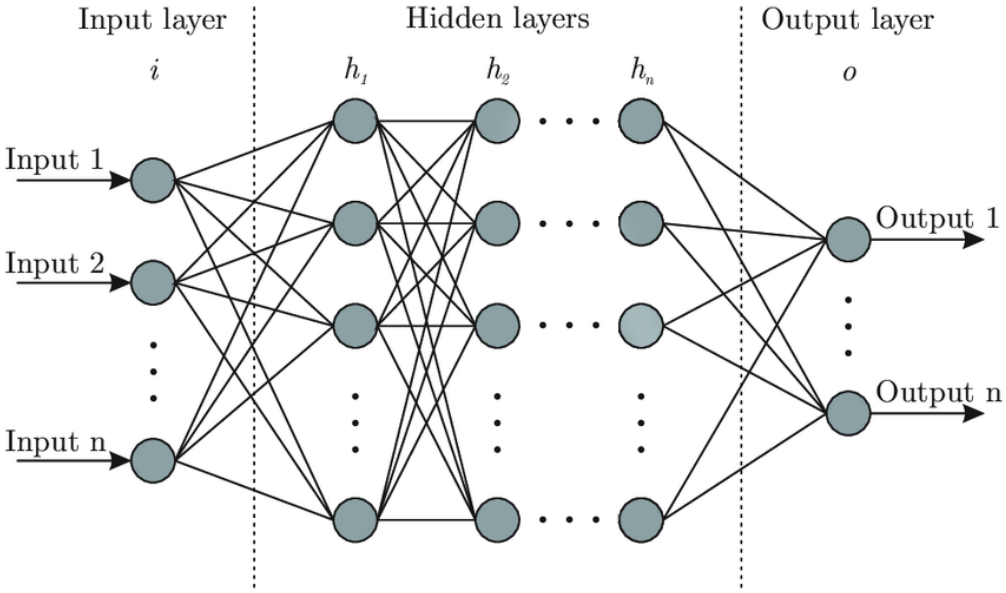


Figure 2.3.: Generic form of a Multilayer Perceptron (Adapted from (Bre et al., 2018))

As can be seen in the Figure 2.3, each neuron in a layer, l, is connected to each neuron in the previous layer, $l-1$. This is achieved using a linear combination of the values of the neurons connected to it. The coefficients of these linear combinations are unique to each neuron and are known as weights. The intercept coefficient is known as the bias. After the linear combination of each layer has been evaluated, a transformation is applied. This transformation is known as an activation function. The activation function can range from the identity function to non-linear functions which restrict the range of values the output can take. This allows an MLP replicate more complex functions. Neural networks use the minimization of a loss function to make iterative changes to the weights and biases within the network. This allows the network to learn the function it is trying to replicate. This process is known as back propagation and is a gradient based optimisation algorithm. The gradient of each layer is calculated recursively using the chain rule which allows the gradient be calculated back through the network.

2.2.2. General Structure

Variational Auto-Encoders have become incredibly popular due to the diverse applications in many fields, particularly the field of computer science where it is used to both de-noise and compress images. Auto-Encoders, which preceded VAEs, are a variation of an MLP which passes an input of dimension, d, through multiple hidden layers until a "neuron bottleneck" is reached, i.e. a hidden layer with d' neurons and $d' \ll d$. From there, the network reconstructs the output to its original dimensionality. The output to the neuron bottleneck is considered a lower dimensional representation of the input. The neural network is usually symmetric around the neuron bottleneck. Therefore, the neural network 'encodes' the data into a lower dimensional representation and then 'de-

codes' the lower dimensional representation back to the original input. This leads to the terminology of encoder and decoder, which represent the portion of the network before and after the neuron bottleneck respectively.

The form of a VAE is very similar, with an encoder and a decoder portion, however a VAE does not encode the lower dimensional representation to a single neuron bottleneck. Instead, a random sampling layer is included, this allows the VAE to encode the input into a continuous lower dimensional latent variable, z. This lower dimensional latent variable is usually taken to be a Multivariate Normal (MVN). The continuity of this latent variable is crucial to many uses of VAEs as the decoder forms part of the generative model. This generative model allows for sampling from the joint distribution of the data and the latent variable, $p(x, z)$.

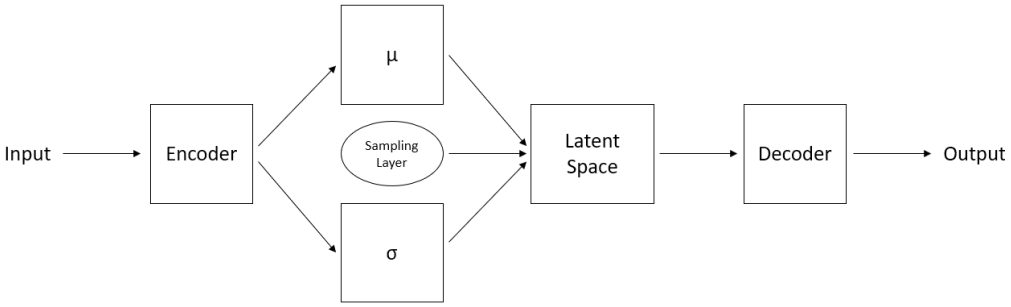


Figure 2.4.: Basic Structure of a VAE with a MVN latent variable z

Encoder

The encoder consists of a feed forward neural network which reduces the dimensionality of the input down to the dimensionality of the chosen latent space $\mathcal{Z}$. Due to the form of the cost function used for training the neural network, the encoder becomes an approximation of the intractable or unknown posterior, $p_\theta(z|x)$. With a chosen structure for the encoder neural network, and hyper-parameters ϕ, the encoder represents:

$$q_\phi(z|x) \approx p_\theta(z|x)$$

A MVN latent variable, z, leads to $z|x \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$. Unless otherwise stated, the latent variable will be assumed to be MVN and $\boldsymbol{\sigma}$ will be a diagonal matrix meaning that the latent dimensions are assumed to be independent. A fully trained encoder will output the mean and log standard deviation of the latent variable given a certain input, x. This can be expressed as follows:

$$\begin{aligned} \text{Encoder}(x) &\rightarrow (\boldsymbol{\mu}, \log(\boldsymbol{\sigma})) \\ q_\phi(z|x) &:= \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}) \end{aligned}$$

Decoder

Like the encoder, the decoder is a feed forward neural network. However, while the encoder allows an input, x, to be encoded into the latent space, the decoder takes an

observation from the latent space and reconstructs the input. Therefore, the structure of the decoder is usually the reverse of the encoder i.e. if the encoder has an input layer, two hidden layers and an output layer, with 100, 32, 16, and 2 neurons respectively, then the decoder will also have an input layer, two hidden layers and an output layer, but with 2, 16, 32, and 100 neurons respectively.

The advantage of a VAE versus other methods is the ability to construct a generative model. In particular, the decoder represents $p_\theta(x|z)$. Using the prior for z, the joint distribution can be constructed using the following factorisation:

$$p_\theta(x, z) = p_\theta(z)p_\theta(x|z)$$

As mentioned previously, the latent space is typically assumed to be MVN, in this case the prior for the latent variable z is assumed to be $\mathcal{N}(\mathbf{0}, \mathbb{I})$. Therefore, reconstructed inputs can be sampled by first drawing $z \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ and passing the sampled values through the decoder to give a sample from the joint distribution.

2.2.3. Usage

Consider a dataset $X = \{x_i\}_{i=1}^N$, where x_i is an i.i.d sample of some continuous or discrete variable x. The process is assumed to be generated by a random process involving an unobserved continuous random variable z. The aim of AEVB is to provide a method to find the optimal parameters θ when the marginal log likelihood $p_\theta(x)$, or posterior $p_\theta(z|x)$, is intractable or computationally expensive to compute. The intractability rules out frequential methods such as maximum likelihood estimation, and expensive computation which may be caused by large datasets, means sampling based solutions such as MC-EM would be too expensive. The key to the ability of AEVB and VAEs is avoiding these intractabilities or expensive computation through the form of the loss function.

2.2.4. Loss Function

As outlined in Section 2.2.3, VAEs are used when the log marginal likelihood is intractable or computationally expensive. The loss function used for VAEs, which is known as the Evidence Lower Bound (ELBO), forms a lower bound for $\log p_\theta(x)$ (Kingma and Welling, 2013). As such it allows for the maximisation of the log marginal likelihood without the need for tractability.

The ELBO has the following forms (Kingma and Welling, 2019):

$$= \mathbb{E}_{q_\phi(z|x)}[\ \log p_\theta(x|z)\ ] - D_{KL}(q_\phi(z|x)\ ||\ p_\theta(z|x)) \tag{2.7}$$

$$= \mathbb{E}_{q_\phi(z|x)}[\ \log p_\theta(x|z)\ ] - D_{KL}(q_\phi(z|x)\ ||\ p_\theta(z)) \tag{2.8}$$

The ELBO seen in Eq. (2.7) is comprised of two parts, the first is the expectation of the reconstructed inputs under the likelihood $p_\theta(x|z)$. This term is equivalent to the log marginal likelihood and is maximised as a measure of the accuracy of the reconstruction of the inputs. The second term denoted D_{KL} is known as the Kullback-Leibler (K-L) divergence and is a measure of how similar two distributions are. A key result of

a VAE is the generative model which is achieved through a continuous distribution of the latent variable in the latent space. The K-L divergence term ensures the continuity of the latent space. It also acts as a measure of divergence between the approximate posterior represented by the encoder, $q_\phi(z|x)$, and the posterior of interest, $p_\theta(z|x)$. The K-L divergence term is non negative and is equal to zero if and only if the two distributions are the same. Therefore, the tightness of the bound to $\log p_\theta(x)$ is dictated by the K-L divergence term. The maximisation of $\log p_\theta(x)$ ensures a good generative model (Kingma and Welling, 2019) and therefore the ELBO forms a two for one. By minimising the K-L divergence term, the approximate posterior becomes a better approximation for the true posterior, and also ensures the tightness of the bound on the log marginal likelihood. As a final note, as can be seen in Kingma and Welling (2013), $D_{KL}(q_\phi(z|x) \,||\, p_\theta(z|x)) = D_{KL}(q_\phi(z|x) \,||\, p_\theta(z))$ which leads to the form of the ELBO in Eq. (2.8). This second form is the most commonly used and it is important to note that it is equivalent to that of Eq. (2.7).

Optimisation of Loss Function

For a dataset with i.i.d data, $\mathcal{D}$, the ELBO of this dataset is calculated using the sum of its value of each individual data point:

$$\mathcal{L}_{\theta,\phi}(\mathcal{D}) = \sum_{x_i \in \mathcal{D}} \mathcal{L}_{\theta,\phi}(x_i)$$

However, the gradient of this function is generally intractable and as neural networks are optimised using back propagation, it should be noted that this back propagation cannot occur through a random sampling layer. These problems are solved using the reparameterisation trick (Kingma and Welling, 2013). Rather than sampling the latent variable directly, $z \sim q_\phi(z|x)$, it is reparameterised to a differentiable function with a noise variable. In the case of a MVN latent variable, the following reparameterisation is used:

$$\begin{aligned} z|x &\sim \mathcal{N}(\boldsymbol{\mu}_{z|x}, \boldsymbol{\sigma}_{z|x}) \\ z &= g_\phi(\epsilon, x) \\ &= \boldsymbol{\mu}(x) + \boldsymbol{\sigma}(x)\,\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbb{I}) \end{aligned}$$

This reparameterisation technique, which was derived in Kingma and Welling (2013), allows for the expression of z as a deterministic function of x which allows for a differentiable Monte Carlo estimate for the first term of Eq. (2.8).

2.2.5. Key Points on VAE Training

As seen in Section 2.2.4, the optimisation problem of a VAE is the minimisation of the negative of the ELBO seen in Eq. (2.8). This term is composed of the reconstruction loss through the expectation of the likelihood and the restriction of the latent variable through the K-L divergence term. The K-L divergence term ensures a continuous latent

variable in the latent space which is important for a good generative model. Considering this, there are four items which must be specified for the training of a VAE which are as follows:

1. The form of the likelihood, $p_\theta(x|z)$.
2. The distribution of the latent variable, z, and its prior $p_\theta(z)$. This includes the specification of the number of latent dimensions.
3. The structure of the neural networks and associated hyper-parameters used as the encoder and decoder.
4. The training scheme used.

Likelihood Specification

The most important decision is the form of the likelihood, as this defines the objective of the reconstruction. The choice of likelihood should reflect the expected form of the data, $\{x_i\}_{i=1,\dots,n}$. In Kingma and Welling (2013), a Gaussian likelihood is suggested for continuous data and a Bernoulli likelihood is suggested for binary data. This choice is highly dependent on the objectives of the application and many well known distributions have been used in VAEs. For example, the multinomial VAE derived in Liang et al. (2018) which assumes a multinomial likelihood and is commonly used for text classification. It is worth noting that in methodology outlined in Kingma and Welling (2013), the output of the decoder is not a direct reconstruction of the input but is the appropriate parameters of the likelihood distribution.

Latent variable

For most applications of VAEs, the latent variable is taken to be MVN with a prior specification of a standard normal. As can be seen in Kingma and Welling (2013), this is not a requirement and details on non-normal latent variables can be seen in Kingma and Welling (2019). The assumption of a standard normal prior, i.e. $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}\mathbb{I})$, incorporates the assumption of independent latent dimensions due to the diagonal covariance matrix. The remaining choice regarding the latent variable is the number of dimensions. The number of dimensions of the latent variable is an important hyper-parameter to be considered. The dimensionality directly impacts the quality of the reconstruction, with higher dimensions allowing for a more expressive latent variable. It is worth noting that the latent representation is viewed as a dimensionality reduction of the key components of the input. This leads to a a higher dimensional latent variable allowing for more information to be encoded into this representation. However, as indicated in the β-VAE framework derived in Higgins et al. (2016), using less dimensions by increasing the weight of the K-L divergence term increases the VAEs ability to learn meaningful latent variables. Therefore the decision regarding latent dimensions is best made in conjunction with the training scheme used and updated according to initial results.

Structure & Training scheme

The structure of the neural networks used for both the encoder and decoder requires a few considerations. The first consideration is the number of layers and neurons. In particular, the total number of neurons is restricted to prevent over-fitting. The choice for these hyper-parameters for the applications seen in this thesis shall be discussed in later sections.

While it is standard to mirror the structure of the encoder onto the decoder it is not a requirement. It should be noted that the encoder and decoder are not restricted to MLPs with both recurrent and convolutional networks being regularly used and the resulting VAEs are referred to as auto-regressive VAEs. With the nature of Hawkes processes being self-exciting, which could be considered an auto-regressive property, then an auto-regressive VAE may have perform well. However, due to the time constraints imposed on this thesis and the additional problems with training auto-regressive VAEs (Bowman et al., 2015), they are not considered.

When initialising the training of a VAE it is common for the K-L divergence term to offer the largest reduction in the cost function. This leads to a standard VAE implementation restricting the latent space to the prior specification before encoding meaningful representations. This can be addressed through annealing the K-L divergence term in the loss function. This annealing process refers to increasing the weight of the K-L divergence term from 0 to 1 across epochs. For example, a monotonic annealing scheme is suggest in Bowman et al. (2015) to reduce the initial focus on the continuity of the latent space and ensure that a meaningful representation of the data is learned. Typically this scheme increases the KL divergence term in a linear or logarithmic fashion. Since the initial developments in Bowman et al. (2015), the development of cyclical annealing was outlined in Fu et al. (2019). In this scheme, the weight of the K-L divergence term is cycled through repeated monotonic schedules. This can be viewed as allowing the algorithm to attempt many "warm starts" in an effort to ensure a continuous latent space with meaningful embeddings. Annealing can be combined with the theory behind β-VAE developed in Higgins et al. (2016), which sees a large weight being placed on the K-L divergence term to ensure the minimal number of latent dimensions are used. This means more useful information is encoded into the latent dimensions. This can be likened to regularisation such as ridge or lasso regression which ensures a minimal number of features are used in a regression problem. When choosing a training scheme, the number of epochs required will depend on the complexity of the problem, as well as the annealing scheme implemented.

3. Literature Review

As covered in Section 2.1, a Hawkes process is defined by the form of its conditional intensity function. Therefore, accurate estimation of the parameters of this function are of interest as it allows for an understanding of the event frequency. In particular, it allows for an understanding of the effect of endogenous (births) and exogenous (immigrants) influences and events. Parameter estimation is a difficult task when the event times of a process **are** available. It is even more difficult when the precise event times are not available. Information is lost in this case due to intermittent observation (discretisation) or imprecise observation (rounding). Note that data generated by intermittent and imprecise observation should be approximately equivalent assuming the underlying mechanisms are random. This review first covers methods when precise observation or recording is available (continuous data) and then methods when observation or recording is imprecise (discrete data). The continuous data will be in the form of precise event times and the discrete data will be in the form of an incremental count process at regular observation intervals.

3.1. Parameter Estimation for Continuous Data

In general, the focus of this thesis shall be on parametric methods and in particular on parameter estimation for Hawkes processes with the exponential kernel given in Eq. (2.5). A range of methods are available for continuous data, with many modern methods being based on non-parametric estimation of the intensity function. These non-parametric methods are incredibly flexible as they allow for estimation of the conditional intensity function without any restrictions on the form the function may take.

3.1.1. Parametric Estimation

Maximum Likelihood Estimation (MLE)

Maximum likelihood estimation has been the standard for parameter estimation for Hawkes processes with a range of parametric kernels (Bacry et al., 2012). It was first outlined in Ozaki (1979). The log likelihood was given in terms of a general kernel function in Rubin (1972), the derivation of which can be seen in Laub et al. (2015). The general log likelihood is defined as follows:

Definition 3.1.1 (Log likelihood of the Hawkes' model)**.** *Given a Hawkes process with observation times,* $\{t_i\}_{i=1,\dots,n}$ *for an interval* $[0, T]$ *the log likelihood of a Hawkes process*

with conditional intensity function, $\lambda^*(t)$, is given as:

$$\ell(t_1, \ldots, t_n) = -\int_0^T \lambda^*(s)\,\mathrm{d}s + \int_0^T \log \lambda^*(s)\,\mathrm{d}N(s)$$

Using the above definition, the gradients and the hessian matrix of the log likelihood were derived in Ozaki (1979). Then, non-linear optimisation was performed using Newton-Raphson. Note: when applying this method any modern method for non-linear optimisation would be valid. This generalised version of MLE is quadratic in computational time. A recursive method for the exponential kernel, which is linear in computational time, was derived in Ozaki (1979) and presented in detail in Laub et al. (2015). A recursive method is only available for the exponential kernel, meaning this speed up is not present for other kernel choices. As the focus of the subsequent chapters is the exponential kernel, the recursive form of the log-likelihood is given in Eq. (3.1).

$$\ell(t_1, \ldots, t_n) = \sum_{i=1}^{n} \log(\mu + \alpha A(i)) - \mu t_n + \frac{\alpha}{\beta} \sum_{i=1}^{n} \left[\exp(-\beta(t_n - t_i)) - 1 \right] \tag{3.1}$$

where $A(i) = \exp(-\beta(t_i - t_{i-1}))(1 + A(i-1))$

Parametric Expectation Maximisation

A parametric Expectation Maximisation (EM) algorithm was seen for the exponential kernel in Lewis and Mohler (2011) following a method derived in Veen and Schoenberg (2008). This method relied on an expectation step to calculate the probability an event i is a descendent of event j or is an immigrant. This is followed by a maximisation step updating the current estimates for the conditional intensity function. An alternative parametric EM algorithm is outlined in Olson and Carley (2013). No comparison between the performance of these methods or their performance against the maximum likelihood estimator is available. However it should be noted that the maximum likelihood estimation appears to be favoured for parametric estimation problems.

3.1.2. Non-parametric Estimation

Non-parametric estimation for Hawkes processes encapsulates methods which do not assume a form of the kernel function in the conditional intensity function. As seen in the previous section, the parametric methods assumed the form of kernel to be as in Eq. (2.5). The lack of assumption means that non-parametric methods are more flexible, however this usually comes at a cost of accuracy. Therefore, should the form of the kernel be known, it is usually best to use a parametric method.

Non-parametric Expectation Maximisation

An early non-parametric method in the form of an EM algorithm known as Model Independent Stochastic Declustering (MISD) had been previously proposed in Marsan and Lengline (2008). This method was derived for homogeneous background intensities and

non-parametric kernels. This method is similar in methodology to Veen and Schoenberg (2008), as the expectation step includes estimating the probability an event i is a descendent of event j or is an immigrant. The maximisation step then requires the estimation of the constant background intensity and a piece-wise constant estimate of the non-parametric kernel function is calculated. This is repeated until convergence.

Another non-parametric EM algorithm was developed in Lewis and Mohler (2011). This EM algorithm utilises a penalised likelihood method and was designed to estimate a Hawkes process with a varying baseline intensity in conjunction to the non-parametric kernel function. This method added a penalty term to both the function being optimised for the varying background intensity and the non parametric kernel. These penalty terms allow for the minimisation problem to be re-framed as an ordinary differential equation which is discretised and solved iteratively. The flexibility of this approach makes it attractive when no information about the form of either the baseline intensity or the kernel function is available.

Non-parametric estimation using first and second order statistics

A non-parametric method for Hawkes processes with symmetric kernel matrices was first proposed by Bacry et al. (2012). Note: The kernel function is replaced with a kernel matrix in the multi-variate setting. A subsequent method which did not rely on the symmetry of the kernel matrix was proposed in Bacry and Muzy (2014b) and extended in Bacry and Muzy (2014a) and Bacry and Muzy (2016). The first method relied on linking the auto-covariance matrix at a given lag to the kernel matrix through a Fourier transform. The requirement for the kernel matrix to be symmetric only affects the multivariate case and not the univariate case discussed in this thesis. The method proposed requires the choice of two hyper-parameters for the empirical estimation of the covariance matrix. The hyper-parameters are the lag and scale of the covariance matrix, where the lag relates to the lag of the auto-covariance and the scale, which can be viewed as the discretisation of the kernel function. In the conclusion of Bacry et al. (2012), this method is suggested to be reliable for series of greater than 10^5 events and to be used as an initial exploration of kernel shape before proceeding to a classical parametric method such as MLE. The extensions made in Bacry and Muzy (2014a) allowed for the symmetric kernel requirement to be dropped. This method uses the matrix of conditional expectation, which can be empirically estimated following results in Bacry et al. (2012), and required the use of a Nystrom method on the resulting equations.

3.2. Parameter Estimation for Discrete Data

This section outlines methodologies which could be directly applied to the problem of parameter estimation for aggregated Hawkes Processes.

Non Parametric Estimation using Auto-Regressive Timeseries

In Kirchner (2016), the weak convergence of an Integer valued Auto-Regressive (INAR) timeseries with infinite lags to a Hawkes process was shown. This initial work led to a non parametric method which was presented in Kirchner (2017) which relies on discretising the Hawkes process into small bins of size Δ. The infinite auto-regressive term was approximated using p auto-regressive lags. The discretisation is usually performed on the realisations of a Hawkes process but this results in a method which, with caveats, can be used on discrete data. The approximation was made in three steps which are outlined as follows:

$$\begin{aligned}\mathbb{E}(N_i^{(\Delta)}|\mathcal{H}_{i-1}^{(\Delta)}) &\approx \Delta\mu + \Delta\int_0^{(i-1)\Delta} k(i\Delta - u)N(\mathrm{d}u) \\ &\approx \Delta\mu + \Delta\int_{(i-p-1)\Delta}^{(i-1)\Delta} k(i\Delta - u)N(\mathrm{d}u) \\ &\approx \Delta\mu + \sum_{n=1}^{p} \Delta k(\Delta n)N_{i-n}^{\Delta}\end{aligned}$$

Where $\mathcal{H}_{i-1}^{(\Delta)}$ denotes the history of incremental counts up to the $i-1$ bin. With each approximation above, there is a corresponding source of error which was well documented in Kirchner (2017). For this method, the largest source of error comes from the first approximation, where the kernel function is assumed to be piece-wise constant. As a result of this distributional error, events which occur in the same bin cannot be conditional on each other, i.e. an event in a bin cannot be said to have triggered another event in the same bin.

To estimate the parameters of the INAR(p) process, conditional least squares was used which results in the following estimation process for $\{N_i^{(\Delta)}\}_{i=1,\dots,n}$:

1. Choose large p, with $p < n$. This represents the number of bins in the piece wise constant estimate of the kernel function
2. Calculate the conditional least squares estimators: $\theta = YZ^T(ZZ^T)^{-1} \in \mathbb{R}^{(p+1)}$
3. The first p entries in the resulting vector represent the piecewise constant estimate of the kernel, with the last entry representing the baseline intensity estimate
4. To complete estimation, a choice of smoothing should be applied to the kernel function estimates

Where

$$Z = \begin{pmatrix} N_p^{(\Delta)} & N_{p+1}^{(\Delta)} & \dots & N_{n-1}^{(\Delta)} \\ N_{p-1}^{(\Delta)} & N_p^{(\Delta)} & \dots & N_{n-2}^{(\Delta)} \\ \vdots & \vdots & \dots & \vdots \\ N_1^{(\Delta)} & N_2^{(\Delta)} & \dots & N_{n-p}^{(\Delta)} \\ 1 & 1 & \dots & 1 \end{pmatrix}, \quad Y = \begin{pmatrix} N_{p+1}^{(\Delta)} & N_{p+2}^{(\Delta)} & \dots & N_n^{(\Delta)} \end{pmatrix}$$

The effect of the value of p and Δ are discussed in Kirchner (2017). This method suggests for accurate estimation that the largest count in the discretised process is equal to 1. In the continuous time setting this can be controlled through the value of Δ. This choice is a trade-off between bias and variance with a smaller Δ favouring bias. This could be viewed as a bias/computational time trade off as the estimation process relies on a matrix inversion. However, in the case of discrete data due to imprecise observation, this choice is removed from the user. This means this method is suitable for this application, however it is understood that estimates will have a large variance should the observations be infrequent. It should also be noted that a semi-parametric method was suggested. However, this was shown in Kirchner and Bercher (2018) to have reduced performance even when the parametric model was equal to the true underlying intensity kernel. The performance of this method was compared to MLE in Kirchner and Bercher (2018), where it was shown that MLE outperformed this method. However, this was deemed to be due to the non-parametric element of the method. The main advantage of this method is the linear computational time compared to the naive MLE with is quadratic.

Binned Maximum Likelihood

This method was developed in Shlomovich et al. (2020) and can be considered the naive application of maximum likelihood estimation given discrete data. The conditional intensity function was treated as piecewise constant and is denoted henceforth as $\lambda^{(\Delta)}(i) \equiv \lambda^{(\Delta)}(i\Delta|\mathcal{H}_{i-1}^{(\Delta)})$. This led to the following log likelihood approximation:

$$\ell = \sum_{i=1}^{n} N_i^{(\Delta)} \log[\Delta\lambda^{(\Delta)}(i)] - \Delta\lambda^{(\Delta)}(i)$$

The assumption of a piecewise constant conditional intensity function was equivalent to assuming $N_i^{(\Delta)} \sim \text{Poisson}(\Delta\lambda^{(\Delta)})$. Therefore, this method has the same limitation as the INAR(p) method as events within a bin cannot trigger events in the same bin. To estimate the parameters, the log likelihood was maximised using constrained optimisation.

Monte Carlo - Expectation Maximisation

The main method derived in Shlomovich et al. (2020) was a Monte Carlo-Expectation Maximisation (MC-EM) algorithm for Hawkes processes. An MC-EM algorithm is used when the expectation of the posterior required during expectation step is analytically intractable. This can be solved using Monte Carlo integration. However, Monte Carlo integration requires a sample from the underlying posterior which is not available in this case. The proposed method in Shlomovich et al. (2020) uses importance sampling to draw a "legal" set of time points, $\{\tilde{t}_j\}_{j=1,\dots,k}$, given the observed process $\{N_i^{(\Delta)}\}_{i=1,\dots,n}$ and where $k = \sum_{i=1}^{n} N_i^{(\Delta)}$. This novel method of generating samples was conducted sequentially i.e. time points are drawn for each bin in order, and the time points sampled up to a given bin influence the sample of time points within a given bin. This was

achieved as follows: Given a sample up to event t_j which occur in the first $i-1$ bins and suppose there are m events in the ith bin. The joint truncated PDF of the m events given the history $t_1, \ldots, t_j$ is:

$$\frac{\prod_{s=1}^{m} f(t_{j+s}|\mathcal{H}_{t_{j+s}})}{F(i\Delta|\mathcal{H}_{t_{j+m-1}}) \prod_{s=1}^{m-1} F(t_{j+s}|\mathcal{H}_{t_{j+s}}) - F((i-1)\Delta|\mathcal{H}_{t_j}) \prod_{s=2}^{m} F(t_{j+s}|\mathcal{H}_{t_{j+s}})} \tag{3.2}$$

Where $f(\cdot|\cdot)$ denotes the conditional PDF and $F(\cdot|\cdot)$ denotes the conditional CDF as outlined in Shlomovich et al. (2020). By sequentially sampling each bin by maximising Eq. (3.2), a sample which maximises the likelihood can be generated. These samples are used to approximate the Expectation step using importance sampling and then proceed to the Maximisation step as normal. This method was shown to outperform the INAR method and the binned likelihood method on a small sample of processes with an exponential kernel. In particular, an analysis relative to discretisation size was conducted, showing that the MC-EM algorithm had the lowest bias for all step sizes from 0.2 to 2.

3.3. Variational Auto-Encoders and Hawkes Processes

3.3.1. VAEs for Self Excited Discrete data

Zhao et al. (2020) outlined a VAE which used a negative binomial likelihood (NBVAE) to model sparse and over-dispersed text data. This text data was in a form known as "bag of words" meaning the data was a sequence of counts of occurrences of words in a given document. Each word which occurred in the training data was assigned a location in a vector. An observation was then represented by a vector which contains integer values, with the integer in the ith position indicating the number of occurrences of the word in the ith position. The data encoded by this VAE was of similar form to the incremental count data produced by discretising a Hawkes process. The NBVAE claimed to encode both self-exciting and cross exciting properties of words within a document. Self-exciting in the context seen in Zhao et al. (2020) refers to the occurrence of a word resulting in more occurrences of the same word. The cross exciting property refers to the excitation of other related words given the occurrence of a word. The property which is referred to as self-exciting in Zhao et al. (2020) corresponds to the self excitation of a Hawkes process occurring within a the same bin and the cross-exciting property corresponds to self excitation of a Hawkes process occurring in a subsequent bin. Therefore, NBVAE has the potential to encode the self-exciting property of the Hawkes process.

3.3.2. Bayesian Inference in VAE Frameworks

A Bayesian inference framework was outlined in Mishra et al. (2020) which suggests, given a new data-point, y, and a fully trained VAE on data $\mathcal{D}$, a sample from the posterior distribution, $z|y$, can be created using MCMC sampling. By sampling from this posterior distribution, the decoder can subsequently be used to convert this sample to a sample from the predictive posterior $\tilde{y}|y$. It is possible that this method could be

used to generate new samples of aggregated Hawkes processes if data quantity was a problem. Alternatively, this Bayesian inference could be applied under a "Dueling" decoder framework as developed in Seybold et al. (2019). This framework relied on two decoders with differing target outputs using the same latent variable. This allowed for information to be encoded into the latent space which would not have been recovered by a single decoder. This required separate specifications of reconstruction loss for the separate decoders. The methodology in Seybold et al. (2019) suggested that the secondary decoder must have a distinct aim from the primary one as otherwise the methodology was simply equivalent to re-weighting the reconstruction loss in a standard VAE. This method may allow for improved Bayesian inference by using a secondary decoder which aims to reconstruct the $\theta = \{\alpha, \beta, \mu\}$. The Bayesian inference may be improved by a secondary decoder by increasing the quality or quantity of information encoded into the latent space, as well as allow for Bayesian inference around the predictive posterior of θ. This will be outlined in the next chapter.

An alternative Bayesian inference methodology using a complex VAE structure was proposed in Gabbard et al. (2019). The approach aimed to define the distribution of parameters underlying gravitational wave time series, θ, using two encoders and a single joint decoder. Due to the nature of the data (time series) the encoders and decoder were convolutional neural networks. The Bayesian inference on θ is achieved using one encoder which was conditional on θ and another encoder which was not. The conditional encoder was used to shape the latent space and an altered K-L divergence term was used to map the output of the non conditional encoder to the output of the conditional encoder. The decoder was then fed the latent variable along with the original input. This is far from the standard application of a VAE as the original input was fed to the decoder, however the decoder was not recreating the input, but instead defining the moments of the distributions of θ. As this method was developed for time series, it is possible that this method could be adapted for application to Hawkes processes. This would allow the distribution of $\theta = \{\alpha, \beta, \mu\}$ to be specified.

Super-Resolution

Another application of VAEs to aggregated Hawkes processes is one of super-resolution or interpolation. A VAE framework such as π-VAE seen in Mishra et al. (2020), was used to impute the temperature values of unobserved locations in Africa. The strong performance of this VAE framework in this imputation task would suggest it has potential application to aggregated Hawkes processes. This would be accomplished by sampling possible event times at a more frequent level of observation than the original data. This would allow the application of more traditional parameter estimation methods such as maximum likelihood estimation.

4. Methodology

The first section in this chapter outlines the form of the data which will be encountered, as well as the data generation process used for both the training and testing data. This data generation process formed an important step in the methodology outlined in this chapter. Following this, Variational Auto-Encoders and the generative model they provide, are applied to the problem of process reconstruction and parameter estimation. Subsequently, parameter estimation is approached using supervised learning techniques.

4.1. Data

The data is in the form of an aggregated Hawkes process, $x = \{N_j^{(\Delta)}\}_{j=1,\dots,T/\Delta}$, with horizon, T, and observation intervals of length Δ. As noted in Section 2.1.1, $N_j^{(\Delta)} = N(\Delta j) - N(\Delta(j-1))$. This value represents the number of events which occurred over the interval $(\Delta(j-1), \Delta j]$. Data of this form is generated due to censoring of event times of the underlying Hawkes process. This censoring occurs due to low frequency observation of the count process, $\{N(t)\}_{t>0}$, at regular intervals of length Δ, or due to rounding of event times.

4.1.1. Data Generation

To generate data for training, a realisation of a Hawkes process is simulated and this is then converted to the form of $x = \{N_j^{(\Delta)}\}_{j=1,\dots,T/\Delta}$. To generate a realisation of a Hawkes process with the kernel outlined in Eq. (2.5), the self-exciting rate, α, the intensity decay rate, β, the baseline intensity, μ, and the time horizon, T, must be specified. Alternatively, it is equivalent to specify either α or β and the branching ratio, η and then use $\eta = \frac{\alpha}{\beta}$ to solve for the unspecified hyper-parameter. It was decided that specifying β and η would be used in this thesis as it allowed greater control of the self-exciting properties. Therefore, generating the training data for the models which are outlined later in this chapter, requires the generation of the hyper-parameter sets $\{\beta, \eta, \mu\}$.

The Hawkes process realisations used in this thesis were generated using the "hawkes" package in R (Zaatour, 2014) which employs the methodology developed in Ogata (1981). To convert a Hawkes process realisation to the form of $x = \{N_j^{(\Delta)}\}_{j=1,\dots,T/\Delta}$ required the further specification of the discretisation step size, Δ. Δ represented the length of the observation interval. This data aggregation was achieved using a custom function in R which counted the number of events which occurred over each interval.

The expected activity of a Hawkes process, which shall be denoted with E, was presented in Eq. (2.6) and this was a dominating feature in early results. It was hypothesised that expected activity could easily be estimated from observed data of interest and therefore the decision was made to remove it as a variable from any subsequent model. To do so required the data generation process to generate processes with an approximately equal level of expected activity. A method for generating the hyper-parameter sets of n training processes is outlined below.

The following steps were taken to generate n training processes, $\{x_i\}_{i=1,\dots,n}$, with an expected activity level of E:

1. Sample a random normal $\epsilon_i \sim \mathcal{N}(E, \sigma^2) \quad \forall\, i = 1, \dots, n$
2. Sample $\eta_i \sim \mathcal{U}(a, b)$ where $0 < a < b < 1$ are set according to prior beliefs
3. Calculate $\mu_i = \frac{\epsilon_i}{T}(1 - \eta_i)$
4. Sample $\beta_i \sim \mathcal{U}(p, q)$ where $0 < p < q$ are set according to prior beliefs
5. Calculate $\alpha_i = \beta_i \eta_i$
6. Generate the ith training processes using an appropriate method and corresponding parameter set: $\{\alpha_i, \beta_i, \mu_i\}$
7. Discretise the event times at intervals of length Δ to form the respective incremental count processes $x_i = \{N_j^{(\Delta)}\}_{j=1,\dots,T/\Delta}$

Prior Specification

The prior specification for β and η was chosen to be the uniform distribution as this only infers the minimum and maximum values to be generated. These values vary during the thesis and are outlined in each section accordingly.

In a real world application of the methods seen in this thesis, the true values of β and η would be unknown. Therefore, the prior specification for both parameters would ideally contain the true values in their range. While this area requires further research, a possible method for prior specification for β in a real world application could be conducted in the following manner;

First, identify the shortest time-frame, $t_{\min}$, at which it is believed that the proportion of an events influence, i, will be diminished below some value near zero, δ. Using $\exp(-\beta t_{\min}) \leq \delta$, solve for the value of β. This value of β would represent the maximum value for the uniform distribution, q.

Similarly, identify the longest time-frame, $t_{\max}$, at which it is believed that the proportion of an events influence, i, will be diminished below some value near zero, δ. Using $\exp(-\beta t_{\max}) \leq \delta$, solve for the value of β. This value of β would represent the minimum value for the uniform distribution, p.

For example, consider the real event being earthquakes and their aftershocks. Let's define an event's influence to be fully diminished once it has dropped below 0.01 of

its initial influence. Let's say it is believed that the influence of an earthquake on the process intensity drops below 0.01 after at least 1 day or at most 7 days after the initial event. Using the above method, and a time unit length of one day:

$$\exp(-p(7)) = 0.01 \rightarrow p = -\log(0.01)/7 \approx 0.65$$
$$\exp(-q(1)) = 0.01 \rightarrow q = -\log(0.01)/1 \approx 4.6$$

A more simple process would be conducted for the minimum and maximum values of η. The use of domain knowledge to set limiting values of the number of expected first generation descendants should be straight forward. As will be shown in later sections, the range of η can be quite large, for example: $[0.05, 0.8]$, without negatively affecting performance.

It should also be noted that the prior specification of β and η is not inherently limited to a uniform distribution, however the effect of other prior distributions requires further study.

Test Data

As the test data used throughout this thesis was simulated, it was decided that the test processes would be generated using the same methodology as the training processes. For example, if the simulated test data was generated using the hyper-parameters outlined in Table 4.1, then the training data would be generated using the same values.

Parameter	Value
Intensity Decay Parameter (β)	$\mathcal{U}(p = 1, q = 3)$
Branching Ratio (η)	$\mathcal{U}(a = 0.05, b = 0.8)$
Expected Activity (E)	500
Time Horizon (T)	100
Interval Length (Δ)	1

Table 4.1.: Example Test Data Parameters

However, in real world applications of these methods, the underlying hyper-parameters which generated the aggregated Hawkes processes which are of interest would be unknown. Therefore, for a real world application of the methods seen in this thesis, there are additional steps that should be conducted before generating the training data:

Given m aggregated Hawkes processes of interest, $\{y_i\}_{i=1,\dots,m}$, with time horizon T and a discretisation step size of Δ, an estimate of the expected activity of these processes must be calculated. Assuming the m processes are generated by the same parameters, this would be done by calculating the mean of the total number of events seen for each process y_i:

$$\bar{y} = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{T/\Delta} \left(N_j^{(\Delta)} \right)_i$$

Using $E = \bar{y}$, proceed using the steps 1-7 outlined above to generate the training data.

Training Data Generation

The default number of training processes was 100,000. These were generated using the method outlined in Section 4.1.1 with the following parameters:

Parameter	Value
Intensity Decay Parameter (β)	$\mathcal{U}(p = 1, q = 3)$
Branching Ratio (η)	$\mathcal{U}(a = 0.05, b = 0.8)$
Expected Activity (E)	500
Time Horizon (T)	100

Table 4.2.: Default Parameter Values

4.2. Bayesian Inference and Variational Auto-Encoders

As mentioned in Section 2.2.2, the main advantage of a VAE is that they provide a generative model through the trained encoder and decoder. Therefore, with a properly trained VAE, the possibility of deriving a joint distribution of $\theta = \{\eta, \mu\}$, was investigated using the Bayesian inference frameworks outlined in Section 3.3.2. This application would be useful in understanding the dynamics of a Hawkes process, for example in financial markets it is desirable to understand the level of market endogeneity (Filimonov and Sornette, 2012) & (Filimonov et al., 2014). This behaviour could be described by the posterior distribution of $\theta = \{\eta, \mu\}$, as the value of η is equal to the expected number of first generation descendants. Before investigating the possibilities of the Bayesian Inference on Hawkes Processes, the reconstruction performance of various VAEs were tested on the data. Their ability to encode information into the latent space was investigated and using these results, the optimal VAE structure was chosen for the Bayesian inference methodologies.

4.2.1. Training

As with ordinary applications of MLPs, an excess of neurons allows the network to overfit to the data. This is because the network has enough capacity to memorise individual results rather than learn generalised properties. As the training data is generated for purpose, there was infinite training data available. This reduced the concern of overfitting due to the large training dataset. As a starting point, a two layer encoder and decoder were used. Two hidden layers allowed for the encoder and decoder to replicate more complicated functions and ensures the number of layers would not hinder the learning of the VAE. The number of neurons in the hidden layers of the encoder was dependent on the dimension of the input and dimension of the latent space. This ensured a steady compression of the input to the lower dimensional latent space and steady expansion of the latent variable to the reconstructed output. In some latter sections, the decoder was not constructed to be the mirror of the encoder and the chosen structure shall be explicitly stated for each decoder.

Note: As a MVN latent variable is suitable for most applications of VAEs, there is no reason to believe they shall not be suitable for applications to Hawkes processes. Therefore, as mentioned previously, the latent variable shall be assumed to be MVN henceforth. Only the dimension of the latent space shall be specified in the individual methods.

Bernoulli VAE

The aim of this VAE was to reconstruct the input, $x_i = \{N_j^{(\Delta)}\}_{j=1,\dots,T/\Delta}$. With a sufficiently fine level of discretisation, $N_j^{(\Delta)}$ would be binary in form. This representation has the lowest loss in information due to discretisation and therefore represents a test of performance when the least information is lost. The aim of this VAE is to test if reconstruction is possible when given maximum information. As mentioned previously for binary data, a Bernoulli likelihood is suggested in Kingma and Welling (2013). Given data $\{x_i\}_{i=1,\dots,n}$, the log likelihood is as follows:

$$\log(p(x_i|z_i)) = \sum_{i=1}^{n} x_i \log(p_i) + (1 - x_i) \log(1 - p_i)$$

where $p_i = \text{Decoder}(z_i)$ and $z_i = \text{Encoder}(x_i)$, i.e. p_i represents the output after point x_i is passed through the full VAE.

This represents the probability of a successful Bernoulli trial. An advantage to this method is the form of a VAE for binary data is well researched due to its application to image compression. This method was implemented with a simple training scheme of 5000 epochs with the following encoder and decoder structure:

	Encoder	Decoder
Layer 1	5000 act = ReLU	500 act = ReLU
Layer 2	500 act = ReLU	5000 act = ReLU

Table 4.3.: Structure of Encoder and Decoder for the Bernoulli VAE

Poisson VAE

The aim of this VAE was to reconstruct the input $x_i = \{N_j^{(\Delta)}\}_{j=1,\dots,T/\Delta}$. Due to the link between Hawkes processes and the Poisson process which is outlined in Section 2.1, the likelihood of a Poisson distribution is a natural choice for the likelihood of $N_j^{(\Delta)}$. Given data $\{x_i\}_{i=1,\dots,n}$, this leads to a log likelihood of:

$$\log(p(x_i|z_i)) = \sum_{i=1}^{n} x_i \log(\lambda_i) - \log(x_i!)$$

where $\lambda_i = \text{Decoder}(z_i)$.

λ_i can be interpreted as the expected activity over the period of bin i and as an inhomogeneous Poisson process this value is equal to the integral of the (conditional) intensity function over the interval $((i-1)\Delta, i\Delta]$. This interpretability is the main advantage of a Poisson likelihood.

The discretisation level was taken as $\Delta = 1$, which resulted in an input dimension of 100. The dimension of the latent space was tuned by examining the performance of the reconstruction and the latent space encoding. It was found that 15 latent dimensions were sufficient as there was a diminishing reduction in reconstruction loss at dimensions greater than 15. The structure used for the encoder and decoder can be seen in Table 4.4.

	Encoder	Decoder
Layer 1	75 act = ReLU	37 act = ReLU
Layer 2	37 act = ReLU	75 act = ReLU

Table 4.4.: Structure of Encoder and Decoder for the Poisson VAE

To ensure best performance, the VAE was trained for a total of 10,000 epochs. This included a cyclical annealing scheme (Fu et al., 2019) to ensure the K-L divergence was minimised after the reconstruction loss had been minimised. In particular, it was found that if the K-L divergence term was not annealed, the VAE found a local minimum in which the reconstructed output predicted the expected activity per time unit for all bins. The annealing schedule used for the training can be seen in Figure 4.1. The annealing weight was introduced after 2000 epochs after which it was increased over 500 epochs to 1/8. The weight was then held constant for 500 epochs before returning to 0. Over the next 500 epochs, the weight was annealed to 2/8. This was repeated until the weight reached 1, after which it was trained for a further 500 epochs. This allowed for the weight to be introduced more slowly at first, which ensured the shape of the latent space which minimised the reconstruction loss was not lost. The choice of activation function was made as a result of exploding gradients. However, this was not sufficient in isolation and so gradient clipping of values above 1000 was also implemented.

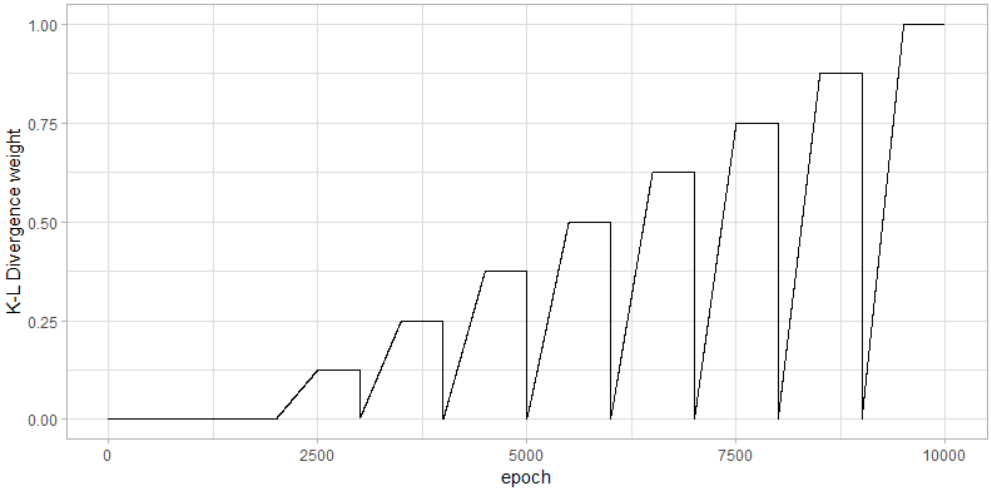


Figure 4.1.: Plot of Annealing Schedule

Negative Binomial VAE

The aim of this VAE was to reconstruct the input $x_i = \{N_j^{(\Delta)}\}_{j=1,\ldots,T/\Delta}$. Despite the interpretability of the Poisson likelihood, Zhao et al. (2020) demonstrated that a VAE with a negative binomial likelihood can capture both self-exciting and cross exciting properties in discrete data. This method was developed to be applied to text data, which has a similar form to x_i. The log likelihood is as follows:

$$\log(p(x_i|z_i)) = \sum_{i=1}^{n} \log \binom{x_i - 1}{r_i - 1} + r_i \log(p_i) + (x_i - r_i) \log(1 - p_i)$$

where $\exp(r_i) = \text{Decoder}_1(z_i)$, $\text{logit}(p_i) = \text{Decoder}_2(z_i)$, and $z_i = \text{Encoder}(x_i)$. As the negative binomial distribution requires the specification of two different parameters; the number of predefined successes, r, and the probability of success, p, the method derived in Zhao et al. (2020) outlines the use of two separate decoders. A discretisation level of $\Delta = 1$ and a 15 dimensional latent space was used. The following encoder and decoder structures were used:

	Encoder	Decoder 1 & 2
Layer 1	75 act = tanh	37 act = tanh
Layer 2	37 act = tanh	75 act = tanh

Table 4.5.: Structure of Encoder and Decoders for the Negative Binomial VAE

The scheme suggested in Zhao et al. (2020) used monotonic annealing to a final K-L divergence weight of 0.2. Due to the anticipated inference frameworks as outlined in

Section 3.3.2, the latent space required a more strict K-L divergence term. Therefore, it was decided to use the annealing scheme as outlined in Fig 4.1. Exploding gradients were not encountered as the tanh activation function has a restricted gradient.

Dueling Decoders

The aim of this VAE framework was to construct two decoders given a single encoder according to methodology derived from the work in Seybold et al. (2019). The encoder was designed to take input $x_i = \{N_j^{(\Delta)}\}_{j=1,\dots,T/\Delta}$. The aim of the primary decoder was to reconstruct $\theta_i = \{\eta_i, \mu_i\}$ given the input x_i. This was indicated to be possible due to preliminary results of the Poisson VAE, where the latent space indicated that θ_i could be encoded. As the aim of the primary decoder was purely the reconstruction of θ_i, the Mean Square Error (MSE) was used as the reconstruction loss. This was equivalent to assuming a Gaussian likelihood. Given the latent variable, the aim of the secondary decoder was to reconstruct x_i. Due to its performance, a Poisson likelihood was assumed and the structure matched that seen for Poisson VAE. By implementing this secondary decoder, it was hypothesised that the posterior of θ, given a data point, could be sampled from using MCMC. The primary decoder ensured that latent variable had information about θ explicitly encoded, which was not a guarantee under the Poisson VAE. The secondary decoder ensured that the latent variable represented the input in lower dimension as this was required for the inferential framework.

Due to the dual decoder framework, the reconstruction loss was comprised of two separate terms. The primary decoder contributed through the MSE and secondary decoder contributed through the log Poisson likelihood. These two terms were of different magnitudes; the Poisson likelihood was the sum of 100 independent likelihood values, while the MSE term was a mean error over a two dimensional vector. Therefore, the reconstruction loss of the Gaussian decoder required re-weighting. This re-weighting was not exact, but a factor 250 was chosen based on the size of the reconstruction error seen for the Poisson VAE. The Poisson likelihood was also weighted using a factor of $\frac{1}{1+\text{K-L weight}}$. This was done as a precaution, to ensure that the VAE would preserve the prediction of θ as the K-L divergence term was annealed. This was achieved by reducing the gradient of the secondary decoder reconstruction loss. Therefore, any increase in reconstruction loss caused by the annealing of the prior, would come from the secondary decoder.

The VAE was trained for 10,000 epochs and used the annealing scheme outlined in Figure 4.1. A discretisation of $\Delta = 1$ and latent dimension of 15 were used. The structure of the encoder and decoders were as follows:

	Encoder	Primary Decoder	Secondary Decoder
Layer 1	75 act = ReLU	15 act = ReLU	37 act = ReLU
Layer 2	37 act = ReLU	15 act = ReLU	75 act = ReLU

Table 4.6.: Structure of Encoder and Decoders for the Dueling Decoder VAE

As before for the Poisson VAE, gradient clipping at a value of 1000 was required due to exploding gradients.

Dual Encoders

This VAE structure was based on the methodology derived in Gabbard et al. (2019). As explained in Section 3.3.2, the VAE was comprised of two encoders and one decoder. This idea was explored for application to Hawkes processes as an alternative to the dueling decoders framework. It was deemed inappropriate due to the decoder receiving the initial input, $x_i = \{N_j^{(\Delta)}\}_{j=1,\dots,T/\Delta}$, as well as the latent variable. As shall be outlined in the following sections, neural networks can predict θ_i given x_i as the input. Therefore, the size of the contribution made by the latent variable is questionable in this application.

4.2.2. Bayesian Inference

Due to the positive results produced by the Poisson VAE and in the dueling decoder framework, the Bayesian inference scheme outlined in Mishra et al. (2020) was applied.

Given a VAE trained on data $\mathcal{D}$, and a new data point y, the inference scheme allows for sampling from the posterior distribution $p(z|y,\mathcal{D})$. This is achieved using the unnormalised posterior and MCMC sampling. The unnormalised posterior can be stated as:

$$p(z|y,\mathcal{D}) \propto p(y|z)p(z)$$

As outlined in Section 2.2.5, the form of both the likelihood, $p(y|z)$, and the prior, $p(z)$, is specified during the training of the VAE.

A sample from the posterior $p(z|y,\mathcal{D})$ can be converted to a sample from the predictive posterior distribution using a decoder. This converts the posterior sample to a sample from $p(\hat{y}|y,\mathcal{D})$. The unnormalised posterior in both the Poisson VAE and dueling decoder framework has the following form:

$$p(z|y,\mathcal{D}) \propto \prod_j {\lambda_j}^{y_j} \exp(-\lambda_j) \prod_i \exp(-0.5(z_i^2)) \tag{4.1}$$

where $\lambda_j = \Big(\text{Decoder}(z)\Big)_j$.

Using Eq. (4.1), MCMC sampling was completed using rStan (Stan Development Team, 2020).

Using the Poisson VAE, the decoder was applied to the resulting sample, leading to a sample of intensities of aggregated Hawkes processes. In the case of the dueling decoders, the primary decoder which reconstructed θ was used to create a sample from $p(\hat{\theta}|y, \mathcal{D})$. Additionally, the secondary decoder can be applied to the posterior sample to sample the corresponding intensities.

4.2.3. Testing VAE Performance

As was noted in Section 2.1, the incremental counts of a Hawkes process on the interval $(s, t]$ follows a Poisson random variable with intensity as given in Eq. (2.4). Therefore, the reconstruction performance of the Poisson VAE was tested by comparing the reconstructed Poisson intensities to the integral of the true conditional intensity over intervals of length Δ. These integrals were calculated according to Eq. (2.4). To compare the reconstructed intensity and the integrated true conditional intensity, the Normalised Root Mean Square Error (NRMSE) was used. This is as follows:

$$\text{NRMSE}(y) = \frac{\text{RMSE}(y)}{\max y - \min y}$$

The reason for selecting NRMSE over the regular RMSE was that the magnitude of the integrated intensity and thus, the magnitude of the error, rose with the magnitude of α. Therefore, normalising the error allows for the difference in magnitudes to be accounted for.

A comprehensive set of tests were conducted by calculating the NRMSE for 100 test processes using both the Poisson VAE and Dueling decoders. The test processes were generated with the parameters outlined in Table 4.7.

Parameter	Value
Intensity Decay Parameter (β)	$\mathcal{U}(p = 1, q = 3)$
Branching Ratio (η)	$\mathcal{U}(a = 0.05, b = 0.8)$
Expected Activity (E)	500
Time Horizon (T)	100
Interval Length (Δ)	1

Table 4.7.: Parameter values for VAE reconstruction tests

To further understand the performance of the methods, four processes were simulated with the following combinations of parameters:

Parameter	Test 1	Test 2	Test 3	Test 4
Intensity Decay Parameter (β)	1	3	1	3
Branching Ratio (η)	0.2	0.2	0.7	0.7
Interval Length (Δ)	1			
Expected Activity (E)	500			
Time Horizon (T)	100			

Table 4.8.: Parameters of VAE reconstruction tests

The NRMSE was calculated for the reconstruction produced by the Poisson VAE and the Dueling decoders. Bayesian inference for the parameters, $\theta = \{\eta, \mu\}$, was conducted and a density plot for $p(\theta|y, \mathcal{D})$ was calculated.

4.2.4. Super Resolution

As outlined in Section 3.3.2, this problem may have been approached using the π-VAE framework (Mishra et al., 2020). However, due to time constraints this was not pursued and it is believed to remain a viable approach for this problem.

4.3. Supervised Learning

Due to the promising initial results seen for VAEs, in particular the successful encoding of η and μ into the latent space, it was hypothesised that neural networks possessed the ability to predict η and μ as a regression task. This was investigated using an initial set of tests, which were followed by a broad selection of tests due to positive results. The methodology derived shall be outlined in the following sections, including an extension to allow for the estimation of α and β.

4.3.1. Estimation of Baseline Intensity and Branching Ratio

The method derived aims to accurately predict both the branching ratio, η and baseline intensity, μ as a regression task using neural networks. This was hypothesised to be possible with the incremental count data, $x_i = \{N_j^{(\Delta)}\}_{j=1,\dots,T/\Delta}$, as input due to the initial results with VAEs. Given a set of test processes $\{y_i\}_{i=1,\dots,m}$ with horizon T and discretisation step Δ, 100,000 training data processes were generated according to the method outlined in Section 4.1.1. The test data points $\{y_i\}_{i=1,\dots,m}$, are realisations of Hawkes processes with the same conditional intensity function parameters. Both m disjoint observations of length T/Δ and m segments of a single observation of length mT/Δ were used with the supervised learning method.

A feed forward MLP with input dimension T/Δ was used. The network contained six hidden layers with T neurons in each layer. The output layer consisted of two neurons, which represent $\theta = \{\eta, \mu\}$. The hidden layers used the ReLU activation function and L2 regularisation with weight 0.001. The final output layer used the linear activation function. Despite the output range being strictly non-negative, the addition of a ReLU

activation function caused issues with performance. This general network structure was designed with the aim of unsupervised deployment of a supervised learning algorithm. The aim of this network's design was to ensure that the learning was not limited by neuron capacity. However, in the event that the capacity was too large, the regularisation term would ensure that unnecessary neurons were down weighted. While this design may not be optimal for all applications it provides sufficient flexibility for most.

The network was trained for a maximum 500 epochs with early stopping defined to be less than a 0.01 reduction in validation loss over 25 epochs. This meant most applications trained fully within 200 epochs. A validation split of 0.05 was used to monitor performance.

4.3.2. Estimation of Self Excitation and Intensity Decay

Neural networks

Despite promising results for the prediction of η and μ using neural networks, initial results did not indicate the same performance for predicting α and β. The network structure and training scheme used were the same as seen in Section 4.3.1.

Linear model

Given that α and β could not be directly estimated using neural networks, the subsequent aim was to investigate alternative methods. As an estimate of η could be calculated using the method outlined in Section 4.3.1, it was only necessary to derive a method to estimate either α or β, due to the relation $\eta = \frac{\alpha}{\beta}$.

A hypothesis was formed that the maximum value of $x_i = \{N_j^{(\Delta)}\}_{j=1,\dots,T/\Delta}$ would be a descriptive statistic for the value of α_i. This hypothesis can be explained using two arguments, the first relies on the decrease in expected waiting time due to an event and the second relies on the contribution of an arrival to the conditional intensity. As mentioned in Section 2.1, the incremental counts of a Hawkes process follow a Poisson random variable defined by the integral of its conditional intensity. This link shall be utilised in both arguments.

For an inhomogeneous Poisson process, the inter-arrival times, $\{\tau_i\}_{i=1,\dots,m}$, are distributed according to an inhomogeneous exponential distribution with the same intensity function. The conditional intensity of a Hawkes process jumps by a value of α at the moment of an event. This jump in intensity leads to a corresponding jump in intensity in the distribution of arrival times. Using that $X \sim \text{Exp}(\lambda) \ \rightarrow \mathbb{E}(X) = 1/\lambda$, a jump in intensity of size α leads to a decrease in the expected waiting time which is proportional to the value α.

For example, suppose two processes with equal values of η and μ, have differing values of α and β. While both processes have the same level of expected activity, the form of the conditional intensity of the processes shall be very different. The larger the value of α, the larger the increase in intensity after an event, leading to a larger decrease in expected waiting time. However, this is momentary, as the expected activity of the

processes is equal. The process with the larger value of α shall have a corresponding larger value of β. This means the larger jump in intensity shall decay faster. This larger jump and shorter term decrease in expected waiting times was hypothesised to result in a large maximum value of $N_j^{(\Delta)}$ due to clustering of events within a bin.

The alternative argument relies on examining the contribution of the self-exciting function to the conditional intensity. Given a Hawkes process with the exponential kernel and branching ratio, η, suppose an event occurs at time t. The contribution to the conditional intensity of that event up to time $t + \delta$ can be calculated using the following:

$$\int_0^\delta \alpha \exp\left(-\frac{\alpha}{\eta}s\right) \mathrm{d}s = \eta\left[1 - \exp\left(-\left(\frac{\alpha}{\eta}\right)\delta\right)\right]$$

Looking at this equation, it should be noted that as stated in Section 2.1, as $\delta \to \infty$ then the RHS of the equation tends to η. However for a small δ, a larger value of α results in larger value of intensity due to the self-exciting function. This larger contribution to intensity was hypothesised to result in a large maximum value of $N_j^{(\Delta)}$ due to clustering of events within of bin.

Using this hypothesis, the following methodology was formed to identify the value of α and β using $\hat{\eta}$. This methodology was formed to be used in conjunction with the neural network prediction outlined in Section 4.3.1 but can easily be adapted to work with other η estimation methods. The aim of the method was to form training data which had an approximately linear relationship with the value of α and fit a linear model to it. Initial results using the maximum incremental count suggested there was too much variability in the observed maximums over short horizons. However, categorising processes by the value of α, and taking the mean maximum incremental count allowed for a predictive relationship. This was achieved as follows:

Given m discretised test processes, $\{y_i\}_{i=1,\dots,m}$, with time horizon T and a discretisation step size of Δ, the following steps were taken estimate α and β:

1. Using an estimate of η, $\hat{\eta}$, generate training processes, $\{x_i\}_{i=1,\dots,n}$, using the method outlined in Section 4.1.1 with $\eta_i \sim \mathcal{U}(\hat{\eta} - \delta_\eta, \hat{\eta} + \delta_\eta)$. Note: δ_η is chosen by the user, according to the confidence in $\hat{\eta}$.

2. For $k = 1, \dots, K$, Calculate:

$$s_k = \mathrm{mean}_i(\{\max_j(x_{ij}) \;\mid\; \alpha_i \in [(k-1)\delta_\alpha,\ k\delta_\alpha)\})$$

   where $K = \lceil \max_i(\alpha_i)/\delta_\alpha) \rceil$ and δ_α is chosen depending on the level of variability in the data.

3. Using s_k as the independent variable, a normal linear model is fit to the response $(k - 0.5)\delta_\alpha$

4. Calculate the median value of $\max_j(y_{ij})$

5. Using the fitted linear model, predict the corresponding value of $\hat{\alpha}$.

6. Calculate $\hat{\beta} = \frac{\hat{\alpha}}{\hat{\eta}}$

4.3.3. Testing and Comparison

Testing Effects of Parameters

To understand the performance of the method outlined in Section 4.3, a set of comprehensive tests were derived. The tests consisted of a set of 100 randomly sampled conditional intensity parameters. For each set of parameters, 200 realisations of an aggregated Hawkes process were generated. The parameters of the generated processes were estimated and the error on each set of parameter estimates was calculated. The error was taken to be the difference between the true values and estimated values. The distributions of the errors were then examined.

The parameters of interest were the parameters which directly influenced the creation of the training data. The testing data was assumed to come from the true underlying distributions which were used to generate the training data. In particular, the effect of the following parameters were investigated: the branching ratio η, the rate of decay β, the discretisation step size Δ, and the expected activity E. Each parameter was tested separately to ensure no interactions. The following default values were used:

Parameter	Default Value
Intensity Decay Parameter (β)	$\mathcal{U}(p = 1, q = 3)$
Branching Ratio (η)	$\mathcal{U}(a = 0.2, b = 0.6)$
Expected Activity (E)	500
Time Horizon (T)	100
Interval Length (Δ)	1

Table 4.9.: Default Parameter values

A range of values were selected to test each parameter. The values tested for each parameter were as follows:

Intensity Decay Parameter (β)	Branching Ratio (η)	Interval Length (Δ)	Expected Activity (E)
[0.5,2.5]	[0.1,0.4]	0.25	50
[1.75,3.75]	[0.3,0.6]	0.5	100
[3,5]	[0.5,0.8]	1	250
[0.5,3]	[0.1,0.6]	2	500
[1.5,4]	[0.2,0.7]	5	1000
[2.5,5]	[0.3,0.8]		
[0.5,4]	[0.05,0.6]		
[1.5,5]	[0.05,0.7]		
[0.5,5]	[0.05,0.8]		

Table 4.10.: List of Parameter Ranges/Values tested

4.3.4. Comparison of Parameter Estimation Methods

To further understand the distribution of the errors, the method derived in Section 4.3 was compared to the MC-EM algorithm derived in Shlomovich et al. (2020). This method was chosen for comparison as it was shown to outperform both the binned log likelihood and the INAR(p) methods. The errors for the maximum likelihood estimates were also calculated as a benchmark of performance in a continuous time setting.

A set of 100 test processes with a time horizon of $T = 1000$ were generated using randomly sampled conditional intensity function parameters. An estimate of the parameters for each process was calculated using each method. Each test process was split into 10 subsections of length $T = 100$ to allow for prediction using the supervised learning method. The test processes were generated in this way to ensure that each method used the same data. The supervised method was also tested in a "high data" environment by generating 200 realisations of length $T = 100$ for each of the testing parameter sets. The test processes were generated using the following values:

Parameter	Value
Intensity Decay Parameter (β)	$\mathcal{U}(p = 1, q = 3)$
Branching Ratio (η)	$\mathcal{U}(a = 0.5, b = 0.8)$
Expected Activity (E)	1000
Time Horizon (T)	1000
Interval Length (Δ)	1

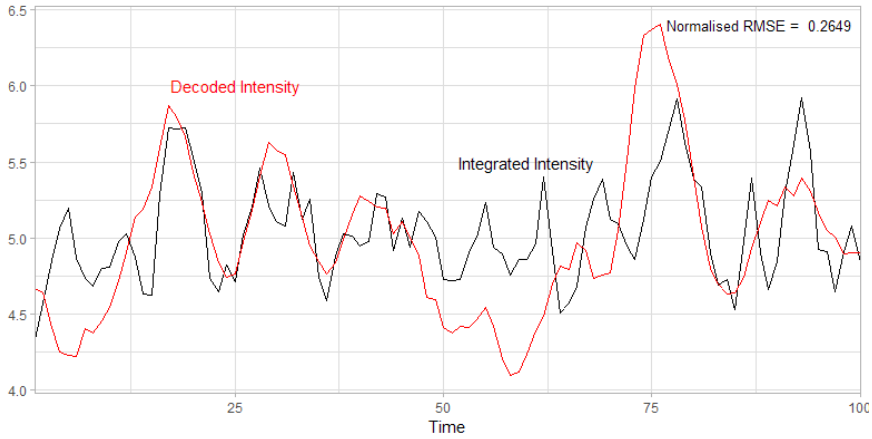
Table 4.11.: Default Parameter values

5. Results & Discussion

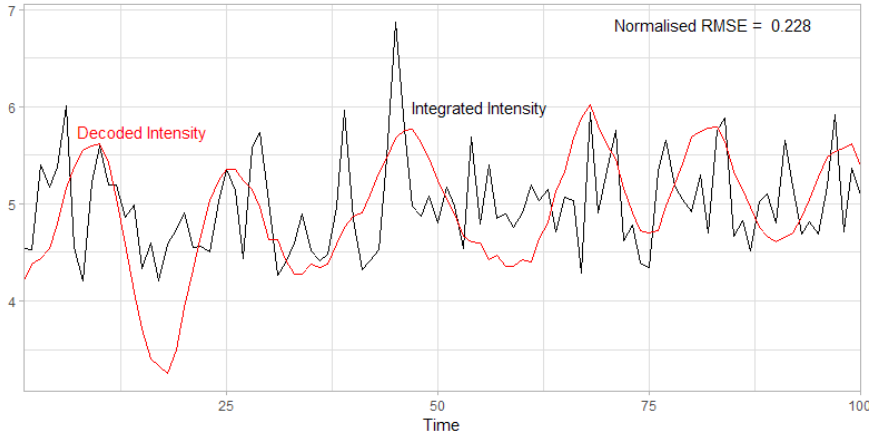
The results presented in this chapter are separated into two sections. The first contains an analysis of the performance of the VAEs which were outlined in the previous chapter. Particular attention is paid to the performance of the Poisson VAE and the Dueling decoder framework. This section also includes an example of the MCMC sampling for both VAEs. The second section shall contain the results of the comprehensive set of tests conducted on the supervised learning algorithm. This is concluded with a comparison of performance with the MC-EM algorithm and Maximum likelihood estimates.

5.1. Evaluation of Variational Auto-Encoders

The results for the Poisson VAE of the reconstruction of the four test processes outlined in Table 4.8 are presented first. The results to tests 1-2 and tests 3-4 are presented in Figures 5.1 and 5.2 respectively. As outlined in Section 4.2.3, these figures consist of a comparison of the reconstructed intensity and the integral of the true conditional intensity for each process. This is followed by the reconstruction of the four test processes using the Poisson decoder trained under the Dueling decoders framework. A density plot of the sample from the joint distribution of $\theta = \{\eta, \mu\}$ is also presented. The results to test 1,2,3, and 4 are presented in Figures 5.3, 5.4, 5.5, and 5.6 respectively. Finally, a comparison of the NRMSE achieved by the Poisson VAE and Dueling decoder on the 100 test processes outlined in Table 4.7 is presented in Figure 5.7.



(a) Test 1: $\{\alpha, \beta, \mu\} = \{0.2, 1.0, 4\}$

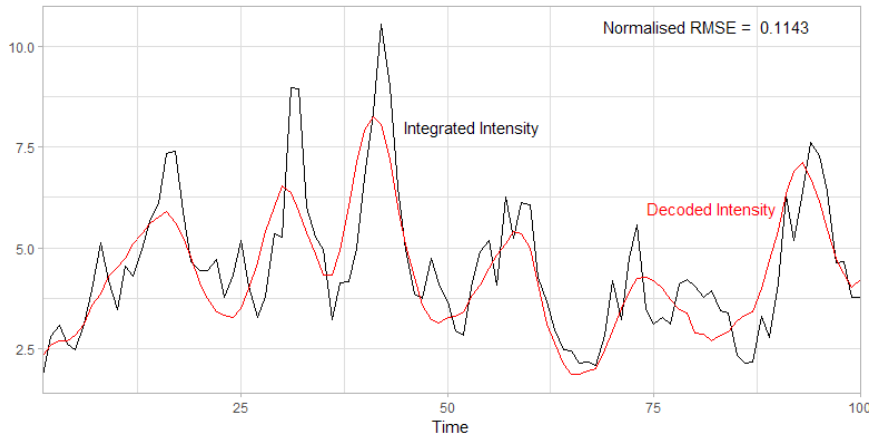


(b) Test 2: $\{\alpha, \beta, \mu\} = \{0.6, 3.0, 4\}$

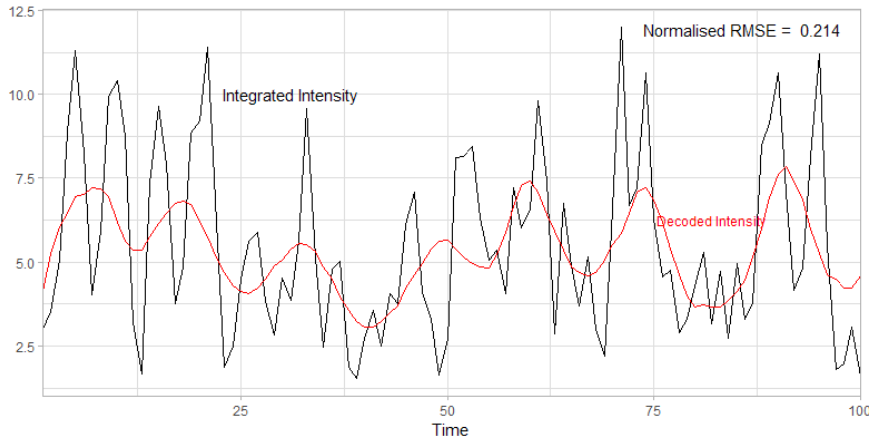
Figure 5.1.: Reconstruction of Integrated Intensity using Poisson VAE

The reconstruction of test 1 is shown in Figure 5.1a. This test scored the highest reconstruction error of 0.2649. Of interest is the overestimation of the spike in intensity seen at time 75. This is one of the few occasions where the VAE over-estimates the size of a significant spike in the reconstruction. This indicates that an unusual event occurred in this process, with a very large spike in the number of events at this bin. The performance on the section before 50 time units is better, with the reconstructed intensity capturing the rise and fall in intensity after time 10 very well. The reconstruction also manages to capture the final spike at approximately time 90.

The reconstruction of test 2 is shown in Figure 5.1b, the performance on this process is much improved compared to the last. Disregarding the dip in intensity before time 25, the reconstructed process manages to approximate the general rise and fall. The reconstructed process is much smoother when compared to the true integrated intensity. This appears to be a common occurrence. This is an understandable result of the compression to the latent space as precise information about the intensity is lost. A



(a) Test 3: $\{\alpha, \beta, \mu\} = \{0.7, 1.0, 1.5\}$



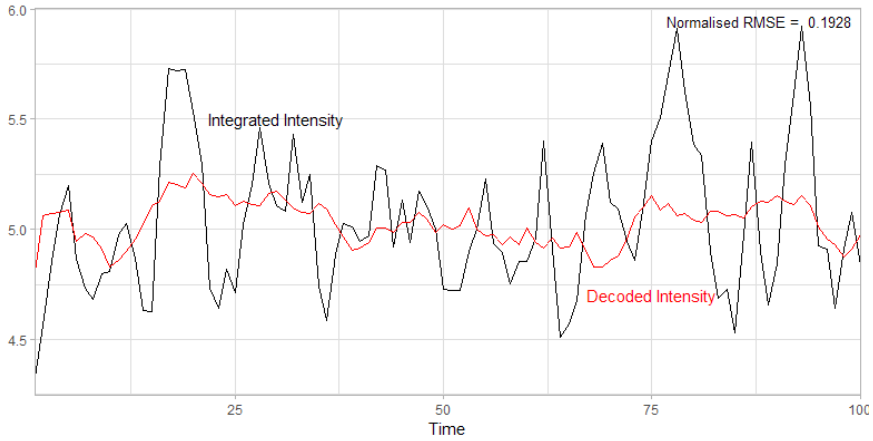
(b) Test 4: $\{\alpha, \beta, \mu\} = \{2.1, 3.0, 1.5\}$

Figure 5.2.: Reconstruction of Integrated Intensity using Poisson VAE

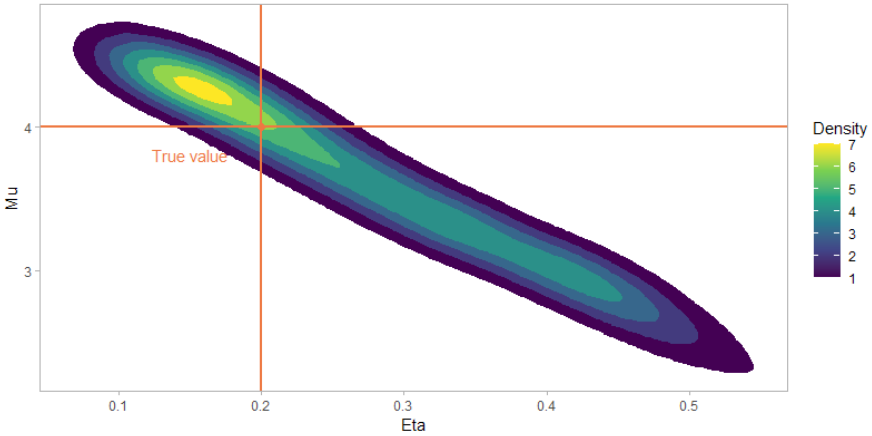
NRMSE of 0.2280 is relatively high. This is most likely caused by the dip before time 25 and the underestimated spike at time 45.

The reconstruction for test 3 is seen in Figure 5.2a, and scores the lowest reconstruction error of any of the processes demonstrated. With a NRMSE of 0.1143, the reconstruction captures each peak and trough very well, though the peaks between time 25 and 50 are slightly underestimated. This plot is a perfect example of when the VAE performs exceptionally and, as will be seen later, this performance should not be expected in the current implementation. This reconstruction performance does indicate the potential of the method.

Figure 5.2b shows the reconstruction of test 4 under the Poisson VAE. A similar smoothing property as was seen for the lower η process in Figure 5.1b is seen here. The peaks of the true intensity are much higher when compared to the reconstructed process. The NRMSE value of 0.214 is the third highest out of the tests presented and can be



(a) Reconstruction

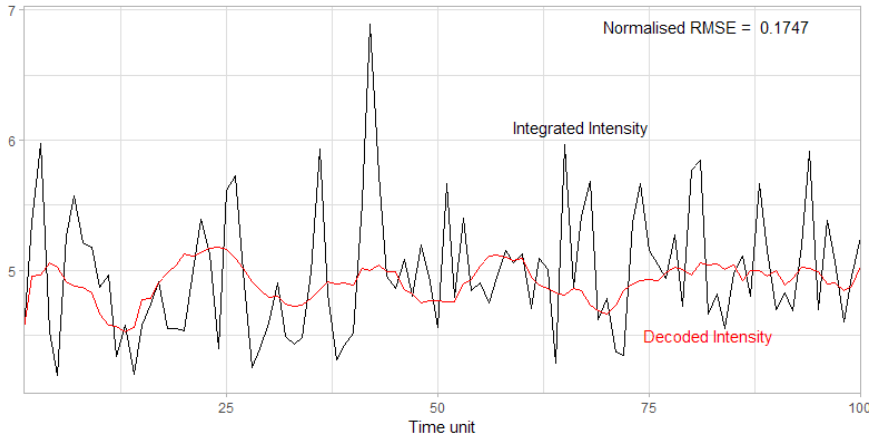


(b) Density plot of Predictive posterior for θ

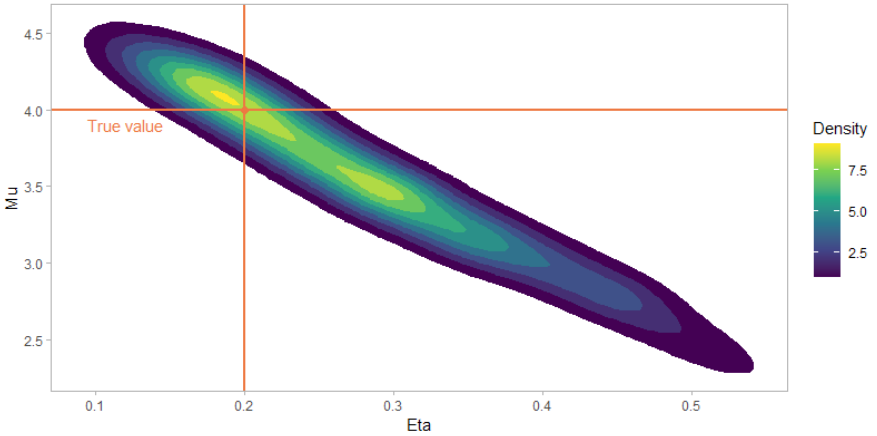
Figure 5.3.: Performance of Dueling Decoders on Test 1: $\{\alpha, \beta, \mu\} = \{0.2, 1.0, 4\}$

attributed to this smoothing. In particular the reconstructed process seems to merge peaks of high activity, for example the 4 peaks seen before time 25 are reconstructed using only two peaks. This reconstruction results in a high error even though periods of raised intensity were captured.

Figure 5.3a shows the reconstruction of test 1 in the Dueling decoder framework. Despite the lower NRMSE of 0.1928, the peaks of the integrated intensity are estimated to be much lower than in the Poisson VAE and seemingly were not captured. In particular, the peaks and troughs seen before time 50 are reconstructed poorly when compared to the Poisson VAE. However, the spike in reconstructed intensity at time 75 has been removed by the primary decoder and this led to an interesting result seen in Figure 5.3b. This figure shows the true value of η being located less than 0.05 away from the peak of the joint density. This strong reconstruction of θ suggests that the intensity of the process is quite abnormal for the parameters and the θ decoder influenced the placement



(a) Reconstruction



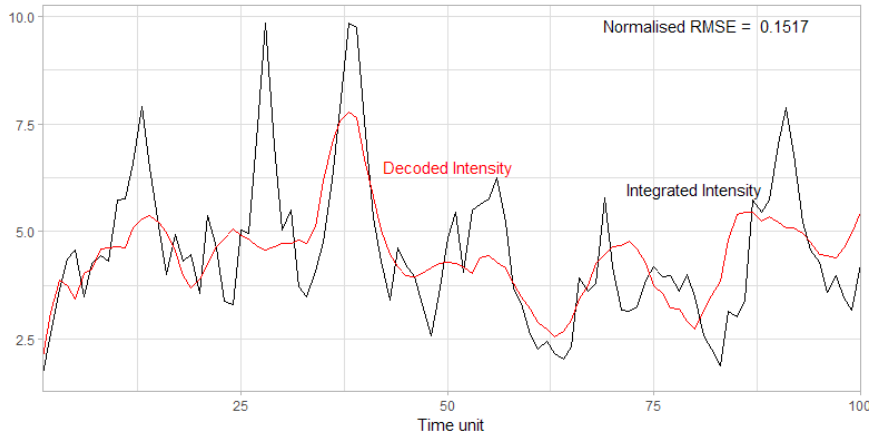
(b) Density plot of Predictive posterior for θ

Figure 5.4.: Performance of Dueling Decoders on Test 2: $\{\alpha, \beta, \mu\} = \{0.6, 3.0, 4\}$

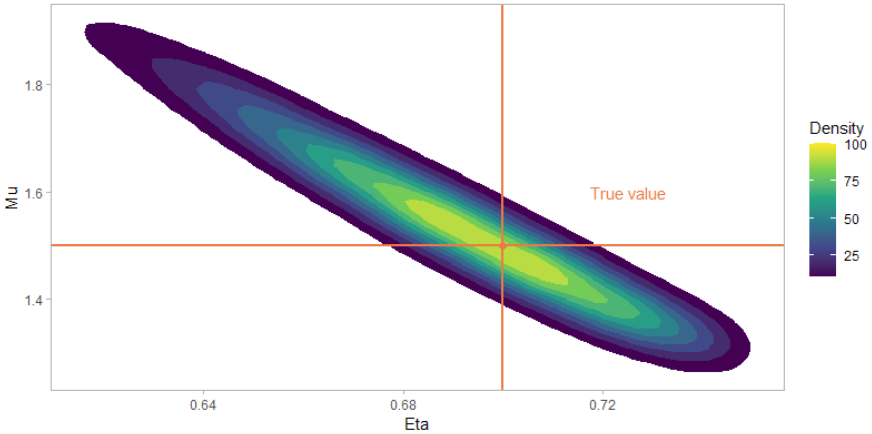
of this process in latent space. This would explain the loss of peaks in intensity in the reconstructed process when compared to Figure 5.1a, as high peaks in intensity should be a rare event with a low branching ratio.

Figure 5.4a shows the reconstruction of test 2 in the Dueling decoder framework. The NRMSE of 0.1747 is significantly lower than the value seen for the Poisson VAE. However on this occasion, as seen with the previous figure, the overall peaks and troughs of the reconstructed intensity do not match the integrated intensity as well as the Poisson VAE. Looking at Figure 5.4b, the estimation of η seems strong, with the true value of η lying almost exactly on the taller peak of the multi-modal distribution seen. Therefore, once again, a trade-off has been seen where the reconstruction of the process is influenced in a negative manner by the primary decoder. However, this influence being negative is in spite of the lower NRMSE.

The reconstruction of test 3 under the Dueling decoder is shown in Figure 5.5a. While

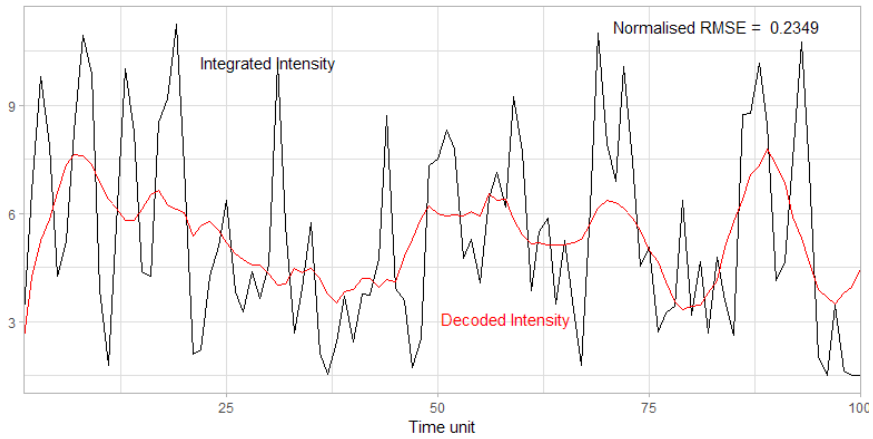


(a) Reconstruction

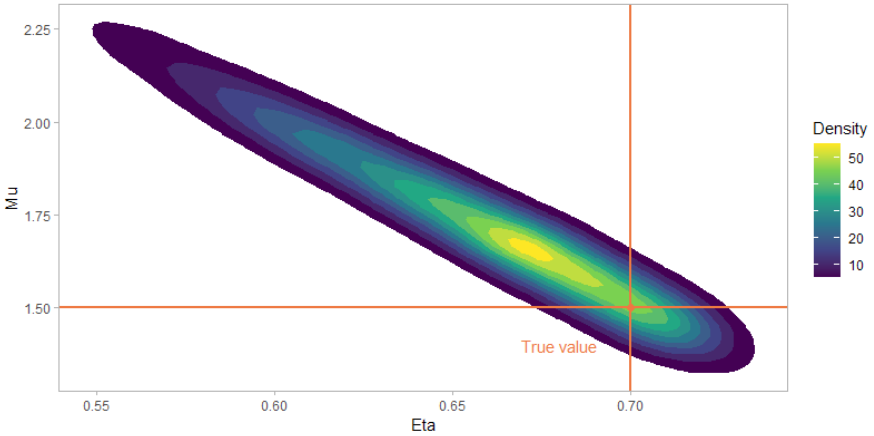


(b) Density plot of Predictive posterior for θ

Figure 5.5.: Performance of Dueling Decoders on Test 3: $\{\alpha, \beta, \mu\} = \{0.7, 1.0, 1.5\}$



(a) Reconstruction



(b) Density plot of Predictive posterior for θ: $\{\alpha, \beta, \mu\} = \{2.1, 3.0, 1.5\}$

Figure 5.6.: Performance of Dueling Decoders on Test 4

the NRMSE is higher when compared to the Poisson reconstruction, the reconstruction performance is still strong. The peaks and troughs are mostly captured with the exception of the peak immediately after time 25. The result seen in Figure 5.5b shows a very tight density around the true value of $\theta = \{0.7, 1.5\}$. The performance seen in this pair of plots shows the huge potential of this method, allowing the integrated intensity to be reconstructed well, and a strong estimate of the density of θ to be produced.

A slight increase in NRMSE is seen for the reconstruction of test 4 in Figure 5.6a when compared to the Poisson VAE. Though as before, this trade-off is made due to the influence of the primary decoder which performs very well. The density seen in Figure 5.6b has the peak located less than 0.05 away from the true value of η and 0.15 away from the true value of μ.

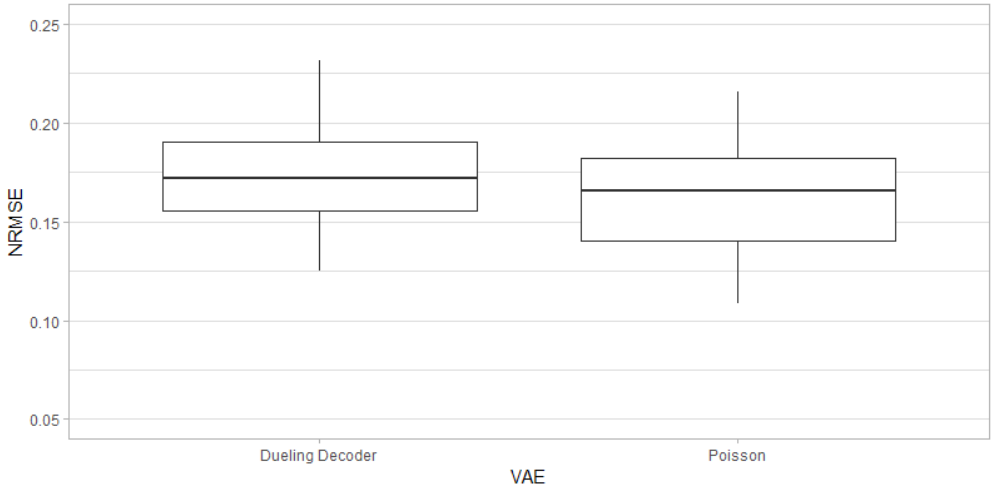


Figure 5.7.: Comparison of Reconstruction Performance

Figure 5.7 shows the NRMSE as a measure of reconstruction performance over 100 test processes. It can be seen that the two VAEs have similar distributions, with the Poisson VAE being shifted slightly lower when compared to the Dueling decoder. However, as seen previously the magnitude of NRMSE is not a complete metric and to fully understand the reconstruction performance would require further work.

Discussion of Overall Performance

The reconstruction performance of the Poisson decoder and density estimates produced in the Dueling decoder framework highlight the potential of the methodologies outlined. The Poisson VAE managed to reconstruct the integrated intensity well in most cases, managing to capture the peaks and troughs of the intensity. The Dueling decoder framework had a measurable effect on the reconstruction performance and this was as expected. Recall, the reconstruction loss of the intensity was reduced as the K-L divergence was annealed to ensure that the accuracy of parameter reconstruction was preserved. This was due to the particular interest in sampling from the density of θ. Given that reconstruction of the processes was of interest, it is possible that with slight alterations to the annealing scheme, the performance of Poisson decoder under the Dueling decoder framework could be improved. For example, if the weight of the θ reconstruction loss was reduced with the K-L divergence annealing, it is possible the resulting decoders may have outperformed the Poisson VAE in terms of intensity reconstruction. This is a suggested area of future research as strong reconstruction performance could hold great value in cases of low data quantity, as using the Bayesian inference outlined could allow for samples of aggregated Hawkes processes to be drawn while preserving the underlying unknown conditional intensity parameters.

It is worth noting that the training of the VAEs was challenging. As mentioned previously, the size of the latent dimension restricted the performance greatly. For

example, when only 2 latent dimensions were used, the resulting reconstructed intensities for any test process were approximately constant at the level of expected intensity. Despite this, the latent space seemed to encode θ but was unable to extract meaningful representations. The latent dimension of 15 was chosen to provide the flexibility to reconstruct the input. Replicating these results with a lower dimensional latent space is an area which requires further research.

Comment on Bernoulli and Negative Binomial VAE Performance

Results are not presented for the Bernoulli and Negative Binomial VAEs for the sake of brevity. The Bernoulli VAE failed to reconstruct the Hawkes processes due to their sparsity. This was due to the high level of discretisation needed to convert the Hawkes process to binary form. The step size required was as low as 10^{-5} in magnitude. It should be noted that if a process is discretised to a level in which its aggregated form is binary, the loss in performance from converting the binary vector to approximate event times and using traditional parameter estimation methods such as MLE would be minimal.

For the Negative Binomial VAE, the performance of the reconstruction was difficult to measure as, unlike the Poisson VAE, the Negative Binomial distribution does not share a link with the Hawkes process. Comparing the expected value of the Negative Binomial random variables with the integrated intensity did not indicate good performance, however this was a sub-par measure. Due to the claims made in Zhao et al. (2020), this VAE still holds potential for this application.

5.2. Evaluation of Supervised Learning Method

The results presented in Section 5.2.1 demonstrate the effect that the parameters which can be varied during training can have on the errors associated with the estimation of α, β, μ. The ranges/values of parameters tested are outlined in Table 4.10. The comparison outlined in Section 4.3.4 between the supervised learning method, MLE, and the MC-EM algorithm (Shlomovich et al., 2020) can be seen in Section 5.2.2. This is followed by a discussion about the performance of the supervised learning method in Section 5.2.3.

5.2.1. Test of Effect of Parameter Values/Ranges

The results which demonstrate the effect that the range of β has on estimation are presented in Figure 5.8. This is followed by Figure 5.9, which shows the effect that Δ has on estimation. The effect of the range of η has on estimation error is presented in Figure 5.10. This section is concluded with Figure 5.11 which shows the performance of the supervised learning method when the expected activity of the processes are varied.

Testing the Effect of the Range of Beta

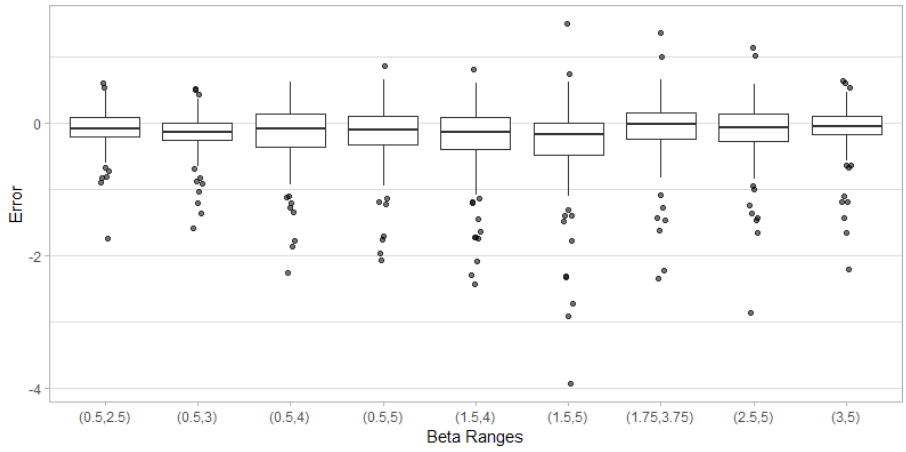
First, it should be noted that the range of values which β can take affects the range of values which α can take. Recall that the default values of η are $[.2, .6]$ as outlined in Table 4.9.

Looking at Figure 5.8, note the long negative tails in the distributions across all three plots. As the error is calculated as the predicted value minus the true value, this indicates the method underestimates all three parameters more often than it over estimates.

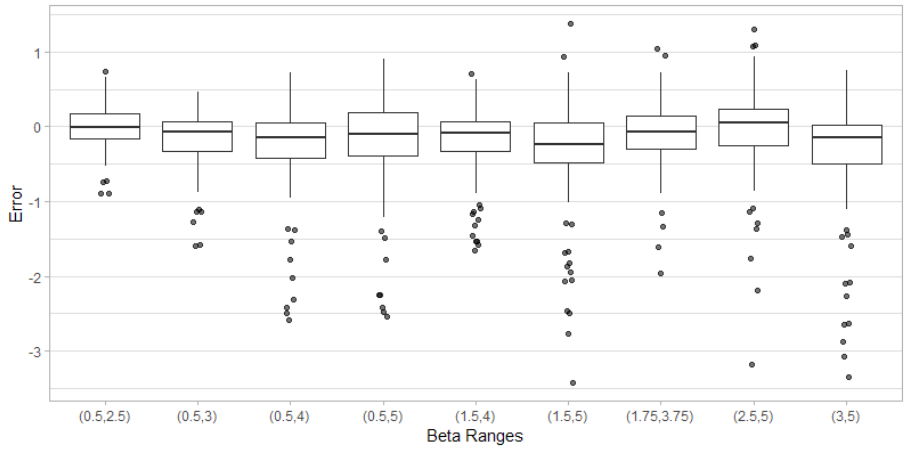
Looking at Figure 5.8a, the estimation error is generally consistent, with the inter-quartile ranges for all β ranges being approximately within ± 0.5. Note that the three box-plots with the thinnest tails are when the range of β is $(0.5, 2.5)$, $(0.5, 3)$, and $(3, 5)$. This would suggest the range of β has a slight effect on the estimation of α. Looking at the difference in magnitudes between β ranges (0.5,3) and (3,5), this may indicate that the magnitude of β does not affect the prediction of α. The longest tailed distribution appears when β ranges from (1.5,5). However, the long tails observed for this range of β values can be attributed to outliers. This can be concluded due to the performance when β ranged from 0.5 to 5.

Looking at Figure 5.8b, first note the change in scale on the y-axis. Also note that the performance of η prediction can be inferred by comparing Figure 5.8a and Figure 5.8b. It can be seen that the estimation of β appears to depend more significantly on the range of the values taken. In particular, note the shorter tail seen for (3,5) in the previous plot has disappeared. This suggests that the estimation of η is dependent on the magnitudes of β. This appears to be a consistent trend of lower valued ranges of β leading to lower errors and higher valued ranges leading to larger errors, as the three longest tailed distributions are those which have an upper value of 5.

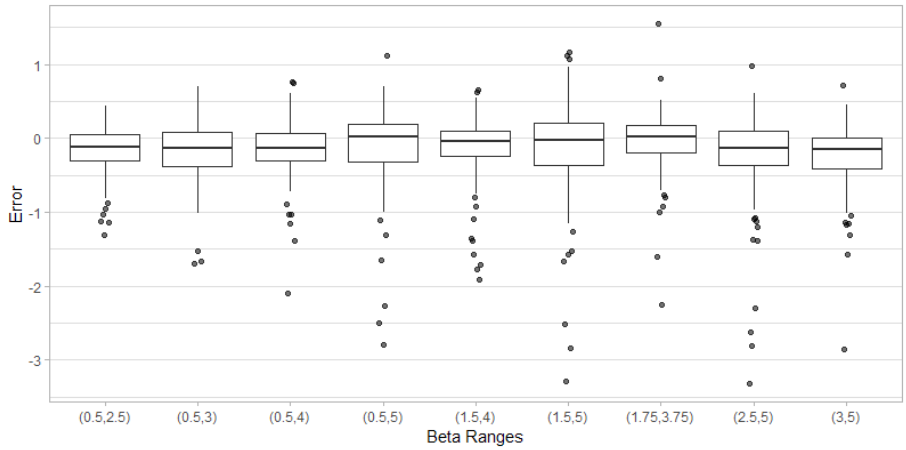
Finally, examining Figure 5.8c, the magnitude of β has a distinct effect on the estimation of μ. This is most obvious when comparing the first four ranges, which have longer



(a) Effect on α estimation



(b) Effect on β estimation



(c) Effect on μ estimation

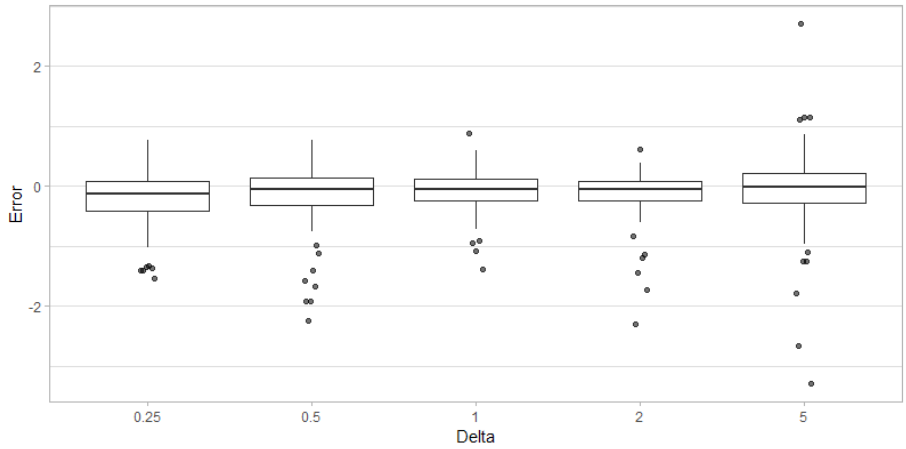
Figure 5.8.: Testing the Effect of the Range of β on the Supervised Learning method

tailed distributions as the maximum value increases. This confirms the conclusion drawn from the previous plot and allows for the conclusion that the estimation of β (through η) and μ are affected by the magnitude of β, while α is not.

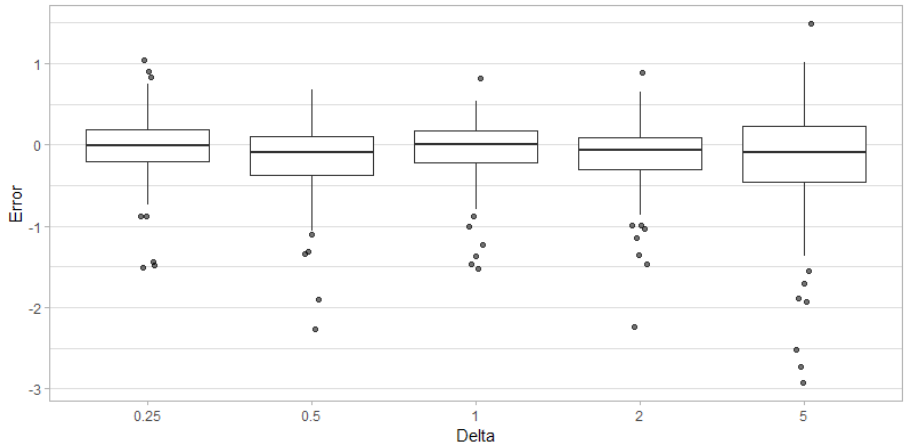
Testing the Effect of the Value of Delta

Figure 5.9 shows the distribution of errors for all three parameters at multiple values of Δ. While the tails of the distributions remain quite long, in particular the negative tails, the overall performance appears to be quite similar across all discretisation levels. For Figure 5.9a, the interquartile ranges are all within ± 0.5 with the whiskers of the plot being inside ± 1.0 for all discretisation levels. This performance is incredibly strong allowing for discretisations of 5 time units with only a marginal drop in performance. This result is surprising as the effectiveness of this estimation method should be directly linked to the level discretisation. This can best be seen by considering the change in $\max_i(N_i^{(\Delta)})$ as Δ tends to extremes. In particular, as $\Delta \to 0$, the maximum count of a process given an event occurred shall tend to 1 regardless of the value of α. Similarly, as $\Delta \to T$, the maximum count shall tend to the total number of events seen over the horizon. As $T \to \infty$, this statistic shall tend to the expected activity which is dependent on η and μ rather than α.

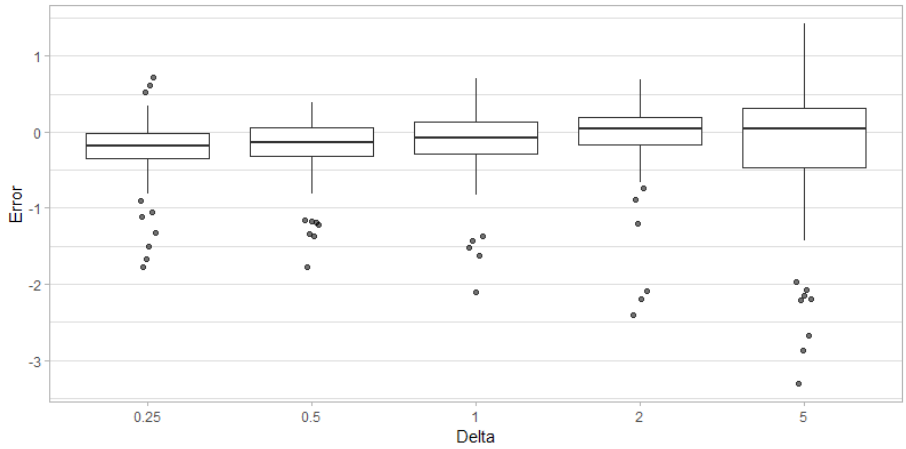
Looking at Figures 5.9b and 5.9c, the results are approximately similar with good performance at all levels. There is slightly more deviation when $\Delta = 5$. This suggests the neural network is slightly more affected by the level of discretisation, however this difference is slight, potentially allowing for tuning of the neural network structure to overcome this difference. Alternatively, the neural network may require a longer time horizon at larger levels of discretisation. In conclusion, prediction performance for all three parameters is broadly unaffected by the level of discretisation.



(a) Effect on α estimation



(b) Effect on β estimation



(c) Effect on μ estimation

Figure 5.9.: Testing the Effect of the Value of Δ on the Supervised Learning method

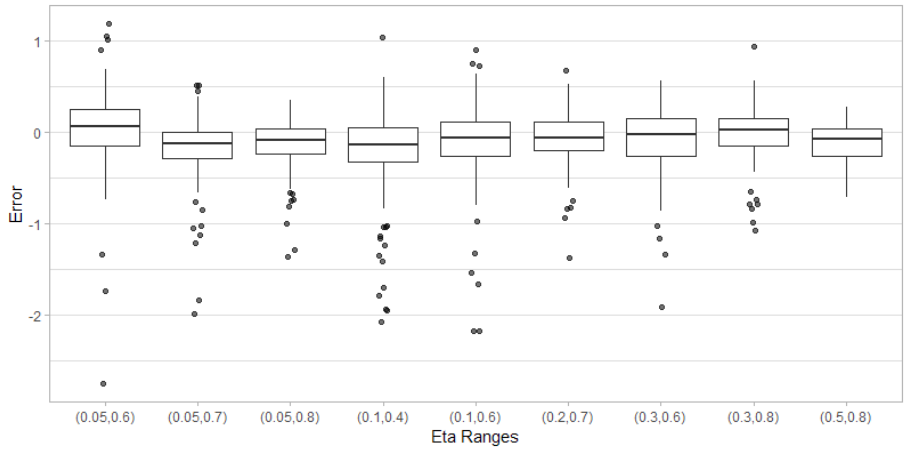
Testing the Effect of Range of Eta

The range of η has an effect on the ranges of α and μ due to the data generation process. Figure 5.10, shows the same distribution of errors seen in the previous two tests, where the long negative tails are a primary feature. However, these tails are more consistent across each set of ranges. Examining Figure 5.10a, note the tight inter-quartile ranges and short whiskers. The plot suggests that while the inter-quartile range and whiskers are not affected by the range of η, that the severity of outliers is lower for high values of η. This can be seen by careful examination of the first three and the last boxplot in Figure 5.10a. This relationship is intuitive as the higher values of η allow for larger ranges of α which would make the linear model less susceptible to outliers.

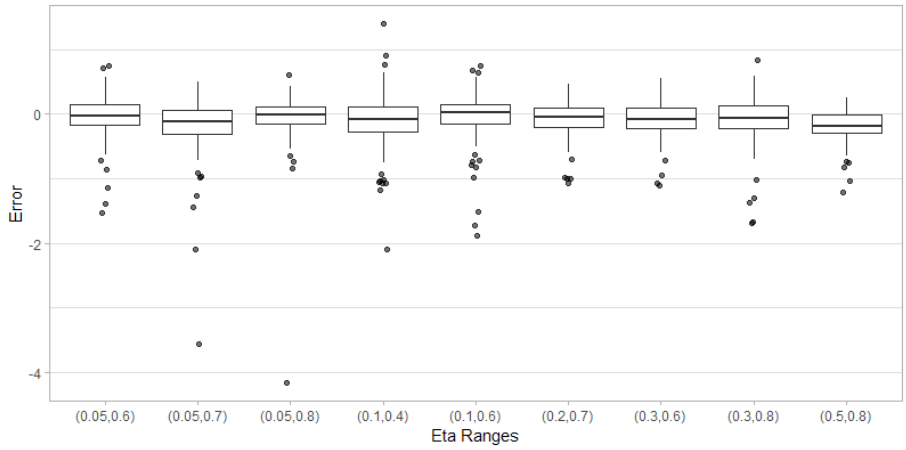
Looking Figure 5.10b, the distribution of errors appears to have a strong consistent trend with a few extreme outliers less than -2. The trend of increased outlier distribution and higher values of η appears to be more subdued, suggesting that this trend is present for α estimation but not present for η estimation. The extreme outlier of -4, seen for the η range of (0.05,0.8), appears to be due to the neural network prediction of η. This conclusion can be drawn as the outlier is not present for prediction of α but is present of the prediction of μ. Figure 5.10c appears to have a relation between the estimation of μ and the range of η. The trend appears to be similar to that seen for α with high ranges leading to lower errors. However, this may be due to higher ranges of η leading to lower ranges of μ and while the magnitudes of the errors change, the relative error may not change. In conclusion, the performance is widely stable but concern for outliers would be prudent for lower ranges of η.

Testing the Effect of the Value of Expected Activity

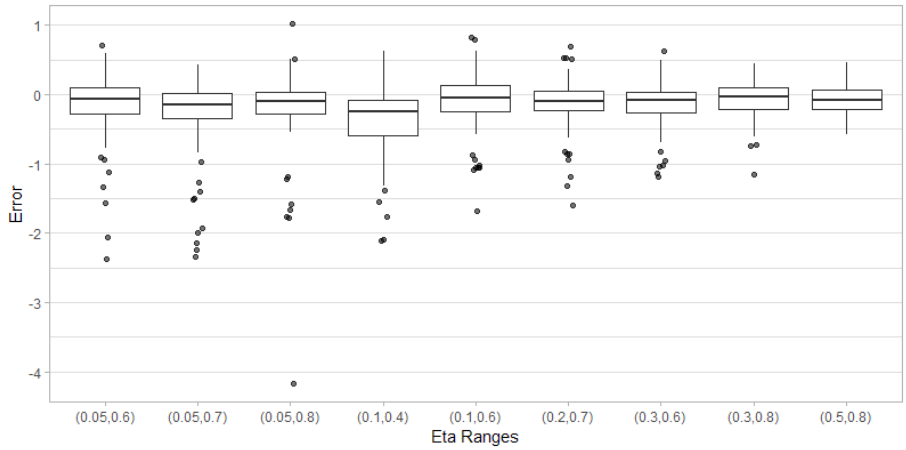
Examining the effect of expected activity gives a good insight into the estimation accuracy. Note, the previous tests were conducted with an expected activity of 500. Due to the performance of the estimation being consistent across all levels of discretisation, one may have expected similar results for expected activity. However, recall that the expected activity directly affects the values which μ can take. Due to the fixed η range, a higher expected activity corresponds to a higher range of μ. The first item to note when examining Figure 5.11, is the long negative tails which have been a feature of the previous 3 tests are balanced by positive outliers in the low expected activity boxplots. This means the consistent under-estimation of parameters is directly caused by the higher values of μ distorting the self-exciting property. This comes with the caveat that the total number of outliers appears to be lower for higher expected activity. This is particularly evident looking at Figure 5.11c, where the bounds on the outliers are tighter for the higher values of expected activity. The reason this is of interest, is the values of μ are larger at these values of expected activity meaning that the relative error is significantly lower on μ estimation for these values. This leads to the conclusion that the overall performance is relatively unaffected by the expected activity however there are concerns with the tails of the error distribution which appear to be thicker for smaller values of expected activity.



(a) Effect on α estimation

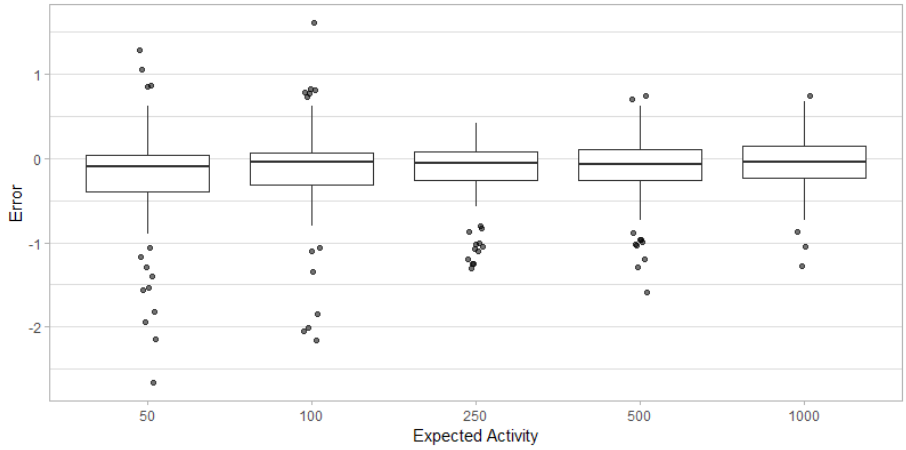


(b) Effect on β estimation

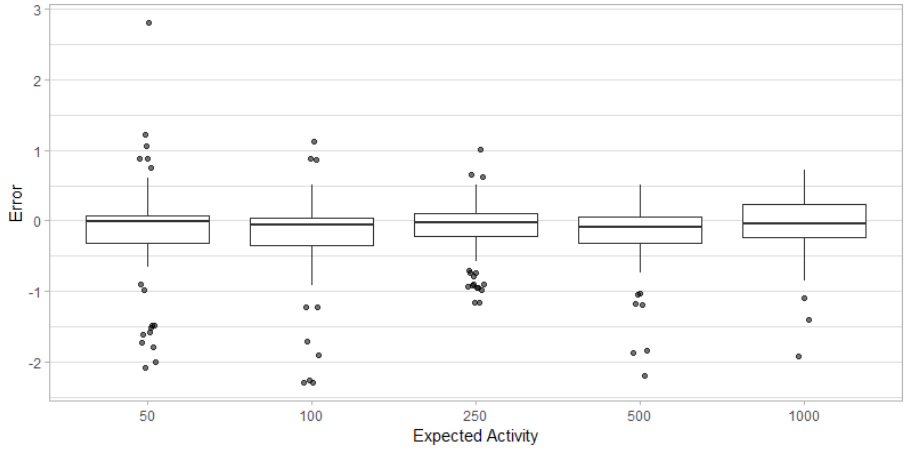


(c) Effect on μ estimation

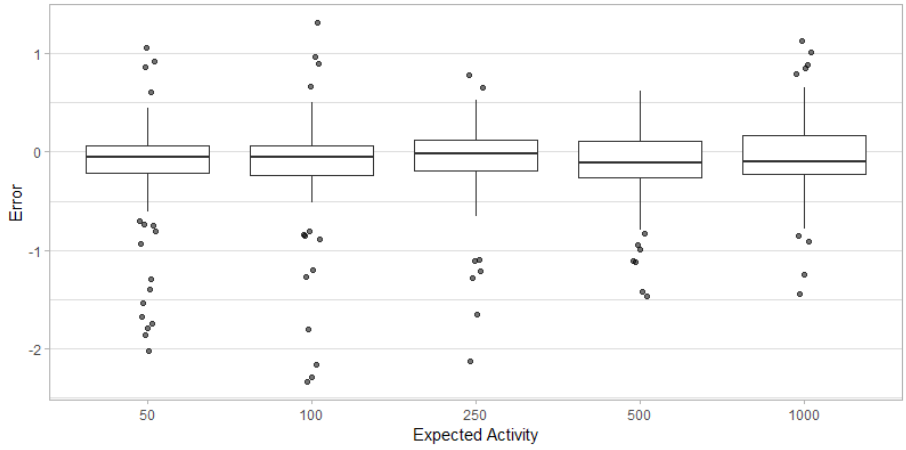
Figure 5.10.: Testing the Effect of the Range of η on the Supervised Learning method



(a) Effect on α estimation



(b) Effect on β estimation

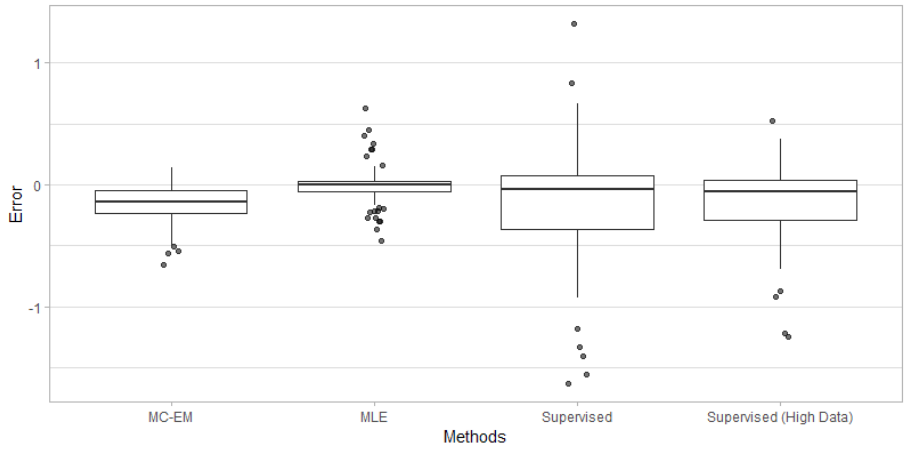


(c) Effect on μ estimation

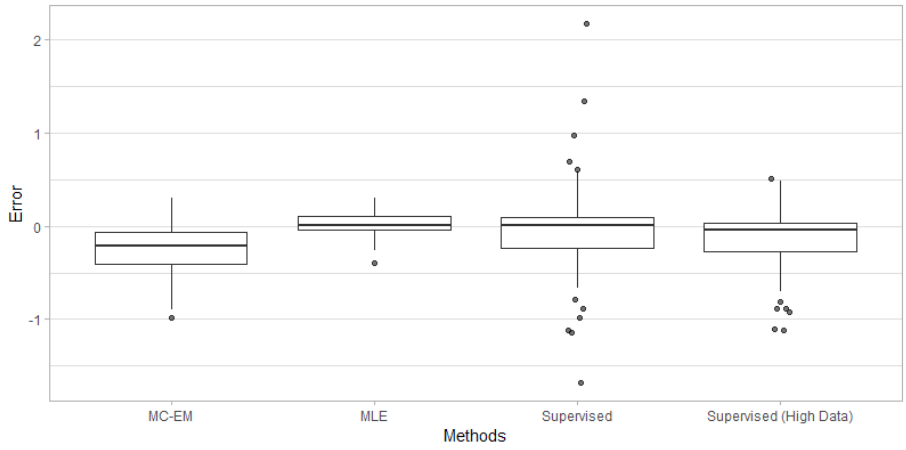
Figure 5.11.: Testing the Effect of the Value of Expected Activity on the Supervised Learning method

5.2.2. Comparison with MLE and the MC-EM Algorithm

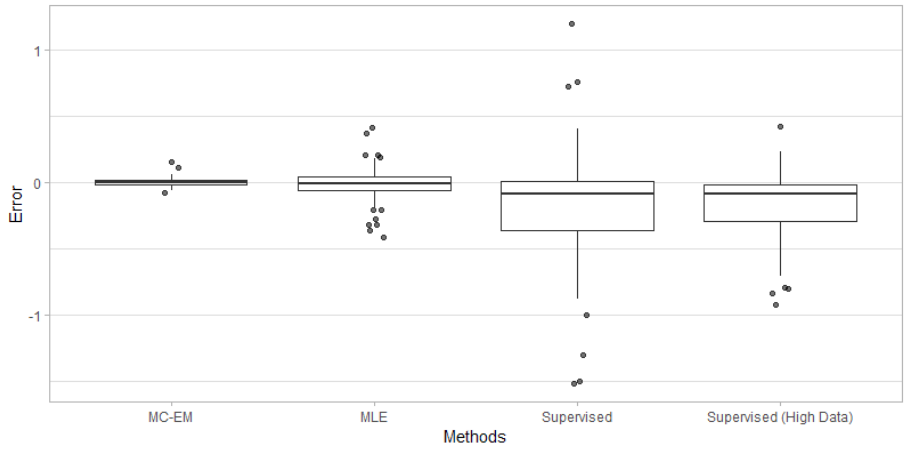
The parameters of the test which produced the results which are presented in this section were outlined in Section 4.3.4. This test comprises of a comparison of the estimation error produced by the supervised learning method, MLE, and MC-EM algorithm (Shlomovich et al., 2020) on 100 test processes. This comparison is presented in Figure 5.12. Examining Figures 5.12a and 5.12b, the supervised learning method with both quantities of data performs worse than the MC-EM algorithm. However, the difference in performance between the high data supervised method and the MC-EM algorithm is less significant. It should be noted that the high data estimation uses 20 times more data than that of the MC-EM algorithm. The largest outlier for the high data supervised learning method is double the magnitude of that of the MC-EM algorithm. Neither of the aggregated methods managed to perform as well as the maximum likelihood estimates, however this is as expected. The estimation of β is where the supervised learning method performs best when compared to the MC-EM algorithm with the distribution being very similar. This does not hold when the supervised learning method is used on the same quantity of data. Looking at Figure 5.12c, the remarkable performance of the MC-EM algorithm manages to significantly outperform the supervised learning method and even maximum likelihood estimation. It can be concluded that the MC-EM algorithm performs better across all three parameters. While the high supervised learning errors are close (for α and β) to the MC-EM algorithm, the error distributions have significantly longer tails for all three parameter estimates when an equal amount of data is used. This does not mean that the supervised learning algorithm is not a beneficial development, the arguments for which shall be presented in the next section.



(a) Comparison of α estimation



(b) Comparison of β estimation



(c) Comparison of μ estimation

Figure 5.12.: Comparison of Parameter Estimation Methods

5.2.3. Discussion of General Performance and Comparison

The results seen in Sections 5.2.1 and 5.2.2 have a followed a similar trend. It can be concluded that the tails of the errors distribution are long but that general performance of the algorithm is positive. It is possible that the tails of the distributions seen in the test environment are unfavourable on the method as they represent an unsupervised application of supervised learning algorithms. The hyper parameters used were not tuned for optimal prediction in each test and it is possible that these outliers could be minimised through proper application of these methods. This is particularly relevant for estimation of α which is conducted using a linear model. The data generated to fit the linear model was not checked for outliers or to ensure the underlying assumptions of a normal linear model were not violated. When taking this into account, it can be concluded that this method holds a large potential for prediction of parameters underlying aggregated Hawkes Processes. The training is robust to a range of parameters, most notably the level of discretisation. While this method did not achieve more favourable results at lower levels of discretisation as seen for other methods in Shlomovich et al. (2020), it achieved comparable results when a process was observed in steps of 5 time units. This performance was admirable, and would likely out-perform the MC-EM algorithm in this setting, though further testing would be necessary for this to be proven. Given the robust nature of the algorithm, with the caveat of proper training and hyper-parameter tuning, which is available when being deployed in a non testing environment, this algorithm poses value despite it being outperformed by the MC-EM algorithm.

The main benefit of this algorithm is a reduction in computational time. Once a neural network has been trained the estimation time is very low, allowing for hundreds of processes to estimated with ease. This is in contrast with the MC-EM algorithm which does not benefit from this speed up. It should also be noted that in its current implementation the MC-EM algorithm took multiple hours to conduct the tests seen in Figure 5.12, whereas the supervised learning method took less than 10 minutes including data generation. Therefore, the supervised learning method should be significantly faster even when neural network training time and hyper parameter tuning is accounted for. This was utilised in the high data test, as the supervised learning method could easily predict the parameters of more sub processes without a penalty in time. This allowed the supervised learning method to gain performance with little trade-off to run time. This would not be possible with the MC-EM algorithm as it would require a much longer run time and would have been unlikely to see the same gains in performance. It could be hypothesised that in applications where imprecise recording of event times leads to these aggregated Hawkes processes, such as finance or network data, that data quantity is unlikely to be a problem. Another advantage of the supervised learning method is that the main contribution to run time is the horizon and input dimension, as these will affect the number of neurons in the MLP to be trained. This means the training time is not affected adversely by an increase in expected activity. This does not hold for the MC-EM algorithm and this is the reason that expected activity was reduced from 5 events per time unit to 1 event per unit when compared to Section 5.2.1. The level of expected activity was set to 5000 for the comparison test initially, however this

was reduced to 1000 as the expected run time for the MC-EM algorithm was over one week. The efficiency of the MC-EM algorithm may improve as the concept is further developed but the sequential sampling used results in the computational times being linked to expected activity. As previously mentioned this does not affect the supervised learning method, which is dependent on the size of the neural network. The methodology outlined in the Section 4.3, could also be argued to be more accessible due to the simple nature of the MLP and the linear model used, both of which are widely understood and taught at undergraduate level. A further advantage is that the supervised learning method can utilise disjoint subsections of a process to improve performance, whereas the MC-EM algorithm requires a single uninterrupted process to make accurate estimation. Therefore, the supervised learning method has an advantage in which it can be applied in the case of disjoint observations.

6. Conclusion

This thesis aimed to address the problem of parameter estimation for Hawkes processes when the event times had been censored due to imprecise observation. This imprecise observation leading to aggregated Hawkes processes is a growing problem for conducting statistical inference on Hawkes processes in areas such as finance and cyber-security. The aim of this thesis was to alleviate these problems surrounding statistical inference using deep learning.

The work with Variational Auto-Encoders has established the viability of the Bayesian Inference proposed in Mishra et al. (2020) for aggregated Hawkes processes. The VAE with Poisson likelihood demonstrated the ability to reconstruct the underlying integrated conditional intensity in an unsupervised manner. This was achieved while also successfully encoding the branching ratio, η, and baseline intensity, μ, into the latent space. Under a Dueling Decoder framework, the Poisson decoder was used as a secondary objective to shape the latent space and, through the primary encoder, allowed for the estimation of the underlying parameters η and μ. This Dueling decoder framework was demonstrated to allow for the joint distribution of η and μ to be accurately estimated through MCMC sampling. This methodology has the potential to be a powerful technique which allows for statistical inference around the parameters of an aggregated Hawkes process despite the censored event times. The structure of the encoder and decoders are an element which may benefit from further development. For example, auto-regressive neural networks such as recurrent neural networks may posed value to this methodology. The level of compression achieved was 15% of the original dimension, however this is an area that also warrants further study. It is possible that the learning of disentangled latent representations is possible using a method such as β-VAE (Higgins et al., 2016).

The novel application of supervised learning algorithms proposed for the estimation of α, β, and μ was shown to perform robustly for many parameter ranges. Of particular note is the consistent estimation across different levels of discretisation. Despite the long tailed error distributions which were seen, the proposed method performed close to the level of the MC-EM algorithm (Shlomovich et al., 2020) with distinct advantages in computational time, accessibility, and flexibility regarding data. The lower computational time of the method proposed in this paper will lead to significant time savings, especially in the case of high expected activity processes. The flexibility regarding data and increased accessibility ensures this method has a wider range of use cases. Suggested areas for further study include an investigation into the viability of the method for other kernel forms, the potential of other supervised learning techniques, and the use of auto-regressive neural networks such as recurrent neural networks.

This thesis has established two new methodologies for performing statistical inference

surrounding parameter estimation of Hawkes processes when event times have been censored. This area of study has only begun to develop recently as an extension of Hawkes processes. The methods presented in this thesis form a significant contribution to the area and shall hopefully establish a foundation of study in conjunction with the work in Shlomovich et al. (2020).

Bibliography

E. Bacry and J. Muzy. Second Order Statistics Characterization of Hawkes Processes and Non-Parametric Estimation. *arXiv preprint arXiv:1401.0903*, 2014a.

E. Bacry and J.-F. Muzy. Hawkes Model for Price and Trades High-Frequency Dynamics. *Quantitative Finance*, 14(7):1147–1166, 2014b.

E. Bacry and J.-F. Muzy. First- and Second-Order Statistics Characterization of Hawkes Processes and Non-Parametric Estimation. *IEEE Transactions on Information Theory*, 62(4):2184–2202, 2016.

E. Bacry, K. Dayri, and J.-F. Muzy. Non-Parametric Kernel Estimation for Symmetric Hawkes Processes. Application to High Frequency Financial Data. *The European Physical Journal B*, 85(5):1–12, 2012.

E. Bacry, I. Mastromatteo, and J.-F. Muzy. Hawkes Processes in Finance. *Market Microstructure and Liquidity*, 1(01):1550005, 2015.

S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating Sentences from a Continuous Space. *arXiv preprint arXiv:1511.06349*, 2015.

F. Bre, J. M. Gimenez, and V. D. Fachinotti. Prediction of Wind Pressure Coefficients on Building Surfaces using Artificial Neural Networks. *Energy and Buildings*, 158: 1429–1441, 2018.

V. Filimonov and D. Sornette. Quantifying Reflexivity in Financial Markets: Toward a Prediction of Flash Crashes. *Physical Review E*, 85(5):056108, 2012.

V. Filimonov, D. Bicchetti, N. Maystre, and D. Sornette. Quantification of the High Level of Endogeneity and of Structural Regime Shifts in Commodity Markets. *Journal of international Money and finance*, 42:174–192, 2014.

H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin. Cyclical Annealing Schedule: A Simple Approach to Mitigating kl Vanishing. *arXiv preprint arXiv:1903.10145*, 2019.

H. Gabbard, C. Messenger, I. S. Heng, F. Tonolini, and R. Murray-Smith. Bayesian Parameter Estimation using Conditional Variational Autoencoders for Gravitational-Wave Astronomy. *arXiv preprint arXiv:1909.06296*, 2019.

A. G. Hawkes. Spectra of Some Self-exciting and Mutually Exciting Point Processes. *Biometrika*, 58(1):83–90, 1971.

A. G. Hawkes. Hawkes Processes and their Applications to Finance: A Review. *Quantitative Finance*, 18(2):193–198, 2018.

I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning Basic Visual Concepts with a Constrained Variational Framework. 2016.

D. P. Kingma and M. Welling. Auto-encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

D. P. Kingma and M. Welling. An Introduction to Variational Autoencoders. *arXiv preprint arXiv:1906.02691*, 2019.

M. Kirchner. Hawkes and INAR(∞) Processes. *Stochastic Processes and their Applications*, 126(8):2494–2525, 2016.

M. Kirchner. An Estimation Procedure for the Hawkes Process. *Quantitative Finance*, 17(4):571–595, 2017.

M. Kirchner and A. Bercher. A Nonparametric Estimation Procedure for the Hawkes Process: Comparison with Maximum Likelihood Estimation. *Journal of Statistical Computation and Simulation*, 88(6):1106–1116, 2018.

G. Last and M. Penrose. *Lectures on the Poisson Process*, volume 7. Cambridge University Press, 2017.

P. J. Laub, T. Taimre, and P. K. Pollett. Hawkes Processes. *arXiv preprint arXiv:1507.02822*, 2015.

E. Lewis and G. Mohler. A Nonparametric EM Algorithm for Multiscale Hawkes Processes. *Journal of Nonparametric Statistics*, 1(1):1–20, 2011.

D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference*, pages 689–698, 2018.

D. Marsan and O. Lengline. Extending Earthquakes' Reach through Cascading. *Science*, 319(5866):1076–1079, 2008.

S. Mishra, S. Flaxman, and S. Bhatt. π VAE: Encoding Stochastic Process Priors with Variational Autoencoders. *arXiv preprint arXiv:2002.06873*, 2020.

G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita. Self-Exciting Point Process Modeling of Crime. *Journal of the American Statistical Association*, 106(493):100–108, 2011.

Y. Ogata. On Lewis' Simulation Method for Point Processes. *IEEE transactions on information theory*, 27(1):23–31, 1981.

Y. Ogata. Space-Time Point-Process Models for Earthquake Occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402, 1998.

J. F. Olson and K. M. Carley. Exact and Approximate EM Estimation of Mutually Exciting Hawkes Processes. *Statistical Inference for Stochastic Processes*, 16(1):63–80, 2013.

T. Ozaki. Maximum Likelihood Estimation of Hawkes' Self-exciting Point Processes. *Annals of the Institute of Statistical Mathematics*, 31(1):145–155, 1979.

E. Perez. *The Speed Traders: An Insider's Look at the New High-Frequency Trading Phenomenon That is Transforming the Investing World*. McGraw Hill Professional, 2011.

I. Rubin. Regular Point Processes and their Detection. *IEEE Transactions on Information Theory*, 18(5):547–557, 1972.

B. Seybold, E. Fertig, A. Alemi, and I. Fischer. Dueling Decoders: Regularizing Variational Autoencoder Latent Spaces. *arXiv preprint arXiv:1905.07478*, 2019.

L. Shlomovich, E. Cohen, N. Adams, and L. Patel. A Monte Carlo EM Algorithm for the Parameter Estimation of Aggregated Hawkes Processes, 2020.

Stan Development Team. RStan: The R Interface to Stan, 2020. URL `http://mc-stan.org/`. R package version 2.21.2.

M. J. Turcotte, A. D. Kent, and C. Hash. Unified Host and Network Data Set. *arXiv preprint arXiv:1708.07518*, 2017.

A. Veen and F. P. Schoenberg. Estimation of Space–Time Branching Process Models in Seismology using an EM-type Algorithm. *Journal of the American Statistical Association*, 103(482):614–624, 2008.

R. Zaatour. *Hawkes: Hawkes Process Simulation and Calibration Toolkit*, 2014. URL `https://CRAN.R-project.org/package=hawkes`. R package version 0.0-4.

H. Zhao, P. Rai, L. Du, W. Buntine, D. Phung, and M. Zhou. Variational Autoencoders for Sparse and Overdispersed Discrete Data. In *International Conference on Artificial Intelligence and Statistics*, pages 1684–1694, 2020.

A. Appendix

A.1. Code Repository

The code used to generate the results seen in this thesis is available on the following Github:

`https://github.com/tom-keane/Statistical_Inference_for_Hawkes_Processes_with_Deep_Learning`

Please note, the code which generated the results for the MC-EM algorithm is withheld at the request of Shlomovich et al.. This is due to the github being public, and an agreement between myself and Shlomovich et al. to withhold the results from the public until their paper (Shlomovich et al., 2020) has passed peer review.