Bayesian Nonparametric Hawkes Processes with Applications

Dean Markwick

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Statistical Science

University College London

August 29, 2020

Declaration

I, Dean Markwick, declare that this dissertation represents my own work, except where due acknowledgement is made.

Signed: _____

Date: _____

Abstract

Many statistical problems involve modelling the times at which events occur. There are cases where events can occur in clusters with sudden jumps in the total number of occurrences. To model such data an intensity function can be constructed which describes the probability of an event occurring at a specific time. The Hawkes process is a point process model with a conditional intensity function that provides a change in intensity for each event occurrence and as such the Hawkes process can be used to explain event clustering. The flexibility and extendability of the Hawkes process will be highlighted in this thesis.

I extend the Hawkes process by using nonparametric Bayesian methods where different components of the Hawkes process are constructed using a Dirichlet process which is a Bayesian prior for distributions. This allows for a data driven approach and removes the need for parametric assumptions. This Bayesian approach also allows for a hierarchical structure to be integrated in the models where appropriate.

These extended Hawkes process are applied to different application domains including: extreme value theory, financial trading and soccer goal occurrence modelling. Each new application introduces a different extension to the Hawkes process and illustrates how it improves on existing methodology.

From this research I also wrote a new software package for using Dirichlet processes. This software enables users to easily construct Dirichlet process objects that can incorporated into existing inference workflows. This allows users to introduce nonparametric methods without needing to program their own inference methods.

Impact Statement

Hawkes processes are becoming ubiquitous across different disciplines and have seen application in geology, finance and many more research areas. This thesis continues this trend and shows how Hawkes processes are a generalisable model that can be used widely. By extending and demonstrating this methodology future research can be conducted following the same steps and adapting where necessary.

The use of probabilistic methods in this thesis is impacting the development of Bayesian computational methods for the Hawkes process by providing a flexible algorithm for inference of a Hawkes process. Furthermore, the extension into nonparametric methods improves on the existing methodology for data-driven approaches. This flexible algorithm, combined with numerous application examples shows how the Hawkes process can have a major impact on many different areas.

Outside of academia there are numerous uses for Hawkes processes. The financial technology firm BestX are currently implementing the hierarchical nonparametric Hawkes process work (Chapter 5) in their software. This research is enabling them to understand current market conditions based on the recent pattern of trading. BestX are the industry standard in transaction cost analysis and currently have over 100 clients that are responsible for roughly $20 trillion in assets under management.

Secondly, the usage of Hawkes processes as an in-play model for odds movements has direct impact for betting exchanges and odds compilers. The work in Chapter 6 shows how a generative model of goals in a match can lead

to a prediction of the winner of the match. There are many more betting markets that are based on goals scored and a Hawkes model provides a very general framework that can form the basis of adjusting prices in real time.

The software package developed as part of this thesis in Chapter 3 has also had a great impact on the approach of nonparametric Bayesian modelling. Since release it has over 5,000 downloads, with 15 downloads per month on average. It is currently hosted on Github where I have helped numerous people both use and contribute to the software. Finally, it has received its first official citation in the publication Koenker and Gu (2019).

Acknowledgements

I would like to thank my primary supervisor Gordon Ross for his academic guidance throughout my PhD. Similarly, I would also like to thank Paul Northrop for his advice and support throughout the final stages of the process. I thank Pete, Ollie, Aman and the rest of the team at BestX for giving me the opportunity to develop and apply some of this research to a real world problem.

None of this would have been possible without my mother, father and brother supporting my decision embark on this journey and likewise for my friends, keeping me on track and providing the life in work-life balance. Finally, I would like to thank my better half Sophie for the unconditional support, love and advice. Without you it would have been twice as difficult and half as fun.

Contents

List of Figures

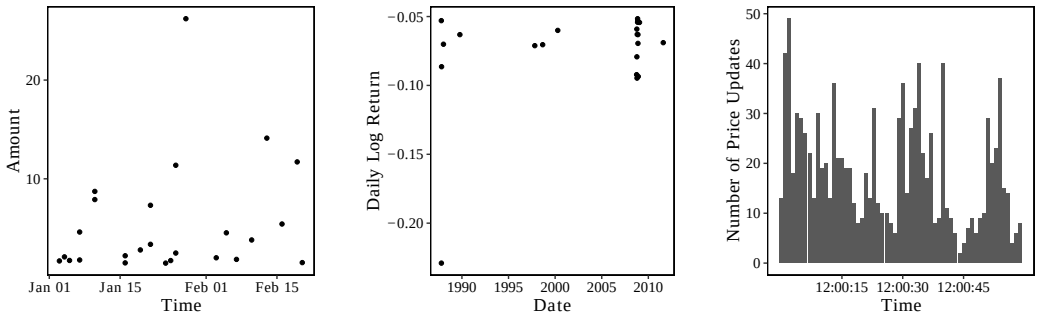
List of Tables

Chapter 1

Introduction

A point process is a mathematical model that describes patterns of random points in a given area. This area could represent a time period where the points describe when a random event occurs or the area could also be used to describe a physical space where the random points describe locations within this space. As a result, point processes have far reaching and diverse applications, from constructing life tables by counting the number of deaths in a time period using mortality data (Graunt, 1973) to counting the number of phone calls down a telephone wire (Erlang, 1909). This thesis focuses on a specific type of point process called the Hawkes process which is used to model the clustering behaviour of the points. These points will refer to event occurrence times in a predefined window of time.

Events from a point process will display clustering when there is a higher concentration of events around other events. There will periods of time where many events occur and there is a higher probability of seeing further events around other events. There are many examples of the clustering of event times in a wide variety of different fields and Figure 1.1 shows three different examples of clustering situations. Figure 1.1a shows the time and value of insurance claims in Denmark for two months in 1980 where the data is taken from the `evir` R package (Pfaff and McNeil, 2018). In this example, the points represent when the claims occurred and they appear to be clustered around each other. Similarly Figure 1.1b shows the occurrences of losses greater than



(a) The dates of fire insurance claims in Denmark.

(b) Days when the price change in the S&P 500 was greater than -5%.

(c) Number of price changes in a second of the Euro and Dollar exchange rate.

Figure 1.1: Example of clustering in the times of event across different data.

5% of the S&P 500, a financial index. There are three distinct periods where these losses occur, the financial crisis of 1987, the dot-com bubble of the early 21st century and the great financial crisis of 2008. There are multiple days where the occurrence of large losses cluster around each other and shows large losses are observed in bursts. Finally, Figure 1.1c shows the number of price changes in a second of the Euro to US Dollar exchange rate over the course of one minute. There is a highly variable rate of price changes where sudden flurries of increased activity are closely followed by further periods of higher activity which is another example of clustering.

One such mechanism to model this clustering behaviour is self-excitation which is a phenomena where the occurrence of an event can increase the future rate of the same type of event. Each of these three examples show how clustering can be viewed as a consequence of self-excitation and suggests that the Hawkes process can be used to model the occurrence of these event times.

A point process is defined by an intensity function that controls the instantaneous probability of an event occurring at a given time. The specific form of the Hawkes process is represented by a conditional intensity function

$$\lambda(t \mid H_t) = \mu(t) + \kappa \sum_{t_i < t} g(t - t_i),$$

where H_t is the history of the process and represent the information filtration up to time t, $\mu(t)$ is the baseline intensity function of events, κ is a positive constant value and $g(t)$ is another positive function. The self-excitation comes from the $\kappa \sum_{t_i<t} g(t-t_i)$ components of the function as with each event that occurs there is a increase of κ in the intensity function that decays at a rate of $g(t)$. Each event that occurs is either generated from the rate $\mu(t)$ or the result of an increase in intensity from previous events that the self-excitation component controls.

The Hawkes process was first introduced in Hawkes (1971) and has seen wide ranging applications ever since. It started as a model for earthquakes (Adamopoulos, 1976; Ogata, 1988) and researchers have since used it to model different problems in fields such as ecology (Balderama et al., 2012), criminology (Mohler, 2013; Porter and White, 2012) and finance (Chavez-Demoulin and McGill, 2012; Filimonov and Sornette, 2012; Rambaldi et al., 2015). In each case, self-exciting behaviours have been found to exist and modelled using a Hawkes process.

Any data where there is a possibility of clustering or self-excitation between events is well suited for a Hawkes process and this thesis will explore a number of different areas where the Hawkes process can be applied and extended.

Like all statistical models, the Hawkes process has a number of free parameters, $\mu(t)$, κ and $g(t)$ that must be inferred from the data at hand. Current popular inference techniques are frequentist in nature and consist of maximising the likelihood function (Ogata, 1988; Lallouache and Challet, 2016; Rambaldi et al., 2015) or expectation maximisation (Veen and Schoenberg, 2008; Lewis and Mohler, 2011). This thesis will be taking a different approach instead and proposes a Bayesian method for inferring the parameters of the model which will provide a number of benefits over the frequentist methods.

Similarly, these parameters of the Hawkes process are typically assumed to be of a known form that can be represented by a finite number of parameters,

for example, a constant baseline intensity $\mu(t) = \mu_0$, where μ_0 is unknown and must be inferred, or an exponential decay for $g(t) = \beta e^{-\beta t}$, where β is unknown. Instead, this thesis takes the next step and uses nonparametric methods. This use of nonparametric models removes the need to make potentially incorrect assumptions about the shape of the data and instead allows for arbitrary distributions to be learnt from the data instead. Nonparametric Hawkes processes have been explored previously (Lewis and Mohler, 2011) but this thesis will bridge the gap between nonparametric statistics, Bayesian methods and how they can be applied to the Hawkes process.

A Bayesian nonparametric approach provides a general method for estimating Hawkes processes. It minimise the need for assumptions of the form of the different components of the Hawkes process such that μ, κ or $g(t)$ can be learnt from the data instead. Furthermore, this allows for a general model to be applied to different datasets reducing the amount of fine tuning required for each application. Specifically in the case of $g(t)$ where the dynamics of the dataset can make the choice of the function difficult, using a Bayesian nonparametric method helps removes the need to choose and instead lets the data decide on the most suitable function shape.

Both $\mu(t)$ and $g(t)$ will be modelled nonparametrically in this thesis. For $\mu(t)$ the use of a nonparametric model will allow for a more flexible approach when it comes to modelling seasonality that is present in data. For $g(t)$ it will reduce the need to make specific assumptions about the data and instead a suitable form of $g(t)$ will be learnt from the data rather than imposed.

The Dirichlet process will drive the nonparametric approach and provides a Bayesian method for specifying a prior distribution for the nonparametric model. The Dirichlet process is a type of stochastic process where each draw is a distribution and also possess beneficial computational properties that aids in the sampling of the posterior distribution. This combination of mathematical and computational elegance allows for a versatile nonparametric model that can be used by the Hawkes process in variety of ways.

However the nonparametric methods are not without their drawbacks. There is an increased computational cost when using such methods to infer the parameters. This can prohibit the use of such methods on very large datasets where the running time would be too long compared to frequentist parametric methods. Therefore there is always the need to judge what technique would be most practical for the size of the data.

Overall this thesis will be exploring the use and application of Hawkes processes. By proposing a Bayesian algorithm for sampling the parameters of a Hawkes process this thesis will be providing the necessary framework to build upon more complex point process models. The framework will be developed with a highly applicative view and each new aspect of the Hawkes process is followed by a suitable application, ensuring that the research has real world impact and each new addition to the Hawkes process has a clear use case. The Hawkes process will be extended using nonparametric and hierarchical methods with topics of application including: extreme value theory, quantitive finance and sports modelling.

1.1 Contributions

Four papers are being submitted to journals from work described in this thesis. Firstly, I have independently written and distributed an R package for fitting Dirichlet process models. *dirichletprocess: An R Package for Fitting Complex Bayesian Nonparametric Models* is the accompanying vignette to this software package and forms Chapter 3. From the work in Chapter 4 I have also written the paper *Hierarchical Bayesian Modelling of Nonstationary Extreme Values.* In this paper the Hawkes process is applied to extreme values and provides a new framework for modelling both the occurrence and magnitude of an extreme event. Chapter 5 has also spawned a paper that is in the process of submission; *Hierarchiacal Non Parametric Hawkes Processes with Applications* where the Hawkes process is used with a nonparametric component and applied to multiple timeseries of financial data. Finally the work on multivariate Hawkes

processes from Chapter 6 forms the paper *Bayesian Multivariate Hawkes Processes with Applications to Soccer Goals*. All four of the papers described here are in the process of submission.

1.2 Thesis Outline

In Chapter 2 the early Hawkes literature is reviewed and the necessary background mathematical detail is outlined. An overview of the Hawkes process with its associated equations and current inference methods is presented before proceeding to a brief introduction of Bayesian statistics and nonparametric methods all of which will be used throughout the thesis.

The Dirichlet process forms the basis of the nonparametric work and is frequently used in chapters of this thesis. This work lead to the development of a software package for creating flexible Dirichlet processes objects in `R`: `dirichletprocess`. Chapter 3 showcases the features and implementation of the package with examples of how users can perform nonparametric Bayesian analysis using Dirichlet processes. The package allows users to perform nonparametric modelling without the need to program the inference algorithms and instead the user can utilise the prebuilt models or specify their own models whilst allowing the `dirichletprocess` package to handle the Markov chain Monte Carlo sampling. The Dirichlet process objects from the package can act as building blocks for a variety of statistical models including and not limited to: density estimation, clustering and prior distributions in hierarchical models.

For Chapter 4 the Hawkes process is applied to extreme value theory and a new framework is proposed for modelling extreme events. The Hawkes process is combined with a Dirichlet process to model the occurrence of the extreme values nonparametrically and then by using the learnt structure between events, the magnitude of the extreme values is also modelled. The Hawkes model provides a conditional exceedance distribution that assesses the probability of an extreme event whilst taking into account the history of the

process. A full posterior sampling algorithm for the Hawkes process parameters is introduced and also used in further chapters. This chapter uses extreme terrorist attacks as its application example and provides a prediction for both when and how large the next attack will be given a recent terror attack. In Chapter 5 the occurrence of trades in the foreign exchange market are modelled using a Hawkes process. There is strong seasonal day-of-the-week effects where the behaviour of a financial market on a Monday is quite different to the behaviour on a Friday. To account for this, a further nonparametric extension of the Hawkes process is developed which uses a hierarchical Dirichlet process to learn multiple day-of-the-week seasonality functions simultaneously. This new model is shown to accurately predict current market conditions given the recent trading activity.

A multivariate Hawkes process is under consideration in Chapter 6. Soccer goals are found to *not* be generated by a Poisson process and thus a multivariate Hawkes process is a suitable model choice. Using the same approach as previous chapters it is shown how the Bayesian inference algorithm can be extended to multiple variables without major changes in the algorithm. This multivariate Hawkes process is then used to explore the occurrences of soccer goals and how the two teams scoring rates can experience both self and mutual excitations. The scoring rates are then used to form predictions of a match outcome which are found to agree with prediction formed from the available market odds.

All three applications outlined above involve a clustering effect where events appear in bursts. The Hawkes process is a natural choice of model where self-excitation can be used to induce this type of clustering behaviour. In all three cases (terror attacks, trades and goals) events can lead to further events of the same type. It is this size and scale of self-excitation that make the Hawkes process well suited to these different problems.

Finally, the findings are concluded in Chapter 7 and further possible avenues of further work are outlined.

Chapter 2

Background

This chapter begins with an overview of the early literature on Hawkes processes and outlines the original usage of the Hawkes process. After this historical reference the necessary background mathematical material is outlined for understanding the Hawkes process and nonparametric Bayesian statistics. The concept of a point process and Bayesian analysis form the foundation of the work in this thesis and each contribute to the development of the nonparametric Hawkes process.

The Hawkes process was first introduced in 1971 by Alan G. Hawkes (Hawkes, 1971) where it is proposed that a new class of point processes with intensity functions that are dependent on the history of the process can be well defined. This paper then proves that such a process cannot be distinguished from one where the intensity function is random and Hawkes (1971) concludes by proposing a number of domains where this self-exciting process could be applied, such as epidemic models with different infection cases and the physical phenomena of radiating bodies. Hawkes and Oakes (1974) extended this by proving that this self-exciting process can also be represented using clusters of Poisson processes. Each event can produce a further generation of events triggered by some rate and this rate is a defining property of the Hawkes process. Adamopoulos (1976) used a self-exciting model for modelling global earthquakes in years 1950-1971. This is one of the first applications of the Hawkes process and showed how a point process with a clustering property is

well suited for modelling earthquakes.

The early applications of the Hawkes processes were focused on seismology and much of its development revolved on the problems that arise in earthquake modelling. One such problem involves identifying which earthquake was the main event and which earthquakes are the aftershocks caused by the main event. As the earthquakes appear in clusters both in time and location it can be difficult to form a causal structure and show what earthquake caused other earthquakes.

This lead to the significant applications of a Hawkes process and the introduction of the Epidemic Type Aftershock Sequence (ETAS) model in Ogata (1988). This paper uses the previous development of the Hawkes process and applies it to the occurrence and magnitude of Japanese earthquakes from 1885 to 1980. The Hawkes process was found to fit the data better than the standard models of that time but reiterated the difficulty in identifying the main and aftershock earthquakes.

The process of untangling this structure between earthquakes is known as declustering and is a key technical consideration when fitting Hawkes processes. Zhuang et al. (2002) again used the ETAS model and a variety of algorithms to determine which earthquakes were main events and which were aftershocks. However, all algorithm proposals were dependent on model choice and thus the functional form of the ETAS model. Furthermore, as the process is stochastic and assigns probabilities to the structure between events each iteration is just a realisation of the true structure in the data. This declustering will prove an important tool for parameter inference.

A simple extension to a point process has lead to a new model where clusters of events can be explained by self-excitation behaviour. This new point process has then seen a wide range of applications and proves that many different phenomena can be seen as self-exciting.

As mentioned in the introduction, Hawkes processes have since been applied to a wide range of different research areas. Balderama et al. (2012)

applied the ETAS model to the spreading of an invasive plant species in Costa Rica and found that the presences of a plant increased the further presence of more plants. In criminology Mohler (2013) use a Hawkes process to model the occurrences of criminal activity in Chicago and also terrorist attacks in Israel, Northern Ireland and Iraq. Similarly Porter and White (2012) also used terrorist attacks in Indonesia as their example data set for applying a self-exciting model.

A large area of work in Hawkes processes has also been dedicated to financial applications such as high frequency trade dynamics (Bacry et al., 2012; Chavez-Demoulin and McGill, 2012; Filimonov and Sornette, 2012), risk modelling (Chavez-Demoulin et al., 2005) and the impact of macroeconomic news on trading intensity (Rambaldi et al., 2015). In each case the clustering behaviour is mechanised by the self-excitation behaviour from the Hawkes process. For a review on the use of Hawkes processes in finance by Hawkes himself see Hawkes (2018).

Finally there has also been experimentation with neural networks and Hawkes processes as demonstrated in Mei and Eisner (2017). In this work they use a neural network to replace the simple summation of event self-excitation which allows for a complex dependence between events and their effects on the future intensities. This model is applied to a variety of datasets including social network interactions, health care visits and financial trades and improves prediction of future events in all cases.

2.1 Point Processes

A point process is a mathematical model for describing a collection of items randomly located in some space. If this space is a time axis then these items refer to event occurrences and are located on the real line. For example, the arrival time of buses at a bus stop forms a point process, where each event is the arrival time of a bus and the space in consideration is the time window in which the experiment takes place.

For any point process there exists an intensity function that describes the probability of an event occurring

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{\Pr(N(t+\Delta t) - N(t) = 1)}{\Delta t},$$

where $N(t)$ is the number of events at time t. For a small interval $[t, t+\Delta t]$, the probability of one event occurring is equal to the intensity function at time t as the vanishing limit of Δt is taken.

The Poisson point process is the most commonly used point process and is characterised by two properties. Firstly, the total number of points in the window is distributed via the Poisson distribution, and secondly, the occurrence of each point is independent of the other. Furthermore, given arbitrary non-overlapping time intervals $A_1, A_2, \ldots$ where $N(A_1), N(A_2), \ldots$ are the respective counts of events in these intervals. If the points arrive as a Poisson process with intensity $\lambda(t)$ then the random variables $N(A_1), N(A_2), \ldots$ have independent Poisson distributions such that the probability of n events in the interval A_i can be written as

$$\Pr(N(A_i) = n) = \frac{\Lambda(A_i)^n}{n!} e^{-\Lambda(A_i)},$$

where $\Lambda(A_i)$ is the intensity function integrated over the interval

$$\Lambda(A_i) \equiv \int_{A_i} \lambda(t) \mathrm{d}t.$$

As this is a Poisson distribution, the expected number of points in each interval can be written as

$$\mathbb{E}\left[N(A_i)\right] = \int_{A_i} \lambda(t) \mathrm{d}t, \quad i = 1, 2, \ldots, \tag{2.1}$$

therefore the counts within any non-overlapping sets are independent. Furthermore, the counts are independent of the history of the process and the past behaviour has no effect on the future occurrences of events.

If the intensity function $\lambda(t)$ is a constant and does not depend on any other variable it is known as the homogeneous Poisson process. If the process has a dependence on another variable, such as time, it is referred to as

an inhomogeneous Poisson processes as long as the above statements remain satisfied.

2.2 Hawkes Processes

The inhomogeneous Poisson process is extended further and includes an effect where the intensity function is now dependent on the past events. This intensity function is now *conditional* on the history up to time t of the process and can be written as

$$\lambda(t \mid H_t) = \lim_{\Delta t \to 0} \frac{\mathbb{E}\left(N([t, t+\Delta t]) \mid H_t\right)}{\Delta t},$$

where H_t represent the set of past events.

The Hawkes process is one such conditional model where the form of intensity function can be written as

$$\lambda(t \mid H_t) = \mu(t) + \kappa \sum_{t_i < t} g(t - t_i), \tag{2.2}$$

where $t_i, i = 1, \ldots, n$ are the event times, $\mu(t)$ is some positive function, κ a constant and $g(t)$ an arbitrary function.

Under this parametrisation and given that $\kappa > 0$ the Hawkes process can be viewed as a branching process and with each event occurrence there is a corresponding increase in the intensity function. If an event does occur from this subsequent increase in intensity then it can be interpreted as an offspring event from the original event. For each event from the process there will be a corresponding number of offspring events (which could be zero). Furthermore, each offspring generation is independent of the other events and each event occurring can cascade into further events which is the signature for self-exciting behaviour. This branching interpretation hinges on the linear superposition of Poisson processes, which itself is a Poisson process (Kingman, 1992) and by being able to separate out the individual Poisson process generating the events a causal structure can be established. From this structure the links between events can be labelled and used to aid inference.

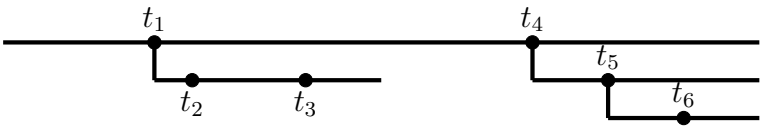


Figure 2.1: Graphical representation of the structure of events arising in a Hawkes process. Each black circle represents an event and shows how we can identify parent events. Only three layers are shown but there can possibly be an infinite amount of layers with events.

Figure 2.1 shows a realisation of the Hawkes process and how the branching structure appears naturally. Events t_1 and t_4 are independent, they have no parent event and spawned randomly. Events t_2, t_3, t_5, t_6 have clear parents and there exists a causal structure between these events and their parent event - these events are caused by the increase in the intensity function. However, whilst this structure between events can be constructed it does not restrict the Hawkes process to situations where there is causal link between events. This self-exciting behaviour is used to describe the clustering of events rather than the causality.

The parameters of the Hawkes intensity function, Eq. (2.2), have an interpretable effect on event generation. Firstly, $\mu(t)$ is the background rate - the rate at which all random events which have no parent are generated, κ is the expected number of child events for each event and $g(t)$ is the distribution of child event times relative to their parent events. For stability, $\kappa < 1$ otherwise the total number of events would explode.

The causal structure between events leads to the introduction of a latent variable for each n events $\mathbf{B} = \{B_1, B_2, \ldots, B_n\}$ that describes the branching structure. If event i was generated by the background rate then $B_i = 0$, or, if event t_i was caused by t_j then $B_i = j$. This variable then allows the formation of two different group, background events and child events. As it is known that the background rate is responsible for the background events and κ and $g(t)$ are responsible for the child events then this latent variable can aid in the estimation of the Hawkes parameters. In practice $\mathbf{B}$ must be estimated as the

true causal structure between events in a real dataset is unknown.

2.2.1 Multivariate Hawkes Processes

Hawkes processes are not limited to the univariate case where events can self-excite leading to more events of the same type. They can be extended across multiple event types where both an event can spawn another event of both the same and different types. To use a financial example from Muni Toke and Pomponio (2011), both buy orders and sell orders of a stock could be considered two different types of events, buy orders could lead to more buy orders and could also excite sell orders. This excitement across both the same dimension and different dimensions defines the multivariate Hawkes process.

Equation (2.2) is extended to account for m dimensions in the time space. This means that the data now consists of m series of time events t_{ij} where i labels the dimension and j labels the event occurrence. Both dimension and event type can be used interchangeably.

There are now m intensity functions, one for each event type

$$\lambda_i(t \mid H_t) = \mu_i(t) + \sum_{j=1}^{m} \kappa_{ij} \cdot \sum_{t_{jk}<t} g_{ij}(t - t_{jk}), \tag{2.3}$$

where again H_t is the entire history of the process, i.e. all the events in all dimensions up to time t.

It can be helpful to imagine that each κ parameter is part of a matrix

$$K = \begin{pmatrix} \kappa_{11} & \dots & \kappa_{1m} \\ \vdots & \ddots & \vdots \\ \kappa_{m1} & \dots & \kappa_{mm} \end{pmatrix},$$

where the diagonal elements are responsible for self-exciting and the off diagonal elements control cross-excitations between dimensions.

Similarly, the kernel functions $g_{ij}(t)$ can also follow the same matrix explanation. Diagonal elements control the decay of the self-exciting impulse

and off diagonal elements control the decay of other dimensional excitements.

$$G = \begin{pmatrix} g_{11} & \cdots & g_{1m} \\ \vdots & \ddots & \vdots \\ g_{m1} & \cdots & g_{mm} \end{pmatrix},$$

Therefore, for m dimensions of the Hawkes process there are m^2 κ values and m^2 kernel parameters that can be assigned. Again, like the univariate Hawkes process, the kernel function is a probability distribution that integrates to unity.

2.2.2 Simulation

At the most basic level, the Hawkes process is an inhomogeneous Poisson process (IHPP) that experiences a change in intensity after every event occurs

$$\lambda(t \mid H_t) = \begin{cases} \mu(t), & 0 < t < t_1 \\ \mu(t) + \kappa g(t - t_1), & t_1 < t < t_2 \\ \mu(t) + \kappa g(t - t_1) + \kappa g(t - t_2), & t_2 < t < t_3 \\ \text{etc}, & \end{cases}$$

therefore the Hawkes process consists of a superposition of multiple IHPP which allows for an elegant simulation of such a process.

Firstly, all the background events must be generated. This is achieved by simulating a Poisson process with rate $\mu(t)$ in the $[0, T]$ interval. Then for each event t_i, a further Poisson process with rate $\kappa g(t)$ is simulated in the interval $[t_i, T]$ and for each event that occurs in this interval the parent event is t_i. Simulations of IHPP can be performed using thinning (Lewis and Shedler, 1979) which is the standard method of simulating from a IHPP.

This approach to simulation highlights the clustering nature of event occurrence and how the clusters can be assigned a causal structure based on what event was responsible. Figure 2.2 is an example of a Hawkes process being simulated. The parameters are held constant at $\mu = 0.5, \kappa = 0.5$ and $g(t) = 0.5 \exp(-0.5t)$ and the process was simulated from $[0, 10]$. The colouring of the points indicates the parent event B_i so that the red dots are the

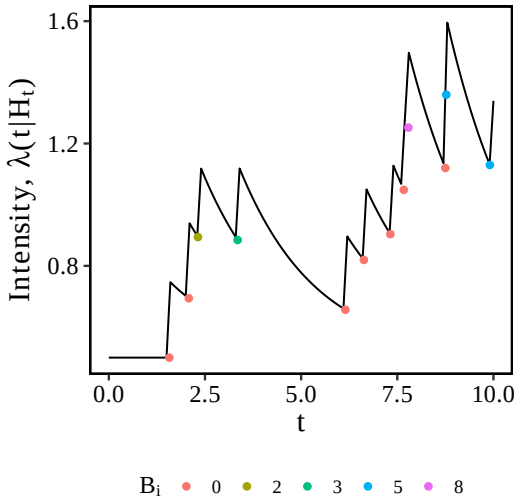


Figure 2.2: A Hawkes process simulation with constant parameters. The location of each dot indicates at what time the event occurred, the colour of each dot indicates the parent of the event.

background events and the other colours show the events whose parent event is a previous event. From this illustration it is shown that the first few events are generated from the background before cascading into more events from the previous events.

2.2.3 Frequentist Inference

As mentioned in the introduction, previous work on the Hawkes process has been largely frequentist in nature. In the available literature maximum likelihood procedures are used to infer the parameters of the Hawkes process (Porter and White, 2012; Mohler, 2013; Balderama et al., 2012) where full likelihoods of the models are composed before being numerically optimised to find the parameters that maximise such likelihoods.

For a general point process the log likelihood can be written as (Daley and Vere-Jones, 2003)

$$\log \mathcal{L}(t_i \mid \Theta) = \sum_{i=1}^{n} \log \lambda(t_i \mid H_{t_i}) - \int_0^T \lambda(t \mid H_t)\mathrm{d}t, \tag{2.4}$$

where Θ is the set of unknown parameters. For the Hawkes process, the intensity is given as (2.2) and the unknown parameters are $\Theta = (\mu(t), \kappa, g(t))$.

This likelihood can be optimised numerically which produces the estimates of the parameters that give the maximum value of the likelihood given the data. There a variety of methods that can be used to estimate this maximum such as the Nelder Mead method (Nelder and Mead, 1965) which is a standard optimisation method or even genetic algorithm based approaches such as that used in Chavez-Demoulin and McGill (2012) where they fit a Hawkes process using a differential genetic algorithm from Storn and Price (1997).

There are a number of potential pitfalls that can inhibit an optimisation task (Weise et al., 2009). Firstly, it must be ensured that the global maximum of the function under consideration is found and not just a local maximum. This local maximum will be misleading and lead to a premature optimal answer that is incorrect. Similarly, the likelihood function might be very flat around the maximum value thus leading to a larger uncertainty as to where the true maximum parameters are located. Both problems are especially troublesome for maximum likelihood estimation of the Hawkes process as the likelihood can be both multimodal and flat (Veen and Schoenberg, 2008). This also motivates a Bayesian method of inference where the uncertainty in parameter estimation can be obtained directly from the sampling procedure.

These potential complications of direct maximum likelihood estimation has lead to other approaches being used to infer the parameters of a Hawkes process.

Expectation Maximisation

Expectation maximisation (EM) is used in parameter inference of Hawkes models (Veen and Schoenberg, 2008) to remedy the problems of direct optimisation of the likelihood. The expectation maximisation algorithm consists of two steps, estimating the log-likelihood and then choosing parameters to maximise that likelihood. These two steps are then iterated until convergence of the parameter estimates is achieved.

Expectation maximisation was introduced in Dempster et al. (1977) to provide a method for estimating the maximum likelihood from incomplete data

sets. For a Hawkes process, the missing part of the data can be interpreted as the unknown background and child event structure (the latent variable $\mathbf{B}$). Using the expectation maximisation algorithm provides a method for estimating this structure and then using this information to estimate the Hawkes process parameters.

The first step in this algorithm is to estimate the expectation of the likelihood function and to estimate whether an event was caused by the background rate or a previous event. To establish an events parent, a probability of parent for each event must be calculated where the background probabilities are labelled as p_{ii} and the probability that event i is caused by event j is labelled as p_{ij}. These probabilities are written as

$$p_{ij}^k = \frac{\kappa^k g^k(t_i - t_j)}{\mu^k + \kappa^k \sum_{j=1}^{i-1} g^k(t_i - t_j)},$$
$$p_{ii}^k = \frac{\mu^k}{\mu^k + \kappa^k \sum_{j=1}^{i-1} g^k(t_i - t_j)},$$

which allows construction of the probability matrix P^k where k is the iteration number. This is the expectation step as it involves evaluating the likelihood of the Hawkes intensity of the events at the current parameter estimates based on whether the event was caused by the background rate (the diagonal elements) or another event (the non-diagonal elements).

The maximisation step involves using these probabilities to generate new values of the unknown parameters. The background rate is updated by

$$\mu^{k+1} = \frac{\mathrm{tr}(P^k)}{T},$$

where T is the window of observation. To update the kernel, a form of $g(t)$ must be specified and in this example consider a simple exponential kernel $g(t) = \beta e^{-\beta t}$. The other two parameters of the Hawkes process can then be updated as

$$\kappa^{k+1} = \frac{\sum_{i>j} p_{ij}^k}{n},$$
$$\beta^{k+1} = \frac{\sum_{i>j} p_{ij}^k}{\sum_{i>j} (t_i - t_j) p_{ij}^k}.$$

These new values of the parameters are then used in a new expectation step to recalculate the P matrix. This process is repeated until the values of the parameters converge to final values and these converged values are the estimated values of the generating process. This method also produces a matrix that indicates the probability of each event's likely parent; either the background rate or another event. Veen and Schoenberg (2008) state that this expectation maximisation algorithm is a more robust method than maximum likelihood as maximum likelihood relies on asymptotic properties of the likelihood whereas expectation maximisation only relies on enough data to estimate the parent probability matrix. Therefore, in situations with limited data, the EM algorithm is preferred.

As outlined previously, both of these methods of maximum likelihood and expectation maximisation are frequentist. In this thesis the problem of inference will be tackled in a Bayesian manner.

2.3 Bayesian Statistics

A Bayesian approach is one that applies Bayes' rule to the inference problem. Bayes' rule can be written as

$$p(\theta \mid y) = \frac{p(y \mid \theta)p(\theta)}{p(y)},$$

where θ is the unknown parameter of the model and y is the observed data. The function $p(y \mid \theta)$ is the likelihood, $p(\theta)$ is the prior distribution and $p(y)$ is the marginal distribution.

For a given model $p(y)$ is a constant with respect to θ and therefore is common to see the posterior distribution written as

$$p(\theta \mid y) \propto p(y \mid \theta)p(\theta).$$

Therefore, with each model a likelihood must be constructed and a prior distribution for the unknown parameters θ must also be chosen. Eq. (2.4) is a general likelihood for a point process and thus combined with Eq. (2.2) for the Hawkes process intensity it is then just a case of using a suitable prior for

the unknown components. In practice, the choice of prior has computational consequences on the ease of calculating and sampling from $p(\theta \mid y)$.

An exact calculation of the posterior distribution is not always possible and only certain combinations of likelihoods and priors produce analytically tractable posterior distributions where direct samples of the posterior distribution can be taken. If the posterior is not tractable, alternative methods are needed to approximate the distribution. These methods include constructing Markov chains of the parameters that emulate the behaviour of the posterior distribution (Hastings, 1970) or optimising an approximation to the posterior distribution (Jordan et al., 1999). It is these samples of the posterior distribution that form the estimate of θ and in contrast to frequentist techniques the estimate is now a probability distribution rather than a single value. As such this probability distribution now provides the ability to propagate uncertainty from the parameters directly into the forecasting of new values from the model.

2.3.1 Bayesian Hierarchical Modelling

In some cases, the prior distributions $p(\theta)$ used for the unknown parameters are also unknown and must be inferred from the data. This process involves specifying another level to the model and introduces a hierarchy of estimation. A basic hierarchical model can be written as

$$\begin{aligned} y &\sim F(\theta), \\ \theta &\sim G(\phi), \\ \phi &\sim N(0, 1), \end{aligned}$$

where some data y is drawn from a distribution F parametrised by an unknown θ, G is the prior distribution parameterised by ϕ which is also unknown and is parameterised with a *hyper-prior*, in this case the standard normal distribution. The full posterior distribution can be written as

$$p(\theta, \phi \mid y) \propto p(y \mid \theta) p(\theta \mid \phi) p(\phi),$$

and can be sampled accordingly. Under this type of model, it is assumed that the values of y are exchangeable, that is that the ordering of the y values

provides no additional information.

Hierarchical Bayesian models provide a key benefit when the available data contains observations from different groups as a hierarchical model allows for the sharing of information between the different groups. The different groups are linked via the prior and hyper-prior distributions which allows for the pooling of information to aid the inferences, again assuming that the parameters of the groups are exchangeable between groups. This feature is even more pronounced when the number of observations between different groups varies widely as the groups with less observations benefit from the information in the larger groups. Throughout this thesis hierarchical models are used with the Hawkes process to allow for the sharing of data when inferring the unknown parameters.

2.3.2 Bayesian Model Assessment

Model assessment consists of deciding how well a constructed model reflect the data it is fitted on. As each model has a number of choices such as likelihood and prior distributions it is important to understand how changing these can effect the model.

In frequentist statistics, p-values and statistical tests can reflect the suitability of a model. However, these can be sensitive to model and data assumptions and lead to difficult interpretations. In contrast, Bayesian model assessment is more direct. It utilises the full posterior samples of the parameters to assess the viability of the model.

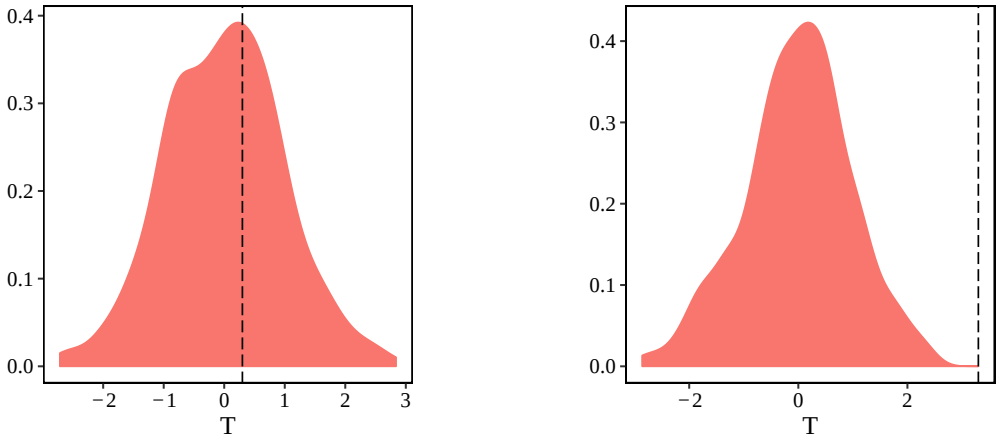
One method of Bayesian model checking is called 'posterior p-values' (Meng, 1994). This approach compares simulated data to the real data to assess whether the specified model is correct and suitable for the observed data.

In practise there exists some data y that has been generated from some unknown distribution. In building a model for the data, a distribution $F(y \mid \theta)$ is chosen for the likelihood. The posterior distribution of the unknown parameters θ is then inferred and sampled from which provides parameter samples

$\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(n)}$. For each posterior sample, a replicated data set can be generated $y^{\text{rep}} \sim F(\hat{\theta}^{(i)})$ leading to n different replications of the true data.

Graphical checks can now be made between the distribution of the generated data and the true data. Simply put, if the generated data does not resemble the real data then the model is unsuitable and a better choice of F is needed.

If a the distribution of the replicated data aligns with the true distribution then the next step can be taken for a posterior p-value. This involves calculating a test statistic $T^{(i)}$ for each of the replicated datasets (labelled by i) and a true statistic T^{true}. If the model is well suited T^{true} will be comparable to the values of $T^{(i)}$ and likewise a bad model will have a T^{true} substantially different from $T^{(i)}$.



(a) The density of $T^{(i)}$ compared to the true value T^{true} shown by the dashed line. In this case the true test statistic value falls inside the distribution of replicated test statistics.

(b) The choice of model does not fit the data well, hence the true value of the test statistic, the dashed line, is not comparable to the replicated values.

Figure 2.3: A graphical representation of posterior p-values.

This is demonstrated in Figure 2.3 where the densities in both Figures 2.3a and 2.3b show the distribution of $T^{(i)}$ which is calculated from data that is simulated from the posterior parameter samples. The dashed line in both

Figure 2.3a and 2.3b shows the true value of the test statistic, in Figure 2.3a the chosen model is correctly replicating the true data as the dashed line falls well inside the density and we would conclude that this is a suitable model. For Figure 2.3b the real test statistic is substantially different from the distribution of replicated values, therefore this model should be rejected.

Choices of the test statistic depend entirely on the problem at hand. It should be a property of the true data that would be difficult for the distribution F to imitate. For example, $T = \max(y)$ is useful for assessing the tails of the distribution. A F with thin tails would be unlikely to generate a large enough maximum value compared to the true data.

Throughout this thesis models will be assessed by simulating from the posterior distributions to check that the model is generating similar data. Good models will produce simulated data that replicates the real data.

Similarly, using unseen data (a test set) and calculating the predictive likelihood on the held out data in another way of assessing the validity of a model. The test set provides a sample of data that is different to the data that the model used to infer the parameters. Therefore, a model that overfits to the training data will fit the test set poorly whereas a model that captures the underlying behaviour of the data will produce a better predictive likelihood. In the Bayesian case, the predictive likelihood is simply the model likelihood $p(y \mid \theta)$ calculated across the parameter samples of θ.

Another method of assessing the viability of a model is the Deviance Information Criteria (DIC) which compares goodness of fit of a model to the number of parameters in the model. As a model increases in the number of parameters used, the fit is likely to improve but the chance of overfitting increases. The DIC metric is able to account for this and provides a balance between ensuring a reasonable fit and the number of parameters. The DIC is defined as

$$\begin{aligned} \text{DIC} &= -2 \log p(y \mid \hat{\theta}) + 2p_{\text{DIC}}, \\ p_{\text{DIC}} &= 2 \left(\log p(y \mid \hat{\theta}) - \mathbb{E}_{\text{posterior}} \log p(y \mid \theta) \right), \end{aligned}$$

where $\hat{\theta}$ is the Bayesian estimate of the parameter values, i.e. the mean of the posterior samples and $\mathbb{E}_{\text{posterior}} \log p(y \mid \theta)$ is the expected value of the likelihood over the posterior samples.

The DIC value can be used to compare different models. After calculating the value from the posterior samples of the parameters of each model, the model with the lowest DIC value is the preferred model.

The p_{DIC} value is an approximation to the effective number of parameters in the model (Spiegelhalter et al., 2002). As such, it is suited where the number of parameters in a model may not be well defined, such as a nonparametric model.

2.4 Nonparametric Statistics

In the above sections it is typically assumed that the likelihood used for the models in question has a functional form. For example, a simple toy problem might propose that some observations $y_i, i = 1, \ldots, n$ are from a normal distribution. In this case, the probability density function of the normal distribution would be used as the likelihood

$$p(y \mid \theta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y_i - \mu)^2\right)$$

where $\theta = \{\mu, \sigma^2\}$ are the mean and variance of the model. This model has two free parameters that are unknown and must be inferred. But what if this likelihood was not suitable for the data? Alternatively, the likelihood could be changed to a different distribution, one with more free parameters or different properties to the normal distribution. This could fit the data better, but ultimately relies on the observed data being well behaved and described by a distribution. In most scenarios, the data is likely to be too complicated to be well described by a distribution assumed by the parametric approach. Instead, the assumption of a tractable likelihood needs to be replaced with a more flexible approach. This introduces the concept of nonparametric statistics where the likelihood can no longer be described by a finite number of parameters and instead it is learnt from the data.

In the case of the Hawkes process, if the form of the component, such as the background rate or kernel, was specified beforehand it might miss certain features in the data or even be completely unsuited for the data resulting in an incorrect model. Instead by learning the form of the components from the data a more flexible model can be obtained.

There is a trade off between building simple, computationally efficient models and those that are more complex and require significantly longer periods to learn the parameters of the model. In the case of nonparametric modelling, the infinite number of parameters leads to an increased computational effort to obtain suitable inferences and therefore it must be considered whether the additional resources needed are worth the flexibility of the model. It is imperative to check that a nonparametric approach does offer something new compared to a simpler model.

There are numerous uses for nonparametric technique in point process modelling. In Weinberg et al. (2007) a nonparametric approach is used to model the arrival of calls to a call centre. By using a nonparametric smoothing method they are able to improve the predictive accuracy of the arrival of calls over traditional methods. Mohler (2013) use a spatial Hawkes process to analyse the clustering of residential burglaries in Los Angeles. They use a nonparametric extension of a Hawkes process to explore the impact a burglary has on the surrounding area over time. By using a nonparametric model they are able to capture complex details of the data that would not have been possible under a parametric framework. Adams et al. (2009) use a Gaussian process to model the intensity rate of a Poisson process. By using latent variables to represent possible 'thinned' events they infer the intensity based on the true observed events and these latent unobserved events. Such a method has an advantage that prior belief of the intensity can be specified but, computationally, it is a very expensive method, with an expected order of $o(n^3)$ for n points. So whilst it provides a Bayesian method of estimating a point process, it is limited in scale due to the computational costs.

These previous works each use a different approach in nonparametric modelling. In this thesis the Dirichlet process will the focus of the nonparametric models. In the next chapter a full treatment of the Dirichlet process will be given, both the mathematics needed to understand how it can be used as a Bayesian nonparametric prior and the computational methods for sampling from such a posterior distribution. The Dirichlet process will then be used to form nonparametric components of the Hawkes process which provides a full Bayesian inference approach thus can capturing and propagating parameter uncertainty.

Chapter 3

dirichletprocess: An R package for Fitting Complex Bayesian Nonparametric Models

The Hawkes process has three free components, the background rate $\mu(t)$, κ and the decay kernel $g(t)$ of which each can be specified to take a particular form and then must be inferred from the data. However it is also possible to take a nonparametric approach and learn the forms of these components from the data which can help prevent miss-specification. As such, when a nonparametric approach is taken, the components of the Hawkes process are built using Dirichlet process models and with each application in this thesis a different component of the Hawkes process is modelled nonparametrically.

This lead to the need for a flexible software package that could perform the appropriate inference and be reused with each experiment. No such software existed with this flexibility and thus the `dirichletprocess` `R` package was written. This chapter details the mathematical information needed to understand how a Dirichlet process can be used as a nonparametric model, the design choices that went into the package development and multiple examples of how the package can be used to perform a wide variety of inference tasks.

While frequentist nonparametrics has a long history, Bayesian nonparametrics was a relatively dormant field until the mid 1990s. Although much

of the theory of nonparametric priors had been worked out in previous decades (Ferguson, 1973), computational issues prevented widespread adoption. This changed with the development of posterior simulation methods such as Metropolis-Hastings (Hastings, 1970) and the Gibbs sampler (Geman and Geman, 1984) which were first applied to the task of nonparametric density estimation using Dirichlet process (DP) mixtures in a seminal paper by Escobar and West (1995). This kick-started research into Bayesian nonparametrics which has now become one of the most popular research areas in statistics and machine learning. While there are now several widely used models within the field of Bayesian nonparametrics including the Gaussian process, beta process and Polya trees, the Dirichlet process mixture model (DPMM) remains popular due to its wide applicability and elegant computational structure.

A Dirichlet process is defined by two parameters, α and G_0 and can be written as

$$G \sim \mathrm{DP}(\alpha, G_0), \tag{3.1}$$

where α is a concentration parameter and G_0 is the base measure of the Dirichlet process. The object G can now be used a prior distribution for the DPMM. At the most basic level, the DPMM can be viewed as an infinite dimensional mixture model which represents an unknown density $f(y)$ as:

$$f(y) = \int k(y \mid \theta) p(\theta \mid G) \mathrm{d}\theta,$$

where $k(\cdot \mid \theta)$ denotes the mixture kernel, and the mixing distribution G is assigned a nonparametric Dirichlet process prior as per Equation (3.1) with a base measure G_0 and concentration parameter α. In the most widely used DPMM, the mixture kernel is taken to be Gaussian so that $\theta = (\mu, \sigma^2)$ and $k(y \mid \theta) = N(y \mid \mu, \sigma^2)$ with a conjugate normal inverse-gamma specification for G_0. The infinite dimensional nature of such a model makes it capable of approximating any continuous distribution to an arbitrary degree of accuracy.

The use of DPMMs is not restricted to simply estimating the density of observed data. Instead, the DPMM can be used at any level in a hierarchical

model where it is considered necessary to represent a density nonparametrically due to a lack of knowledge about its parametric form. For example, consider a (multilevel) random effects model where there are J groups of observations, with observations in each group j following a Gaussian distribution with a group-specific mean μ_j and a common variance σ^2. To share information across groups, the means μ_j are assumed to be exchangeable and assigned a prior $p(\mu_j)$. If $y_{i,j}$ denotes the i^{th} observation in group j, then the model is:

$$y_{i,j} \sim N(y_{i,j} \mid \mu_j, \sigma^2),$$
$$\mu_j \sim p(\mu_j \mid \gamma),$$

where γ is the (hyper-)parameters of the prior. This model is an example of **partial pooling**, where the inference for each mean μ_j is based on the means of each of the other $J - 1$ groups, allowing information to be shared across groups. However, completing the model specification requires choosing a form for the prior distribution of group means $p(\mu_j \mid \gamma)$, which is made more difficult since the group means may not be observable with a high degree of accuracy, particularly when the number of observations is small. In this case, using a nonparametric DPMM specification for $p(\mu_j \mid \gamma)$ would avoid the risks of potentially specifying an inappropriate parametric form.

Typically when working with DPMMs, the posterior distributions are analytically intractable, so inference instead usually involves computational simulation. A variety of simulation algorithms based around Gibbs sampling and Metropolis-Hastings have been developed to draw samples from DPMM posterior distributions, which can then be used for inference. As such, the widespread adoption of DPMMs has to some extent been held back by the level of statistical and programming literacy required to implement them. The purpose of the `dirichletprocess` package is to provide a unified implementation of these simulation algorithms for a very wide class of DP mixture models which makes it easy to incorporate DPMMs into hierarchical models.

The design philosophy of the `dirichletprocess` package is the polar opposite of the existing `DPpackage R` package (Jara et al., 2011) which also

provides an implementation of DPMMs. The purpose of `DPpackage` is to give a fast implementation of several common tasks where DPs are used, such as density estimation using Gaussian mixtures, and a variety of specific regression models. While the `DPpackage` package is very useful in these situations, it cannot be used for any applications of DPs which do not fall into one of the pre-specified tasks incorporated in the package, or use mixture kernels or base measures other than those provided.

In contrast, the purpose of the `dirichletprocess` package is not to automate a pre-specified range of tasks but instead represent DPMMs as objects in `R` so that they can be used as building blocks inside user-specified hierarchical models. The target audience is users who are working with (possibly hierarchical) models which uses a DPMM at some stage, and who require a DPMM implementation which can be used as a part of a more general model estimation scheme. As such, the number of tasks which can be achieved using the `dirichletprocess` is quite large, although the trade-off is that the functions in this package will be slower than those in `DPpackage` when it comes to the specific models which it implements.

Key features of the `dirichletprocess` package include:

- An implementation of DP mixture models for various types of mixture kernel including the Gaussian, beta, multivariate normal and Weibull.
- Implementation of DP posterior sampling algorithms in both the conjugate and nonconjugate cases.
- A object-based interface which allows the user to work directly with DP objects in `R` so that they can be incorporated into hierarchical models.

The latter point should hopefully make the package especially flexible and useful.

For the user already well versed in the mathematics behind Dirichlet processes, Section 3.3 can be skipped as Section 3.4 shows how the package can be used in `R` without any knowledge of the underlying algorithms.

3.1 A Technical Note

The ultimate purpose of this package is to represent Dirichlet process mixture models as objects in R, so that they can be manipulated and used as building blocks. At the time of writing, R currently features three separate object systems (S3, S4 and RC) designed to allow object-orientated programming. This package uses S3. There are two motivations for this design choice which outweigh any advantages that come from using any of other of the R object systems.

1. Speed. While R is an excellent programming language which makes carrying out high level statistical analysis easy, its slow speed remains a bottleneck particularly in tasks such as Gibbs Sampling and Monte Carlo Markov Chain (MCMC) sampling which are inherently sequential and cannot be vectorised. While the base R system is already slow, the S4 and Reference Class (RC) object system suffer from further performance hits since they must search for the correct method for each function evaluation (Wickham, 2014). S3 suffers a similar slowdown but to a lesser extent. The price paid in speed is recovered in code comprehension and ease of development.

2. Ease-of-use. A key feature of this package is that users can themselves specify new DP mixture types if the package does not implement the precise specification they desire. The object systems S4 and RC are geared towards intermediary/advanced R programmers and can be intimidating to novices. The design philosophy of this package allows users to override the behaviour of DP objects and create new mixture types without needing to learn the intricacies of any particular object system. The chosen representation where DP objects are simple S3 structures does not require the user to learn anything about the more obscure intricacies of R objects, and instead they can focus purely on writing the R functions to implement the DP models.

Both of these technical features are demonstrated in Section 3.5.2 two new Dirichlet process models are constructed using just the building blocks that this package provides to highlight the ease of use of the S3 object system and subsequent trade-off in performance.

Current alternatives for nonparametric inference include Stan (Carpenter et al., 2016), PyMC3 (Salvatier et al., 2016) and Edward (Tran et al., 2016). However, whilst all three packages are much more general than the `dirichletprocess` offerings, they do not offer ease of customisation that `dirichletprocess` does. Firstly, Stan does not allow discrete parameters in models. As Dirichlet process models require cluster labels which are inherently discrete parameters this immediately rules out a direct translation of Dirichlet process specifications to Stan code. There are certain 'get arounds' that can factor out discrete parameters in Stan, but a Dirichlet process cannot take advantage of these as there are a potentially infinite number of parameters that cannot be established before the code is compiled. For both the Python libraries Edward and PyMC3, examples exist of building Dirichlet process models in the respective framework. However, these are built on top of TensorFlow and Theano, therefore, being able to build Dirichlet process objects into statistical workflows would require learning these external libraries. Instead our package `dirichletprocess` is written natively in R and abstracts the difficulties away, allowing users to write Dirichlet process models in R code and not worry about computational details.

3.2 Literature Review

For Bayesian computations, the most entrenched software programs are "Bayesian inference Using Gibbs Sampling" (BUGS) (Lunn et al., 2009) and "Just Another Gibbs Samples" (JAGS) (Plummer, 2003). Both programs offer Gibbs sampling for parameter inference in a wide variety of user specified models. However, both of these software packages are being usurped recently by Stan, first released in 2012. It is a probabilistic programming language imple-

menting more modern sampling methods such as; no-U-Turn sampler (NUTS), Hamiltonian Monte Carlo (HMC) and variational inference (Carpenter et al., 2016).

All three programs follow similar structure for performing Bayesian inference. A model with unknown parameters is declared as a likelihood function and each unknown parameter is given a suitable prior. The model is then sampled to produce posterior samples of the parameters which the user can use to perform the appropriate analysis with such samples. The software packages differ in how they arrive at the posterior samples but fundamentally all provide samples of the given posterior distribution.

The above three programs are complete and contained software packages. No other programming language is needed to use them and sample from the models [1]. In the Python ecosystem, there has been an explosion of tools focused on machine learning which can be used for Bayesian inference.

The most similar to BUGS/JAGS/Stan is PyMC3, another probabilistic programming language built on-top of Python (Salvatier et al., 2016). Again, models and priors are declared before samples from the posterior distribution are taken using the NUTS algorithm. The model and priors are written using the Python language and such models can easily be dropped into other Python programs. This allows for seamless integration and a full Bayesian analysis can be undertaken in Python without the need of another language.

A more machine learning focused approach is taken by the Python package Edward (Tran et al., 2016). Again, code descirbing the model is written in Python using Edward specific syntax before being sampled using a specified algorithm. Using integration with TensorFlow (Abadi et al., 2016), Google's machine learning platform, more advanced models can be built; such as the inclusion of neural networks, generative adversarial networks and variational auto-encoders. Furthermore, machine learning training techniques such as mini-batches are available in Edward. Overall, Edward bridges the gap be-

[1] Interfaces do exist for JAGS in R with `rjags` and Stan has `rstan`.

tween Bayesian inference and current machine learning trends.

So whilst the above tools have the ability to implement Dirichlet process models, it will involve learning package specific syntax and furthermore, understand the mathematics behind how a Dirichlet process can be represented and sampled. With the `dirichletprocess` package, this detail is removed from the user and instead, they can work on what they know - writing a statistical analysis and using the Dirichlet process objects where necessary.

3.3 Background Information

This section provides background information about the Dirichlet process and includes the key mathematical properties around which the sampling algorithms in the `dirichletprocess` package are based. The details on how to use the package are discussed in Section 3.4.

It is commonly required to learn the unknown probability distribution F which represents the distribution of the observed data $y_1, \dots, y_n$. In parametric Bayesian inference, F is assumed to belong to a known family of distributions (such as the normal or exponential) with an associated parameter vector θ which is of finite length. The parameters θ must be estimated using the available data $y_1, \dots, y_n$. This leads to the model

$$y_i \sim F(y_i \mid \theta),$$
$$\theta \sim p(\theta \mid \gamma),$$

where $p(\theta \mid \gamma)$ denotes the prior distribution and γ are the prior parameters. The task of inference then involves finding an appropriate value for θ, which is equivalent to choosing which member of the specified family of distributions gives the best fit to the data.

However, in practise it may not be clear how to choose an appropriate parametric family of distributions for F. If the wrong family is chosen, then conclusions based on the estimated model may be highly misleading. For example, if it is assumed that F has a normal distribution with unknown parameters $\theta = (\mu, \sigma^2)$ when in fact the true F is heavy-tailed, this can lead to

severe underestimation of the probability of extreme events occurring (Coles, 2001).

This problem can be avoided by using a nonparametric prior specification which puts positive prior mass on the whole space of probability densities rather than on a subspace spanned by the finite-length parameter vector θ. This allows the estimated F to adapt to the data, rather than being restricted to a particular family of distributions such as the normal or exponential. The Dirichlet process (DP) is one of the most widely used Bayesian nonparametric priors, due to its flexibility and computational simplicity. The aim of this section is not to give a full treatment of the Dirichlet processes and a reader unfamiliar with them should refer to a standard reference such as Antoniak (1974). Instead the properties of the DP that are directly relevant to their implementation in the `dirichletprocess` package will be explained.

The basic DP model has the form:

$$\begin{aligned} y_i &\sim F, \\ F &\sim \text{DP}(\alpha, G_0), \end{aligned}$$

where G_0 is known as the **base measure** and encapsulates any prior knowledge that might be known about F. Specifically, it can be shown that $\mathbb{E}[F \mid G_0, \alpha] = G_0$. The concentration parameter α specifies the prior variance and controls the relative contribution that the prior and data make to the posterior, as the following result shows.

Key Property 1: The DP is a conjugate prior in the following sense: if $y_1, \ldots, y_n \sim F$ and $F \sim \text{DP}(\alpha, G_0)$, then:

$$F \mid y_1, \ldots, y_n \sim \text{DP}\left(\alpha + n, \frac{\alpha G_0 + \sum_{i=1}^{n} \delta_{y_i}}{\alpha + n}\right),$$

where δ_{y_i} denotes a point-mass at y_i. In other words, the posterior distribution of F is a weighted sum of the base measure G_0 and the empirical distribution of the data, with the weighting controlled by α.

The DP is a prior distribution over the space of probability distributions. As such, samples from a DP are probability distributions. The stick-breaking representation first introduced by Sethuraman (1994) shows what such samples look like.

Key Property 2: Suppose that $F \sim \text{DP}(\alpha, G_0)$ is a random probability distribution sampled from a DP prior. Then with probability 1, F can be written as:

$$F = \sum_{k=1}^{\infty} w_k \delta_{\phi_k}, \quad \phi_k \sim G_0$$

where

$$w_k = z_k \prod_{i=1}^{k-1} (1 - z_i), \quad z_i \sim \text{Beta}(1, \alpha).$$

In other words, random probability distributions can be sampled from a DP by first drawing a collection of samples z_i from a beta distribution, transforming these to produce the weights w_k, and then drawing the associated atoms from G_0. Note that in order for F to be a true from a DP, an infinite number of such weights and atoms must be drawn. However in practice, the above summation can be truncated with only a finite number N of draws, while still providing a very good approximation.

By combining Key Properties 1 and 2, a DP posterior distribution, $F \mid y_1, \ldots, y_n$, can be sampled from as follows:

Key Property 3: If $y_1, \ldots, y_n \sim F$ and $F \sim \text{DP}(\alpha, G_0)$ then a sample of the posterior distribution $F \mid y_1, \ldots, y_n$ can be taken as follows:

$$F = \sum_{k=1}^{N} w_k \delta_{\phi_k}, \quad \phi_k \sim \frac{\alpha G_0 + \sum_{i=1}^{n} \delta_{y_i}}{\alpha + n},$$

where

$$w_k = z_k \prod_{i=1}^{k-1} (1 - z_i), \quad z_i \sim \text{Beta}(1, \alpha + n).$$

which provides the basis of how a Dirichet process can be sampled given some observed data.

3.3.1 Dirichlet Process Mixtures

The stick-breaking representation in Key Property 2 above shows that probability distributions sampled from a DP are discrete with probability 1. Therefore, the DP is not an appropriate prior for F when F is continuous. As such, it is usual to adopt the following mixture specification instead, which will be called the Dirichlet process mixture model (DPMM):

$$\begin{aligned} y_i &\sim k(y_i \mid \theta_i), \\ \theta_i &\sim F, \\ F &\sim \mathrm{DP}(\alpha, G_0). \end{aligned} \tag{3.2}$$

In other words, F has a DP prior as before, but rather than the data y_i being drawn from F, it is instead the mixture parameters θ which are drawn from F. These θ values then act as the parameters of a parametric kernel function $k(\cdot)$, which is usually continuous. The most commonly used example is the Gaussian mixture model where $\theta_i = (\mu_i, \sigma_i^2)$ so that $k(y_i \mid \theta_i) = N(y_i \mid \mu_i, \sigma_i^2)$.

The key point here is that since F is discrete, two independent draws θ_i and θ_j from F can have identical values with a non-zero probability. As such, the DPMM can be seen as sorting the data into clusters, corresponding to the mixture components. The above model can hence be written equivalently as the following mixture model, which is infinite dimensional and can be viewed as a generalisation of the finite mixture models commonly used in nonparametric statistics:

$$\begin{aligned} y_i &\sim G, \\ G &= \int k(y_i \mid \theta) F(\theta) \mathrm{d}\theta, \\ F &\sim \mathrm{DP}(\alpha, G_0). \end{aligned} \tag{3.3}$$

When the DPMM is used in practice there are different use cases for the posterior distribution. In some cases, the primary object of interest will be the θ_i parameters from Equation (3.2) which are associated with the $y_1, \ldots, y_n$ observations. This is particularly the case in clustering applications, where the goal is to assign similar observations to the same cluster (i.e. to identical values of

θ). However in other situations it will be the distribution F which is of primary interest, with the θ_i parameters integrated out. The `dirichletprocess` package returns posterior samples of all these quantities, so that the user can decide which are most relevant.

Posterior inference in the `dirichletprocess` package is based around the Chinese Restaurant Process (CRP) sampler (Neal, 2000). This is a Gibbs-style algorithm based on the DPMM representation in Equation (3.2) above, and draws samples of $\theta_1, \ldots, \theta_n$ from their posterior with the distribution F integrated out.

Key Property 4: Let θ_{-i} denote the set of θ values with θ_i excluded, i.e. $\theta_{-i} = (\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_n)$. Then the posterior distribution for θ_i conditional on the other model parameters is:

$$
\begin{aligned}
p(\theta_i \mid \theta_{-i}, y_{1:n}, \alpha, G_0) &= \sum_{j \neq i} q_{i,j} \delta(\theta_j) + r_i H_i, \\
q_{i,j} &= b k(y_i, \theta_j), \\
r_i &= b\alpha \int k(y_i, \theta) \mathrm{d}G_0(\theta),
\end{aligned}
$$

where b is set such that $\sum_{j \neq i} q_{i,j} + r_i = 1$ and H_i is the posterior distribution of θ_i using the prior base measure G_0.

Based on this result, Gibbs sampling is used to repeatedly draw each value of θ_i in turn from its posterior distribution, with all other variables held constant. An important distinction needs to be made between the **conjugate** case where the G_0 base measure is the conjugate prior for θ with respect to the kernel $k(\cdot)$, and the **nonconjugate** case where there is not a closed form for the posterior distribution. In the conjugate case, the integral in Key Property 4 can be computed analytically and the resulting distribution is simply the predictive distribution. In this case, the θ_i values can be sampled directly from their true posterior distribution.

In the nonconjugate case things are slightly more difficult and the integral in Key Property 4 cannot be evaluated. As such, numerical techniques

must be used instead, which will typically result in slower computation. The `dirichletprocess` package handles the nonconjugate case by using Algorithm 8 from (Neal, 2000), which is one of the most widely used techniques for performing this sampling.

In both the conjugate and nonconjugate cases the Gibbs sampling is conceptually similar with the new values of θ_i being proposed sequentially from their respective posterior distributions. However in practice, this can result in poor mixing of the samples. Poor mixing results from the sampling procedure being trapped in a local maximum of the posterior distribution, resulting in the procedure struggling to explore the full posterior distribution. As new values of θ are only proposed from the G_0 distribution this can result in slow convergence. One approach to speed up convergence is to add in additional sample of the θ_i values at the end of the cluster label sampling. For each cluster and its associated data points the cluster parameter is updated using the posterior distribution

$$p(\theta_i \mid y_j) = \prod_{j=i} k(y_j \mid \theta_i) G_0, \tag{3.4}$$

for a conjugate base measure, this posterior distribution is tractable and thus can be sampled directly. For a nonconjugate G_0, a posterior sample is achieved using the Metropolis-Hastings algorithm (Hastings, 1970). This results in a better exploration of the posterior distributions and thus helps the mixing of the samples.

For simple density estimation and non-hierarchical predictive tasks, having a posterior sample of $\theta_{1:n}$ will be sufficient for inference, and the distribution F is not of any intrinsic interest. However when the DP is used as part of a hierarchical model it is also necessary to have samples from the posterior distribution of F. These can be obtained using the following property:

Key Property 5: Given the model from Eq. (3.2) let $\theta_1, \ldots, \theta_n$ be a sample from the posterior $p(\theta_{1:n} \mid y_{1:n}, \alpha, G_0)$ drawn using the CRP sampler. Then, $p(F \mid \theta_{1:n}, y_{1:n}, \alpha, G_0) = p(F \mid \theta_{1:n}, \alpha, G_0)$ is conditionally independent of $y_{1:n}$. As such, $\theta_{1:n}$ can be considered as an i.i.d sample from F, and so F can be sampled from its posterior distribution using Key Property 3 above:

$$F = \sum_{i=1}^{N} w_i \delta_{\theta_i}, \quad w_i \sim \text{Beta}(\alpha + n, 1), \quad \theta_i \sim G_0 + \sum_{i=1}^{n} \delta_{\theta_i}$$

3.3.2 Hyperparameter Inference

In the above discussion, it has been assumed that the concentration parameter α and base measure G_0 were constant. However in practice, better results can often be obtained if they are also learned from the data.

Inferring the Concentration Parameter

Following West (1992) a prior of Gamma(a, b) is used for α. The corresponding posterior distribution depends only on the number of unique values of $\theta_{1:n}$. More specifically, given the model in Equation (3.2) let $\theta_1, \ldots, \theta_n$ denote a sample from the posterior $p(\theta_{1:n} \mid y_{1:n}, \alpha, G_0)$. Suppose that there are k unique values in this sample. Then a sample from $p(\alpha \mid \theta_{1:n}, y_{1:n}, G_0)$ can be obtained as follows:

- Simulate a random number z from a Beta$(\alpha + 1, n)$ distribution
- Define $\tilde{\pi}_1 = a + k + 1$ and $\tilde{\pi}_2 = n(b - \log(z))$, then define $\pi = \tilde{\pi}_1 / (\tilde{\pi}_1 + \tilde{\pi}_2) \in [0, 1]$
- With probability π, draw a value of α from a Gamma$(a + k, b - \log(z))$ distribution, and with probability $(1 - \pi)$ draw it from a Gamma$(a + k - 1, b - \log(z))$ distribution instead.

When fitting a DP the value of α can be easily inferred and sampled by default in the `dirichletprocess` package.

Inferring the Base Measure

The base measure G_0 can be parameterised with values γ which themselves are also random and from some distribution $p(\gamma)$. By placing a prior distribution on the parameters of the base measure this allows for the DP to adapt to the data and ensure that the fitting algorithms converge to the stationary distributions quicker

$$\begin{aligned} \theta_i \mid \gamma &\sim G_0, \\ \gamma &\sim p(\gamma), \\ \gamma \mid \theta_i &\sim H, \end{aligned}$$

where H is the posterior distribution. If $p(\gamma)$ is chosen such that it is conjugate to the posterior, the posterior distribution can be directly sampled, otherwise a Metropolis-Hastings step is include in the computation.

3.3.3 Implemented Mixture Models

One of the strengths of the `dirichletprocess` package is that it allows users to specify DPMMs using whichever choices of the kernel k and base measure G_0 they please. However for ease of use, certain choices of k and G_0 have been implemented directly in the package.

Gaussian Mixture Model

The Gaussian distribution is the most commonly used mixture model. In this case, $\theta = (\mu, \sigma^2)$ for the mean and variance. The kernel is:

$$k(y_i \mid \theta) = N(y_i \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(y_i - \mu)^2}{2\sigma^2}\right).$$

The conjugate prior for θ is the normal-inverse-gamma distribution, with parameters $\gamma = (\mu_0, k_0, \alpha_0, \beta_0)$

$$G_0(\theta \mid \gamma) = N\left(\mu \mid \mu_0, \frac{\sigma^2}{k_0}\right) \text{Inv-Gamma}\left(\sigma^2 \mid \alpha_0, \beta_0\right).$$

the default setting of the parameters is $\mu_0 = 0, k_0 = 1, \alpha_0 = 1, \beta_0 = 1$. It is recommended to rescale the data y such that its mean is 0 and standard deviation is 1 as this leads to the default parameterisation of G_0 being uninformative.

Since this prior is conjugate, the predictive distribution for a new observation $\tilde{y}$ can be found analytically, and is a location/scale Student-t distribution:

$$p(\tilde{y} \mid \gamma) = \int k(\tilde{y} \mid \theta) p(\theta \mid G_0) \mathrm{d}\theta = \frac{1}{\tilde{\sigma}} \text{Student-t}\left(\frac{\tilde{y} - \tilde{\mu}}{\tilde{\sigma}} \mid \tilde{v}\right),$$

where $\tilde{v} = 2\alpha_0, \quad \tilde{\mu} = \mu_0, \quad \tilde{\sigma} = \sqrt{\frac{\beta_0(k_0+1)}{\alpha_0 k_0}}$.

Finally the posterior distribution is also a normal-inverse-gamma distribution due to the conjugacy of the prior

$$\begin{aligned} p(\theta \mid y, \gamma) &= N\left(\mu \mid \mu_n, \frac{\sigma^2}{k_0 + n}\right) \text{Inv-Gamma}(\sigma^2 \mid \alpha_n, \beta_n), \\ \mu_n &= \frac{\kappa_0 \mu_0 + n\overline{y}}{k_0 + n}, \\ \alpha_n &= \alpha_0 + \frac{n}{2}, \\ \beta_n &= \beta_0 + \frac{1}{2}\sum_{i=1}^{n}(y_i - \overline{y})^2 + \frac{\kappa_0 n(\overline{y} - \mu_0)^2}{2(\kappa_0 + n)}. \end{aligned}$$

Multivariate Gaussian Mixture Model - Conjugate

The multivariate Gaussian mixture model is the most widely used nonparametric modelling approach for multivariate data and is heavily used in clustering applications (Maceachern and Müller, 1998). The unknown parameters are $\theta = (\boldsymbol{\mu}, \Lambda)$ and for d dimensional data $\boldsymbol{\mu}$ is a column vector of length d and Λ is a $d \times d$ dimensional matrix

$$k(y_i \mid \theta) = \frac{\mid \Lambda \mid^{\frac{1}{2}}}{2\pi^{-\frac{d}{2}}} \exp\left(-\frac{1}{2}(y_i - \boldsymbol{\mu})^\top \Lambda (y_i - \boldsymbol{\mu})\right).$$

For the prior choice, a multivariate normal distribution for $\boldsymbol{\mu}$ and Wishart distribution for Λ are used

$$G_0(\boldsymbol{\mu}, \Lambda \mid \boldsymbol{\mu_0}, \kappa_0, \nu_0, T_0) = N(\boldsymbol{\mu} \mid \boldsymbol{\mu_0}, (\kappa_0 \Lambda)^{-1}) \text{Wi}_{\nu_0}(\Lambda \mid T_0),$$

where $\boldsymbol{\mu_0}$ is the mean vector of the prior, κ_0, ν_0 are single values and T is a matrix. The default prior parameters are set as $\boldsymbol{\mu} = \mathbf{0}, T = \mathbf{I}, \kappa_0 = d, \nu_0 = d$.

This prior choice is conjugate to the posterior, therefore the posterior

distribution can be expressed analytically

$$\begin{aligned}
p(\theta \mid y_i) &= N(\boldsymbol{\mu} \mid \boldsymbol{\mu_n}, (\kappa_n \Lambda_n)^{-1}) \mathrm{Wi}(\Lambda \mid \nu_n, T_n), \\
\boldsymbol{\mu_n} &= \frac{\kappa_0 \mu_0 + n\overline{\mathbf{y}}}{k+n}, \\
\kappa_n &= \kappa_0 + n, \\
\nu_n &= \nu_0 + n, \\
T_n &= T_0 + \sum_{i=1}^{n} (y_i - \overline{\mathbf{y}})(y_i - \overline{\mathbf{y}})^\top + \frac{\kappa_0 n}{\kappa_0 + n} (\mu_0 - \overline{\mathbf{y}})(\mu_0 - \overline{\mathbf{y}})^\top.
\end{aligned}$$

Again, as this is a conjugate mixture the predictive function for some new data $\hat{y}$ can be written explicitly as

$$p(\hat{y} \mid \mathbf{y}) = \frac{1}{\pi^{\frac{nd}{2}}} \frac{\Gamma_d(\frac{\nu_n}{2})}{\Gamma_d(\frac{\nu_0}{2})} \frac{\mid T_0 \mid^{\frac{\nu_0}{2}}}{\mid T_n \mid^{\frac{\nu_n}{2}}} \left(\frac{\kappa_0}{\kappa_n}\right)^{\frac{d}{2}}.$$

Multivariate Gaussian Mixture Model - Semi-Conjugate

In the semi-conjugate case, the base measures for each parameter are specified independently

$$G_0(\boldsymbol{\mu}, \Sigma) = N(\boldsymbol{\mu} \mid \boldsymbol{\mu_0}, \Sigma_0) \mathrm{Wi}_{\nu_0}^{-1}(\boldsymbol{\Phi_0}).$$

Therefore, sampling from the posterior is achieved by firstly sampling a new Σ and then a new $\boldsymbol{\mu}$

$$\begin{aligned}
\Sigma \mid \boldsymbol{\mu}, \nu_0, \Phi_0 &\sim \mathrm{Wi}_{\nu_n}^{-1}(\Phi_n), \\
\nu_n &= \nu_0 + n, \\
\Phi_n &= \Phi_0 + \sum_{i=1}^{n} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T, \\
\boldsymbol{\mu} \mid \Sigma, \boldsymbol{\mu_0}, \Sigma_0 &\sim N(\boldsymbol{\mu_n}, \Sigma_n), \\
\Sigma_n &= \left(\Sigma_0^{-1} + n\Sigma^{-1}\right)^{-1}, \\
\boldsymbol{\mu_n} &= \Sigma_n \left(\Sigma_0^{-1} \boldsymbol{\mu_0} + n\Sigma^{-1} \overline{\boldsymbol{y}}, \right)
\end{aligned}$$

using the conditional probabilities, each parameter can be sampled using the previous sample. This allows us to use Algorithm 8 and treat the model as a nonconjugate mixture model.

Beta Mixture Model

Dirichlet process mixtures of beta distributions have been considered by Kottas (2006a) for the nonparametric estimation of continuous distributions that are defined on a bounded interval, $[0, T]$. For ease of interpretation, the beta distribution is parameterised in terms of its mean and standard deviation, in this case, $\theta = (\mu, \nu)$ with a known parameter T. The mixture kernel is:

$$k(y_i \mid \theta) = \text{Beta}(y_i \mid \mu, \nu, T) = \frac{y_i^{\frac{\mu\nu}{T}-1}(T - y_i)^{\nu(1-\frac{\mu}{T})-1}}{\text{B}(\frac{\mu\nu}{T}, \nu(1-\frac{\mu}{T}))T^{\nu-1}}.$$

There is no conjugate prior for the mixture kernel. Instead, the `dirichletprocess` package uses the (nonconjugate) prior from Kottas (2006a) where

$$G_0(\mu, \nu \mid T, \alpha_0, \beta_0) = U(\mu \mid [0, T])\text{Inv-Gamma}(\nu \mid \alpha_0, \beta_0).$$

the parameters $\alpha_0 = 2, \beta_0 = 8$ as set the default. The Metropolis-Hastings algorithm is used to sample from the posterior distribution. However, there is also the ability to place a prior distribution on β_0 and update the prior parameter with each iteration. For this a default prior distribution of

$$\beta_0 \sim \text{Gamma}(a, b),$$

with $a = 1, b = 0.125$ is used.

Weibull Mixture Model

The Weibull distribution has strictly positive support and is mainly used for positive only data modelling. Furthermore, it is ubiquitously used in survival analysis. Mixture of Weibull distributions have been considered by Kottas (2006a) for a variety of survival applications. The parameters of the Weibull distribution are the shape a and scale b

$$k(y_i \mid \theta) = \text{Weibull}(y_i \mid a, b) = \frac{a}{b} y_i^{a-1} \exp\left(-\frac{y_i^a}{b}\right),$$

where $\theta = (a, b)$.

A nonconjugate uniform-inverse-gamma distribution for the unknown parameters is used

$$G_0(a, b \mid \phi, \alpha, \beta) = U(a \mid 0, \phi)\text{Inv-Gamma}(b \mid \alpha, \beta),$$

by default ϕ, α and β do not have assigned values. Instead, priors are placed on ϕ and β and updated with each fitting procedure, α remains fixed. For ϕ a Pareto prior distribution is used as this is conjugate to the uniform distribution

$$\begin{aligned} a_i &\sim U(0, \phi), \\ \phi &\sim \text{Pareto}(x_m, k), \\ \phi \mid a_i &\sim \text{Pareto}(\max\{a_i, x_m\}, k + n), \end{aligned}$$

by default $x_m = 6, k = 2$ which is an infinite variance prior distribution.

As b is from an inverse-gamma distribution with fixed shape α it has a conjugate prior which is the gamma distribution.

$$\begin{aligned} b_i &\sim \text{Inv-Gamma}(\alpha, \beta), \\ \beta &\sim \text{Gamma}(\alpha_0, \beta_0), \\ \beta \mid b &\sim \text{Gamma}\left(\alpha_0 + n\alpha, \beta_0 + \sum_{i=1}^{n} \frac{1}{b_i}\right), \end{aligned}$$

with α fixed by the user and $\alpha_0 = 1, \beta_0 = 0.5$ by default. This prior on β with $\alpha_0 = 1$ is a conventional distribution with mean $\frac{1}{\beta_0}$ which allows for the user to decide how disperse the prior needs to be. As this is a nonconjugate model a Metropolis-Hastings step is needed to sample from the posterior distribution.

3.4 Package Overview

The `dirichletprocess` package contains implementations of a variety of Dirichlet process mixture models for nonparametric Bayesian analysis. Unlike several other `R` packages, the emphasis is less on providing a set of functions which completely automate routine tasks (e.g. density estimation or linear regression) and more on providing an abstract data type representation

of Dirichlet process objects which allow them to be used as building blocks within hierarchical models.

To illustrate how the package is meant to be used and how it differs from other `R` packages, consider the task of density estimation. Suppose the density of some data stored in the variable `y` needs to be estimated using a Dirichlet process mixture of Gaussian distributions. This is done as follows:

```
y ← rt(200, 3) + 2 #generate sample data
dp ← DirichletProcessGaussian(y) #create the object
dp ← Fit(dp, 1000, progressBar = FALSE) #fit the object
```

The function `DirichletProcessGaussian` is the creator function for a mixture model of univariate Gaussians and this creates the object `dp`. The function `Fit` is used on this object to infer the cluster parameters, which uses the Chinese Restaurant Sample algorithm described in Section 3.3.1. With each iteration, the assigned cluster label to each datapoint is updated, then the resulting cluster parameters are updated before finally updating the concentration parameter α. Using the `Fit` function the details of the sampling are removed from the user and this provides an 'out-of-the-box' method to easily fit a Dirichlet process to data. Only a specification of the type of mixture model is needed - in this case a Gaussian mixture. The returned object `dp` from the `Fit` function contains the following information: `dp$clusterParameterChains` stores the MCMC samples of the cluster parameters, `dp$weightsChain` stores the associate weights and `dp$alphaChain` stores the samples of the concentration parameter α. These posterior samples can then be used for inference based on what the user is trying to accomplish.

The `dirichletprocess` package currently provides the following features:

- Implementations of Dirichlet process mixture models using Gaussian, beta, and Weibull mixture kernels.
- Implementation of various schemes for resampling model parameters.
- Access to samples from the Dirichlet process in both marginal form, as

well as in (truncated) stick-breaking form

- A flexible way for the user to add new Dirichlet process models which are not currently implemented in the package, and yet still use the resampling functions from the package. To illustrate this, Section 3.5.2 shows how simple it is to create a Dirichlet process mixture model with Poisson and gamma distribution kernels, even though this is not implemented in the package.

- An ability to plot the likelihood, posterior and credible intervals of a Dirichlet process using `plot`.

All of the above features will be demonstrated in the following examples.

Nonparametric Density Estimation

The most simple application of DPMMs is to nonparametrically estimate the distribution of independent and identically distributed observations $y_1, \ldots, y_n$, where:

$$\begin{aligned} y_i &\sim F, \\ F &= \sum_{i=1}^{n} \pi_i k(y_i \mid \theta_i), \end{aligned}$$

where k is some density function parameterised by θ_i and n is some unknown amount of clusters (i.e. F has been specifically nonparametrically as a mixture model). The most widely used specification is the Gaussian mixture kernel with a normal-inverse-gamma base measure G_0, which is described more fully in Section 3.3.3.

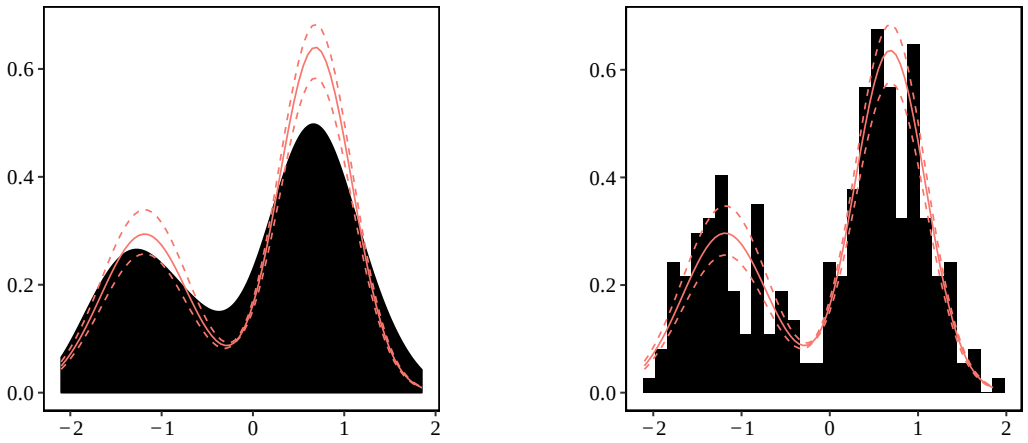
The waiting times between eruptions of the Old Faithful geyser are used as an example. This dataset is available within `R` and called `faithful`. The waiting times are transformed to be zero mean and unit standard deviation before a DPMM with the default settings is fitted. This models the waiting

times as a mixture of normal distributions and can be written as

$$
\begin{aligned}
y_i &\sim F, \\
F &= \sum_{i=1}^{n} \pi_i k(y \mid \theta_i), \quad \theta_i = \left\{\mu_i, \sigma_i^2\right\}, \\
\theta_i &\sim G, \\
G &\sim \mathrm{DP}(\alpha, G_0),
\end{aligned}
$$

where $k(y \mid \theta)$ is the standard normal probability density and G_0 is the base measure as in Section 3.3.3.

```
faithfulTransformed ← scale(faithful$waiting)
dp ← DirichletProcessGaussian(faithfulTransformed)
dp ← Fit(dp, 500)
plot(dp)
```



(a) The estimated density of the data is plotted with the DPMM posterior mean and credible intervals overlaid in red.

(b) Instead of a density estimate, a histogram is plotted for the data.

Figure 3.1: Old Faithful waiting times density estimation with a DPMM of Gaussians.

The resulting posterior distribution from the model is show in Figure 3.1. Both peaks in the data have been accurately modelled, something that a single

normal distribution would not be able to describe.

For most users the `Fit` function is sufficient for practical purposes. However, for the more advanced users who wish to alter how they fit the `dirichletprocess` object there are a number of functions available to help.

By default, the `Fit` function updates the cluster allocation, cluster parameters and the α parameter. In some rare cases, updating α every iteration can delay convergence and instead, the user could modify the function to update α every 10 iterations.

```
dp ← DirichletProcessGaussian(y)
samples ← list()
for(s in seq_len(1000)){
  dp ← ClusterComponentUpdate(dp)
  dp ← ClusterParameterUpdate(dp)

  if(s %% 10 == 0) {
        dp ← UpdateAlpha(dp)
  }
  samples[[s]] ← list()
  samples[[s]]$phi ← dp$clusterParameters
  samples[[s]]$weights ← dp$weights
}
```

The function `ClusterComponentUpdate` iterates through all the data points, y_i for $i = 1, \ldots, n$, updating its cluster assignment sequentially using Key Property 4 in Section 3.3.1. For each data point, it can either be assigned to an existing cluster, or form a new cluster. The probability that the data point is assigned to an existing cluster is proportional to $n_i k(y_j \mid \theta_i)$, where n_i is the number of points already assigned to the cluster θ_i and k is the likelihood of the data point evaluated with the cluster parameter θ_i. The probability that the datapoint forms a new cluster is proportional to α, the concentration parameter. If the datapoint is selected to form a new cluster, then this new cluster parameter θ_{new} is drawn from G_0 and added to the cluster

pool. Subsequent points can now also be added to this cluster.

Once each data point has sample a new cluster allocation the function `ClusterParameterUpdate` is called and updates each of the unique θ_j parameters. The new values of θ_j are sampled from the posterior distribution of the parameter using all the data associated to that cluster parameter as per Equation (3.4).

Finally, `UpdateAlpha` samples a new value of α from its posterior distribution using the method outlined in Section 3.3.2. By manually calling these functions the user has control over the MCMC routine without having to have specific knowledge of the required algorithms.

The key point of the `dirichletprocess` package which the above code highlights is that a) the user controls when to resample the DP parameters, and b) the current sample is contained in the DP object and ready for inspection at any point in the code. This allows DP objects to be used as building blocks within hierarchical models.

3.4.1 Density Estimation on Bounded Intervals

In some situations it will be necessary to estimate densities on bounded intervals. For example, it might be known that the observations y_i are restricted to lie within the interval $[0, 1]$. In this case, a mixture of Gaussian distributions is inappropriate, since this will assign positive probability to the whole real line. An alternative specification is a mixture of beta distributions, since the beta distribution only has positive mass in $[0, 1]$. The full model is the same as in the previous example but replacing k with the beta distribution. Likewise, similar code is used but with the specific beta mixture constructor.

```
y ← c(rbeta(150, 1, 3), rbeta(150, 7, 3))
dp ← DirichletProcessBeta(y, 1)
dp ← Fit(dp, 1000)
```

Figure 3.2 shows the resulting likelihood and posterior draws of the fit to the simulated data.

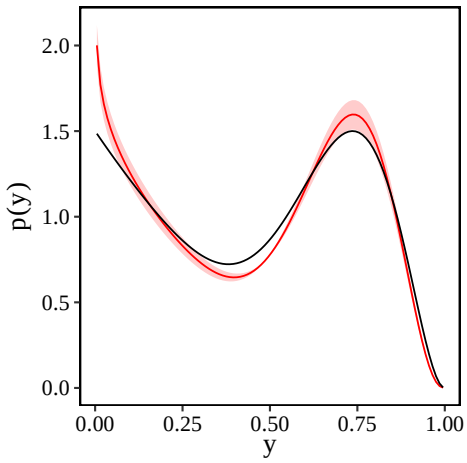


Figure 3.2: Estimated generating density using a beta Dirichlet process mixture model.

3.4.2 Cluster Analysis (Multivariate)

For any Dirichlet model each data point y_i is assigned a cluster parameter θ_i. The collection of unique values of cluster parameters θ_i^* allows for a natural way of grouping the data and hence the Dirichlet process is an effective way of performing cluster analysis. For multidimensional data it is most common to use a mixture of multivariate normal distributions to cluster the observations into appropriate groups. In the `dirichletprocess` package, the clustering labels is available at each fitting iteration and available to the user as `dp$clusterLabels`. Examples of the use of Dirichlet processes in clustering can be found in Teh et al. (2005) and Kim et al. (2006).

To demonstrate this the `faithful` dataset is used again, the length of the eruption as well as the amount of time between eruptions is considered. The full model can be written as

$$
\begin{aligned}
y_i &\sim N(y \mid \theta_i), \\
\theta_i &= \{\boldsymbol{\mu}_i, \Sigma_i\}, \\
\theta_i &\sim G, \\
G &\sim \mathrm{DP}(\alpha, G_0),
\end{aligned}
$$

where the prior parameters of G_0 take on their default value as shown in

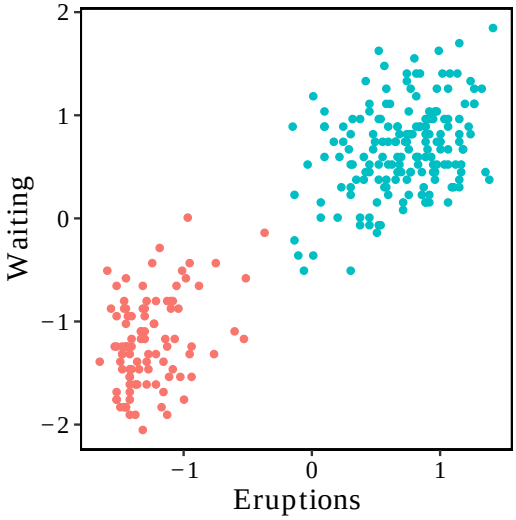


Figure 3.3: The colours of the points indicates that there are groups in the `faithful` dataset.

Section 3.3.3. The cluster labels are used to indicate which group each data point belongs to.

Again, the data is transformed such that each variable is zero mean and unit standard deviation before forming the `dirichletprocess` object and fitting for 1000 MCMC iterations.

```
faithfulTrans ← scale(faithful)
dp ←  DirichletProcessMvnormal(faithfulTrans)
dp ← Fit(dp, 1000)
plot(dp)
```

The final Gibbs sample of the cluster labels is used to analyse the model. Each cluster label is used to assign a colour so that it can be easily visualised which datapoint belongs to each cluster.

Here Figure 3.3 shows the last iteration of the cluster labels and the colours indicate the found clusters. Whilst this example use just two dimensions the code is generalised to work with as many dimensions as necessary.

3.4.3 Modifying the Observations

In some applications of using a Dirichlet process the data available can change from iteration to iteration of the sampling algorithm. This could be because the values of the data change, or because for a full data set $\mathbf{y} = y_1, \ldots, y_n$, only subsets of the data are used at each iteration. When fitting a DP object the function `ChangeObservations` is used to change the observations between iterations.

This function takes the new data, predicts what clusters from the previous fitting the new data belongs to and updates the clustering labels and parameters accordingly. A modified object with the new data associated to clusters and the function `Fit` is ready to be used to sample the cluster parameters and weights again.

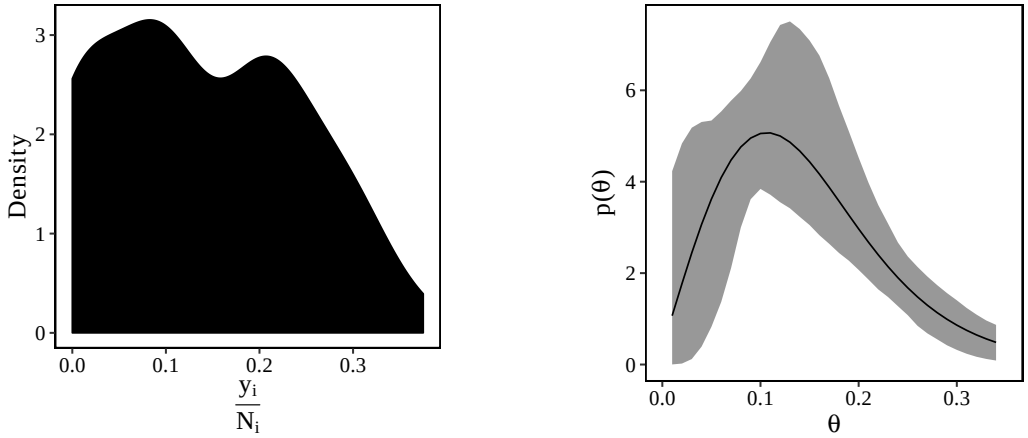
Example: Priors in Hierarchical Models

One application of observations changing with each iteration is using a Dirichlet process as a prior for a parameter in a hierarchical model. An example of hierarchical modelling comes from Gelman et al. (2014) involving tumour risk in rats. In this example, there are 71 different experiments, and during each experiment a number of rats are inspected for tumours, with the number of tumours in each experiment being the observed data. This data is n the package as the **rats** variable; the first column being the number of tumours in each experiment and the second being the number of rats.

A naive approach would model each experiment as a binomial draw with unknown θ_i and known N_i. A beta distribution is the conjugate prior for the binomial distribution and would be used as the prior on θ:

$$y_i \mid \theta_i, N_i \sim \text{Binomial}(N_i, \theta_i),$$
$$\theta_i \sim \text{Beta}(\alpha, \beta).$$

However, Figure 3.4a shows the empirical distribution of $\hat{\theta}_i = \frac{y_i}{N_i}$. This distribution shows hints of bimodality, something that a single beta distribution cannot capture and hence the prior choice of $p(\theta_i)$ is dubious. An alternative



(a) Empirical distribution of $\hat{\theta}_i$.

(b) Dirichlet prior model for $p(\theta_i)$.

Figure 3.4: Rat tumour risk empirical density and fitted prior distribution.

procedure is to use a nonparametric prior on the $\theta_i's$. Since these parameters are constrained to lie between 0 and 1, one choice might be a Dirichlet process mixture of beta distributions. This leads to the following model

$$\begin{aligned} y_i \mid \theta_i, N_i &\sim \text{Binomial}(N_i, \theta_i), \\ \theta_i &\sim \text{Beta}(\alpha_i, \beta_j), \\ \alpha_i, \beta_i &\sim F, \\ F &\sim \text{DP}(\alpha, G_0), \end{aligned}$$

where α and G_0 follow the default implementations of the `dirichletprocess` package. This model can then be written using `dirichletprocess` functions as follows:

```
# First the Dirichlet object is initialised
thetaDirichlet ← rbeta(nrow(rats), 0.01, 0.01)
dp ← DirichletProcessBeta(thetaDirichlet, 1
                          mhStep = c(0.002, 0.005),
                          alphaPrior = c(2, 0.5))
dp ← Fit(dp, 100)
# Then for a number of iterations
for(i in seq_len(its)){
  #Sample from the Dirichlet process
  postClusters ← PosteriorClusters(dp)

  #Sample a prior cluster parameter for each experiment
  wk ← sample.int(length(postClusters$weights),
                  nrow(rats), replace = T,
                  prob = postClusters$weights)

  #Transform the parameters for the rbeta function
  muPost ← postClusters$params[[1]][,,wk]
  nuPost ← postClusters$params[[2]][,,wk]
  aPost ← muPost * nuPost
  bPost ← (1 - muPost) * nuPost

  #Draw a new theta value from the posterior
  newTheta ← rbeta(nrow(rats), aPost + rats$y,
                                bPost + rats$N - rats$y)
  #Update the Dirichlet process
  dp ← ChangeObservations(dp, newTheta)
  dp ← Fit(dp, 100, updatePrior = T, progressBar = F)
}
```

Note the reason why the observations are changing is because the DP mixture model is applied to the θ_i parameters, which are resampled (and hence

have different values) during each MCMC iteration.

Figure 3.4b reveals that the DP is a more suitable prior than the beta distribution. This confirms the finding from the empirical distribution that the data is bimodal.

3.4.4 Hierarchical Dirichlet process

A hierarchical Dirichlet process (Teh et al., 2005) can be used for grouped data. Each individual dataset is modelled using a separate Dirichlet process but where the base measure itself is also a Dirichlet process. Mathematically this can be expressed as

$$
\begin{aligned}
y_{ij} &\sim F(\theta_{ij}), \\
\theta_{ij} &\sim G_j, \\
G_j &\sim \mathrm{DP}(\alpha_j, G_0), \\
G_0 &\sim \mathrm{DP}(\gamma, H),
\end{aligned}
$$

for each dataset $j = 1, \ldots, n$ with data $y_{1j}, \ldots y_{Nj}$ there is a separate Dirichlet process generating the required parameters θ_{ij}. Using the stick-breaking construction, G_0 can be expressed as an infinite sum (Key Property 2), the same procedure can be applied to the G_j measures

$$
\begin{aligned}
G_j &= \sum_{i=1}^{\infty} \pi_{jk} \delta_{\phi_k}, \quad \phi_k \sim H, \\
\pi'_{jk} &= \mathrm{Beta}\left(\alpha_j \beta_k, \alpha\left(1 - \sum_{l=1}^{k} \beta_l\right)\right), \quad \pi_{jk} = \pi'_{jk} \prod_{l=1}^{k-1} (1 - \pi'_{jl}), \\
\beta'_k &\sim \mathrm{Beta}\left(1, \gamma\right), \quad \beta_k = \beta'_k \prod_{l=1}^{k-1} (1 - \beta'_l),
\end{aligned}
\tag{3.5}
$$

where H is the global distribution and each G_j the local distribution. For further details see Teh et al. (2005).

To fit a hierarchical Dirichlet process, Algorithm 8 from Neal (2000) is used as detailed in Section 3.3.1. Each datapoint y_{ij} is further assigned as label k_{ij} which indicates the global parameter ϕ_k it is the data point is assigned to.

Each global parameter is updated using the available data across datasets.

$$\phi_k \mid x_{ij} = h(\phi_k) \prod_{k_{ij}=k} f(x_{ij} \mid \phi_k), \tag{3.6}$$

where h is the density of the global distribution H and f is the density of the mixing distribution F. From these updated parameters a new G_j can be drawn using Key Property 5.

For a hierarchical DP model each individual concentration parameter α_j can be inferred using the usual algorithm as per Section 3.3.2 without modification for each individual dataset. For the top level concentration parameter γ, the number of unique cluster parameters across all the individual G_j's is used for n in the sampling of γ.

In this example two synthetic data sets are used to fit a hierarchical Dirichlet process. A beta Dirichlet mixture model is used and therefore two known beta distributions are used to simulate the test data.

$$\begin{aligned} y_1 &\sim \text{Beta}(0.25, 5) + \text{Beta}(0.75, 6), \\ y_2 &\sim \text{Beta}(0.25, 5) + \text{Beta}(0.4, 10), \end{aligned}$$

where there is a common group of parameters between the two datasets.

```
# Generate the toy data
mu ← c(0.25, 0.75, 0.4)
tau ← c(5, 6, 10)
a ← mu * tau
b ← (1 - mu) * tau
y1 ← c(rbeta(100, a[1], b[1]), rbeta(100, a[2], b[2]))
y2 ← c(rbeta(100, a[1], b[1]), rbeta(100, a[3], b[3]))
```

The appropriate constructor function is used to create a hierarchical Dirichlet object with uninformative priors for the global base distribution and then fitted for 5000 iterations.

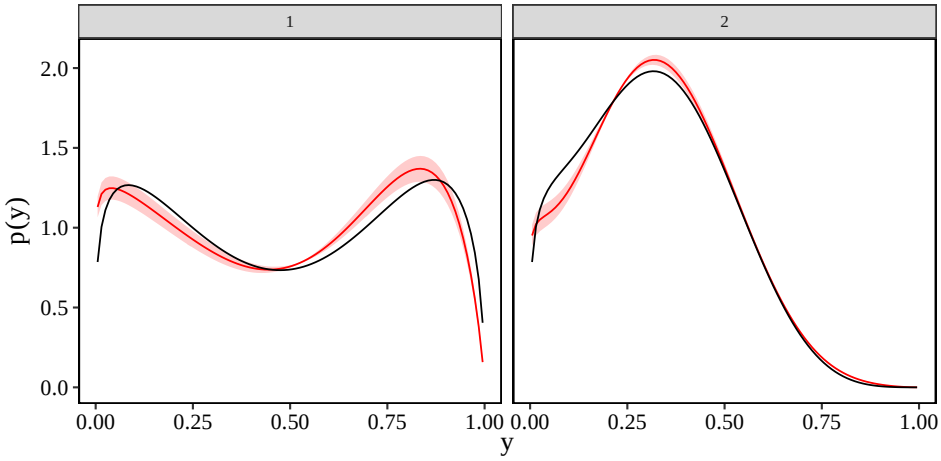


Figure 3.5: Hierarchical beta Dirichlet process mixture results.

```
# Fit the toy data
dpobjlist ← DirichletProcessHierarchicalBeta(list(y1, y2),
  maxY=1,
  hyperPriorParameters = c(1, 0.01),
  mhStepSize = c(0.1, 0.1),
  gammaPriors = c(2, 4),
  alphaPriors = c(2, 4))
dpobjlist ← Fit(dpobjlist, 5000, TRUE)
```

The creator function `DirichletProcessHierarchicalBeta` returns a list of `dirichletprocess` objects for each dataset (in this case two objects), a vector containing the global stick breaking weights, a list of the global parameters and the variable containing the global concentration parameter γ. The function `Fit` updates the cluster allocations locally of each `dirichletprocess` object using Algorithm 8 from Neal (2000), then the local concentration parameter α_j is updated. The global cluster parameters are then updated using all the data pooled from the individual datasets by drawing from Eq. (3.6). Using these parameters a new sample of G_0 is taken from which the individual G_j's are also drawn using the above procedure in Equation (3.5). The resulting G_j's from the above example are plotted in Figure 3.5.

3.4.5 Stick-Breaking Representation

The stick-breaking representation of a Dirichlet process allows for easy posterior inference using Key Property 3 and 5 (Section 3.3). In the `dirichletprocess` package drawing from the posterior is easily achieved using both `PosteriorClusters` and `PosteriorFunction` depending on users need.

- `PosteriorClusters`: Returns the posterior clusters ϕ_k and weights w_k as a list for the user.

- `PosteriorFunction`: Draws the posterior clusters and uses the likelihood function of the mixing distribution to return a function with appropriate posterior weights and parameters. This is a sample of the measure F for models of the form in Eq. (3.2).

3.5 Advanced Features

The material in this section can largely be skipped as long as the user is getting good results from the `dirichletprocess` package using the default specifications. However inevitably problems will arise due to (e.g.) the default hyper parameters being inadequate for a particular data set, or the sampler seeming not to converge due to bad initial default parameter values. Alternatively, the user may wish to use a mixture kernel other than the ones included in the package (normal, beta, Weibull, etc). In this case, the user will need to know what is going on under the hood so that they can change the default settings to better suited values, or otherwise modify the internal sampling procedure. The package aims to ensure that this will usually only require changing a small number of the parameters which control the sampling behaviour, but understanding what needs to be changed (and why) requires some comprehension of how the objects are constructed.

3.5.1 Structure of a DP Object: The Gory Details

A DP object is defined by its kernel mixing distribution. Each mixing distribution has the following functions and variables associated with its class

- `Likelihood(...)`: a function which specifies the density of the mixture kernel $k(y \mid \theta)$.

- `PriorDraw(...)`: a function which returns a random sample of size n from the DP base measure G_0.

- `g0Priors`: a list of parameters for the base measure G_0.

For a conjugate mixing distribution the posterior distribution of θ is tractable and can be sampled directly. The marginal distribution of the data can also be explicitly calculated and evaluated. Therefore two functions are needed to complete the specification of a conjugate mixture model:

- `PosteriorDraw(...)`: a function that returns a sample of size n given data y from the posterior distribution of θ, i.e. a sample from the distribution of $p(\theta \mid y)$.

- `Predictive(...)`: a function that returns the value of the marginal distribution of the data $f(y) = \int k(y, \theta) \mathrm{d}G(\theta)$.

With these specified, the `Fit` function can be used to fit the DP, which carries out the Chinese Restaurant Sampling procedure using Algorithm 8 (Neal, 2000).

For a nonconjugate mixing distribution the posterior distribution $p(\theta \mid y)$ is intractable and cannot be sampled from directly. Neither can the marginal distribution of the data be calculated. Instead the Metropolis-Hastings algorithm is used to sample from the distribution $p(\theta \mid y)$ (Hastings, 1970). The Metropolis-Hastings algorithm works by generating a candidate parameter θ^{i+1} and accepting this candidate value as a sample from the posterior with probability proportional to $\frac{k(y|\theta^{i+1})p(\theta^{i+1})}{k(y|\theta^{i})p(\theta^{i})}$. Typically, the candidate parameter is distributed as $\theta_{i+1} \sim N(\theta_i, h^2)$. From this, the nonconjugate mixture model requires two additional functions and an extra parameter to be defined.

- `PriorDensity(...)`: a function which evaluates $p(\theta)$ which is the DP base measure G_0 for a given θ.

- `mhParameterProposal(...)`: a function that returns a candidate parameter to be evaluated for the Metropolis-Hastings algorithm.

- `mhStepSize`: h, the size of the step to make when proposing a new parameter for the Metropolis-Hastings algorithm.

Once the appropriate mixing distribution is defined a `dirichletprocess` object is created which contains the data, the mixing distribution object and the parameter α. Then the rest of `dirichletprocess` class functions are available.

By using the default constructor functions `DirichletProcessBeta` etc. the base measure prior parameters are chosen to be non-informative, see Section 3.3.3 for the specific values of the prior parameters.

3.5.2 Creating New Dirichlet Process Mixture Types

The `dirichletprocess` package currently implements Dirichlet process mixture models using Gaussian, beta and Weibull kernels. While these kernels should be appropriate for most applications, there will inevitably be times when a user wants to fit a DP model for a kernel which has not been implemented, or otherwise wants to do something complex with a DP which goes beyond the scope of this package. In anticipation, this the package has been designed to for users to construct their own mixture models which can then automatically use the implemented algorithms for fitting a Dirichlet process.

The functions in the package are designed to work on S3 `R` objects, where each object represents a type of Dirichlet process mixture (e.g Gaussian or beta). In order to create new types of Dirichlet process mixtures, the user must create a new S3 object type which encapsulates his model and ensure that its specifications correspond to those of the package. If this is done, then all the package functions for resampling and prediction should continue work on the new DP type. This means that the package can hopefully be used for DP applications that were not considered when writing it, while saving the user from having to write their own functions for resampling and fitting.

To illustrate how this works, this section will work through an extended

example of how to create a new S3 type which represents a DP mixture model. This will be a new mixture not implemented in the `dirichletprocess` package. The S3 objects and associated functions are constructed so that the user will be able to create their own.

Conjugate Mixture

Suppose there is a particular scenario that requires a Dirichlet process mixture of Poisson distributions. This could involve modelling the counts of some observation such as goals in a soccer match. Like all Bayesian inference, a prior must be chosen for the unknown parameter. As the conjugate prior for the Poisson distribution is the gamma distribution this serves as a good example of how to implement a conjugate mixture model with a Dirichlet process.

Firstly, the likelihood of the Poisson distribution is required

$$k(x \mid \theta) = \frac{\theta^x \exp(-\theta)}{x!},$$

as there is only one parameter in the Poisson distribution the parameter list θ is of length 1.

```
Likelihood.poisson ← function(mdobj, x, theta){
  return(as.numeric(dpois(x, theta[[1]])))
}
```

Note that the [[1]] part is essential, since parameters are internally represented as lists even when they only have one element.

Next, the random prior sample function which draws a value of θ from the base measure G_0. The conjugate prior to the Poisson distribution is the gamma distribution

$$G_0 \sim \text{Gamma}(\alpha_0, \beta_0).$$

```
PriorDraw.poisson ← function(mdobj, n){
  draws ← rgamma(n,
                 mdobj$priorParameters[1],
                 mdobj$priorParameters[2])
  theta ← list(array(draws, dim=c(1,1,n)))
  return(theta)
}
```

The prior parameters α_0, β_0 are stored in the mixing distribution object `mdobj`.

The `PosteriorDraw` function to sample from the posterior distribution of θ is tractable as the base measure G_0 is conjugate. This results in a direct sampling function from the posterior distribution

$$\theta \mid x_1, \ldots, x_n \sim \text{Gamma}\left(\alpha_0 + \sum_{i=1}^{n} x_i, \beta_0 + n\right),$$

which using the inbuilt `rgamma` function is trivial.

```
PosteriorDraw.poisson ← function(mdobj, x, n=1){
  priorParameters ← mdobj$priorParameters
  lambda ← rgamma(n,
              priorParameters[1] + sum(x),
              priorParameters[2] + nrow(x))
  return(list(array(lambda, dim=c(1,1,n))))
}
```

Finally the marginal distribution of the data $f(y)$ can be evaluated as it is a conjugate mixture model and translated into the appropriate `R` function.

```
Predictive.poisson ← function(mdobj, x){
  pp ← mdobj$priorParameters
  pred ← numeric(length(x))
  for(i in seq_along(x)){
    alphaPost ← pp[1] + x[i]
    betaPost ← pp[2] + 1
    pred[i] ← (pp[2] ^  pp[1]) / gamma(pp[1])
    pred[i] ← pred[i] * gamma(alphaPost) / (betaPost^alphaPost)
    pred[i] ← pred[i] * (1 / prod(factorial(x[i])))
  }
  return(pred)
}
```

With these functions written for the Poisson mixture model the constructor function `MixingDistribution` needs to be called to create a new object that can be used by the Dirichlet process constructor function, `DirichletProcessCreate`.

The constructor function `MixingDistribution` creates an object of class `distribution`, in this case `poisson`, with prior parameters $\alpha_0, \beta_0 = 1$ and that it is conjugate.

```
poisMd ← MixingDistribution(distribution="poisson",
  priorParameters = c(1, 1), conjugate="conjugate")
```

This object is now ready to be used in a `dirichletprocess` object and the appropriate sampling tasks can be carried out. To demonstrate this new model, new data is simulated and fitted using a Dirichlet process with the new mixing distribution.

```
y ← c(rpois(150, 3), rpois(150, 10)) #generate sample data
dp ← DirichletProcessCreate(y, poisMd)
dp ← Initialise(dp)
dp ← Fit(dp, 1000)
```

As Figure 3.6 shows, the true generating function has been recovered. This

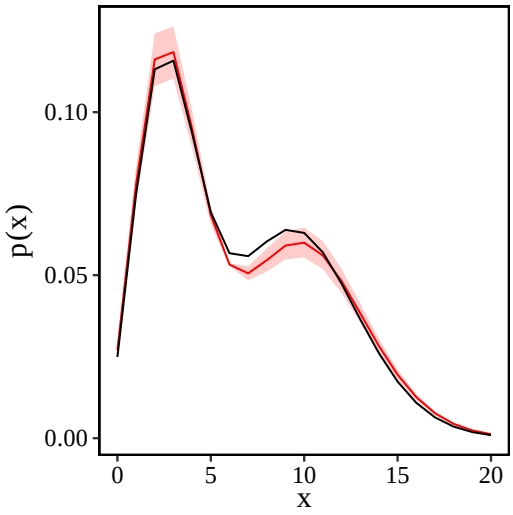


Figure 3.6: The true and estimated distributions from the Poisson mixture model.

shows how easy it is for the user to create their own mixture models using the `dirichletprocess` package. In terms of performance, 1000 iterations took roughly one minute to sample.

Nonconjugate Mixture

Suppose that a particular application requires a Dirichlet process mixture of gamma distributions. Additional steps must be taking when creating the necessary functions at the gamma distribution does not have a conjugate prior distribution.

Again the likelihood function is needed. The gamma distribution has two parameters α, β, therefore the list θ will also have two components. The density of the gamma distribution can be written as

$$k(y \mid \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} y^{\alpha-1} e^{-\beta y},$$

which can be easily translated using `dgamma` in R.

```
Likelihood.gamma ← function(mdobj, x, theta){
  return(as.numeric(dgamma(x, theta[[1]], theta[[2]])))
}
```

The function to draw random parameters from the base measure G_0 is also needed. For the parameters of the gamma distribution two exponential distributions will be used as the base measure $\alpha \sim \text{Exp}(\alpha_0)$ and $\beta \sim \text{Exp}(\beta_0)$.

```
PriorDraw.gamma ← function(mdobj, n=1){
  theta ← list()
  theta[[1]] = array(rexp(n, mdobj$priorParameters[1]),
               dim=c(1,1, n))
  theta[[2]] = array(rexp(n, mdobj$priorParameters[2]),
               dim=c(1,1, n))
  return(theta)
}
```

In contrast to the conjugate example, the posterior distribution must be sampled using the Metropolis-Hastings algorithm and a function that calculates the prior density for a given α, β is required.

```
PriorDensity.gamma ← function(mdobj, theta){
  pp ← mdobj$priorParameters
  thetaDensity ← dexp(theta[[1]], pp[1])
  thetaDensity ← thetaDensity * dexp(theta[[2]], pp[2])
  return(as.numeric(thetaDensity))
}
```

Finally, the Metropolis-Hastings algorithm also needs a function that perturbs the parameters to explore the posterior distribution. For the gamma distribution the parameters α, β are strictly positive and the new parameter proposals must be constrained. This is achieved by taking the absolute value

of a standard normal perturbation

$$\alpha^{i+1} = |\alpha^i + h \cdot \eta|,$$
$$\eta \sim N(0,1),$$
$$\beta^{i+1} = |\beta^i + h \cdot \zeta|,$$
$$\zeta \sim N(0,1),$$

again this is easy to translate into R:

```
MhParameterProposal.gamma ← function(mdobj, oldParams){
  mhStepSize ← mdobj$mhStepSize
  newParams ← oldParams
  newParams[[1]] ← abs(oldParams[[1]] + mhStepSize[1]*rnorm(1))
  newParams[[2]] ← abs(oldParams[[2]] + mhStepSize[2]*rnorm(1))
  return(newParams)
}
```

The mixing distribution object is now ready to be constructed using the `MixingDistribution` function. The arguments of this function specify the prior parameters α_0, β_0 and set the scale h at which the new parameter proposals are made using the parameter `mhStepSize`.

```
gammaMd← MixingDistribution("gamma",
              priorParameters = c(0.1, 0.1),
              "nonconjugate",
              mhStepSize=c(0.1, 0.1))
```

The `dirichletprocess` object can now be created and fit to some test data. As it is a new type of mixture, it must be initialised.

```
y ← c(rgamma(100, 2, 4), rgamma(100, 6, 3))
dp ← DirichletProcessCreate(y, gammaMd)
dp ← Initialise(dp)
dp ← Fit(dp, 1000)
```

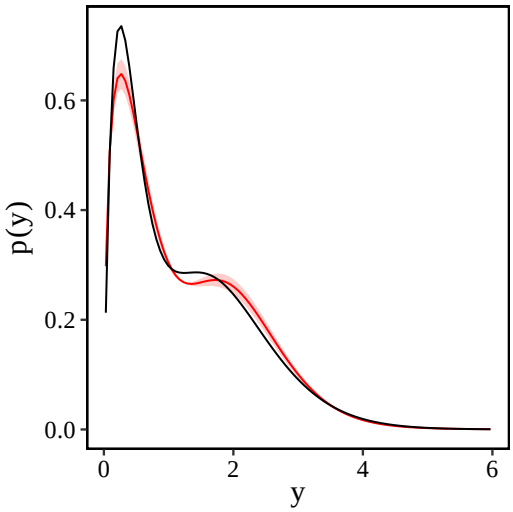


Figure 3.7: The results of implementing the new gamma mixture model.

From Figure 3.7 the true distribution has been correctly identified.

Both of these examples show the ease and flexibility of how a new mixture model can be implemented without needing to know the details of the sampling algorithms from Neal (2000). The fitting procedure took roughly 5 minutes for 1000 iterations. This increase over the conjugate model is down to the need to perform a Metropolis Hastings step with each cluster paramter update.

Overall, the performance of this package is slightly slower than other Dirichlet process implementations not using R, but what is lost in sampling speed is gained in ease of model building and iterating. As all the functions and building blocks are written in native `R` code any user of the `R` can comprehend and construct their own Dirichlet process models using this package.

3.5.3 Resampling Component Indexes and Parameters

When calling the `Fit` function on a DP object the component indexes and parameters are resampled following Algorithm 4 for the conjugate case and Algorithm 8 for the nonconjugate case using the specification from Neal (2000). For both types of DP mixture the two functions that do the majority of the work are `ClusterComponentUpdate` and `ClusterParameterUpdate`.

In a conjugate DPMM new component indexes and new cluster parameters

are drawn directly from the predictive and posterior distributions making the algorithm very efficient. In such cases the only option available to users is to change the prior parameters of the base distribution G_0. Ensuring that the base distribution is correctly parameterised with sensible values for the underlying data will provide optimal performance for the fitting algorithm.

However, in a nonconjugate case new cluster components are proposed from the chosen prior distribution and new cluster parameters are sampled using the Metropolis-Hastings algorithm to obtain a posterior sample. By using the Metropolis-Hastings algorithm, the parameters in question are proposed using a random walk but constrained to the particular support of the parameter. For example, the parameters in a Weibull distribution are strictly positive, therefore the random walk is restricted to fall on the positive real line. An ill proposed prior distribution can severely affect the convergence of the fitting process. The parameter `mhStepSize` in the constructor function for a nonconjugate mixture controls the scale of new parameter proposals for the random walk. When creating a new DP object, the constructor function has a flag `verbose` that outputs an estimated acceptance ratio [2]. As with the conjugate case, care must be taken to ensure that the base measure is well suited for the data.

Overriding Default Behaviour

For both conjugate and nonconjugate mixture models, the user can write their own

`ClusterComponentUpdate` and `ClusterParameterUpdate` functions to override the default behaviour. The user can still benefit from the other S3 methods and structures implemented in `dirichletprocess` but with their custom sampling schemes.

For the nonconjugate mixture models there is a further option available to change the component index and parameter resampling. In Algorithm 8 of Neal

[2]For optimal performance of the Metropolis-Hastings algorithm this value should be around 0.234 (Gelman et al., 1996)

(2000) each datapoint can form a new cluster with parameter drawn from the base measure, these proposals are called 'auxiliary' variables and m are drawn for each data point. By default $m = 3$. However this can be changed in the `Initialise(dp, ...,m=m)` function. Using more auxiliary variables can lead to more changes in the component indexes and greater exploration of the base measure but at the cost of computational time.

3.5.4 Resampling the Base Measure, G_0

It is helpful that the user knows how to best set the parameters of the base measure to correctly represent the underlying data. However, whilst desirable this is not always practical. In which case `dirichletprocess` offers functionality to use hyper-prior parameters on G_0 and update them with each iteration.

For the mixing distributions that allow for re-sampling of the base measure, it is simple to include the flag `Fit(dp,...,updatePrior=TRUE)`. At each fitting iteration the base measure with variable parameters will be updated based on the current cluster parameters. For details on the exact specification of the hyper-prior distributions for each implemented mixture kernel see Section 3.3.3. If a user wishes to change the default prior on the hyper parameters then it is as simple as changing the `PriorParametersUpdate` function for the mixing distribution object.

3.5.5 Component Prediction

Given a fitted DP object and some new data $\hat{y}$ the command `ClusterLabelPredict` can be used to predict the cluster labels of this new data . Using the appropriate algorithm for a conjugate or nonconjugate mixture model the cluster label probabilities are calculated from the new data $\hat{y}$, these probabilities are then sampled once to obtain a cluster label. It is these cluster labels that are returned with the appropriate cluster parameters.

Referring back to the example in Section 3.4.2 where a Dirichlet process is used to cluster the `faithful` dataset, this example is updated to withhold the last five entries of the data as the prediction set and use `ClusterLabelPredict`

to estimate their cluster allocation.

```
faithfulTrans ← scale(faithful)
trainIndex ← 1:(nrow(faithfulTrans)-5)

dp ←  DirichletProcessMvnormal(faithfulTrans[trainIndex, ])
dp ← Fit(dp, 1000)

labelPred ← ClusterLabelPredict(mvDPFaith,
                faithfulTrans[-trainIndex, ])
```

The function `ClusterLabelPredict` works by calculating and sampling the clustering label from the probability that each test point $\hat{y}_j$ belongs to each cluster θ_i or should form its own cluster proportional to α

$$p(i) \propto n_i k(\hat{y}_j \mid \theta_i),$$
$$p(i = \text{new}) \propto \alpha \int k(\hat{y}_j, \theta_i) \mathrm{d}G_0.$$

The function returns a list with multiple entries:

- The predicted cluster labels for the data under `labelPred$componentIndexes`.
- The cluster parameters assigned to the predicted data points in case a new cluster is predicted `labelPred$clusterParameters`.
- The new number of data points per cluster `labelPred$pointsPerCluster`.
- The total number of clusters are also returned `labelPred$numLabels` as this can change with each prediction.

Figure 3.8 shows the test data being correctly identified with the appropriate cluster.

3.5.6 Working with Censored Observations

The following example is intended to illustrate how simple the `dirichletprocess` package makes it for the user to extend Dirichlet process mixture modelling to situations which are not directly supported by the package.

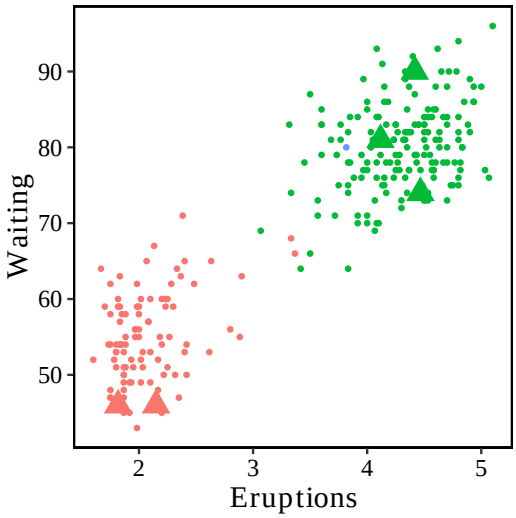


Figure 3.8: The predicted labels of the last 5 entries of the `faithful` dataset against the training data. The predicted values are indicated by a solid colour and triangle shapes.

Survival analysis is an area of statistics concerned with analysing the duration of time before an event happens such as a failure in a mechanical system or a death in a biological context. The aim is to construct a survival function which, as the name indicates, shows how the probability of not experiencing the event changes with time. Survival type data is often generated by observational studies which result in censoring. In the context of medical statistics, censoring occurs due to finite time periods of the studies. When analysing the effects of medical treatments patients events can be censored for a variety of reasons. This is a missing data problem as it is no longer known the exact time at which an event occurred, just that it occurred before or after a specific time. Right censoring is when a patient is yet to be effected by the event after a study ends. Left censoring is when it is not known exactly when the event occurred, just that it occurred before the study started. To deal with this censored information the likelihood must be adapted.

One approach for modelling such data nonparametrically is a Dirichlet process mixture of Weibull distributions. The `dirichletprocess` package does not directly support the analysis of censored data – as stated throughout,

the purpose of the package is not to provide the user with a small number of functions for solving predefined problems, but to make it easy to use Dirichlet process mixtures in a wide variety of contexts. As such, it is very simple for the user to extend the functionality of the package to allow for censoring.

Example: Censored Observations

In this example, the work of Kottas (2006b) will be replicated and use leukaemia remission times taken from Lawless (2011). This dataset contains two groups of censored observations that two different treatments have on leukaemia remission times. A censored Weibull DP is used to model each of the datasets and compare the survival functions. The full model can be written as

$$\begin{aligned} y_i &\sim \text{Weibull}(y_i \mid a_i, b_i), \\ a_i, b_i &\sim G, \\ G &\sim \text{DP}(\alpha, G_0), \end{aligned}$$

where the Weibull density and G_0 follow the form shown in Section 3.3.3.

Censored data can come in the form of two columns - the time it takes for the event to occur and an indicator variable; 1 for a right censored observation, 0 otherwise. Therefore the density of the Weibull distribution can be written as

$$\begin{aligned} k(y \mid a, b) &= \frac{a}{b} y^{a-1} \exp\left(-\frac{y^a}{b}\right) \quad \text{for uncensored,} \\ k(y \mid a, b) &= 1 - \exp\left(-\frac{y^a}{b}\right) \quad \text{for censored.} \end{aligned}$$

This likelihood needs to be translated into the appropriate function for the `dirichletprocess` package.

```
Likelihood.weibullcens ← function(mdobj, x, theta){
  a = theta[[1]][,,,drop=TRUE]
  b = theta[[2]][,,,drop=TRUE]

  y ← as.numeric(
  b^(-1) * a * x[,1]^(a-1) * exp(-b^(-1) * x[, 1]^a))
  y_cens ← as.numeric(1 - exp(-x[,1]^a / b))

  if(nrow(x) == 1){
    if(x[,2] == 0) return(y)
    if(x[,2] == 1) return(y_cens)
  }
  else{
    y_ret ← y
    y_ret[x[, 2] == 1] ← y_cens[x[, 2]==1]
    return(y_ret)
  }
}
```

Two different `mixingDistribution` objects are created for each of the data sets. Again, using the `MixingDistribution` constructor function and provide the resulting object with a custom class such that it can use the new likelihood function.

```
mdobjA ← MixingDistribution("weibullcens", c(1,2,0.5),
             "nonconjugate",
             mhStepSize=c(0.11,0.11),
             hyperPriorParameters=c(2.222, 2, 1, 0.05))
mdobjB ← MixingDistribution("weibullcens", c(1,2,0.5),
             "nonconjugate",
             mhStepSize=c(0.11,0.11),
             hyperPriorParameters=c(2.069, 2, 1, 0.08))

class(mdobjA) ← c("list", "weibullcens",
                  "weibull", "nonconjugate")
class(mdobjB) ← c("list", "weibullcens",
                  "weibull", "nonconjugate")
```

The sampling is then carried out as normal with no other changes needed. The default functions available for the Weibull mixture model are applied to our custom `dirichletprocess` object.

```
dpA ← DirichletProcessCreate(data_a, mdobjA, c(2, 0.9))
dpA ← Initialise(apA)

dpB ← DirichletProcessCreate(data_b, mdobjB, c(2, 0.9))
dpB ← Initialise(dpB)

dpA ← Fit(dpA, 500, TRUE)
dpB ← Fit(dpB, 500, TRUE)
```

The survival function is calculated as $S(y) = 1 - \exp(-\frac{y^a}{b})$ and the parameter samples are easily extracted from the fitted objects.

The resulting density and survival estimate is shown in Figure 3.9 which correctly replicate the findings of Kottas (2006b).

To fully understand what has happened here, it is vital to understand that the DP is defined by its base likelihood and G_0 distribution. In creating a new mixing distribution of class `weibullcens` and `weibull` the new likelihood

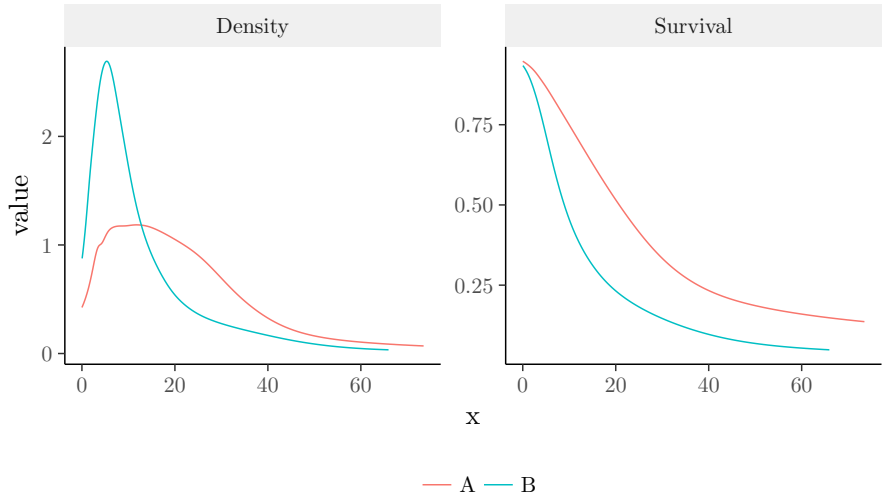


Figure 3.9: Point estimates for the survival and density functions of the two treatments.

can be used whilst using all the previous functions of a Weibull DP mixture. This makes it trivial to define your own likelihoods using the foundations laid in the different classes available.

3.6 Point Process Intensity Estimation

One practical application of beta mixture models is the estimation of an inhomogeneous Poisson process intensity function. As stated previously a Poisson process is a collection of points in space distributed with rate λ. In the inhomogeneous case, the intensity is dependent on time and as such the number of events at time t can be written as

$$N(t) \sim \text{Poisson}(\lambda(t)). \tag{3.7}$$

In parametric estimation, a functional form of $\lambda(t)$ would be constructed i.e. $\alpha_0 + \alpha t$ and the parameters $\{\alpha_0, \alpha\}$ would be estimated. However, the accuracy of such a method would be dependent on correctly identifying the parametric form of $\lambda(t)$. With the nonparametric methods that a DPMM provides, such assumptions can be ignored and an intensity function can be built without the need to specify a parametric form. Firstly, it is assumed that $\lambda(t) =$

$\lambda_0 h(t)$ where $\int_0^T h(t)\mathrm{d}t = 1$, i.e. the intensity rate can be decomposed into an amplitude λ_0 controlling the number of events and a density $h(t)$ controlling the distribution of the events over the window of observation $[0, T]$. To infer the value of λ_0 a conjugate gamma prior can be used and thus the posterior distribution can be directly sampled.

In this example, an intensity rate $\lambda(t)$ will be estimated with each iteration and used to update the data the model uses to infer the parameters. The full model can be written as

$$\begin{aligned} N &\sim \text{Poisson}(\lambda(t)), \\ \lambda(t) &= \lambda_0 h(t) \\ h(t) &= \int k(t \mid \theta)\mathrm{d}F, \\ F &\sim \text{DP}(\alpha, G_0), \end{aligned}$$

where k and G_0 are as per Section 3.3.3 for the beta distribution mixture models. The posterior distribution of G is sampled using Key Property 5 (Section 3.3) which states that a sample of G can be drawn independently of the data using the stick-breaking representation of the data and the model parameters θ.

In this toy model 500 event times are simulated using the intensity function $\lambda(t) = \sin^2 \frac{t}{50}$. Instead of passing the full data set into the Dirichlet process object, a random sample of 100 of these event times are used instead.

```
# Generate some toy data
y ← cumsum(runif(1000))
pdf ← function(x) sin(x/50)^2
accept_prob ← pdf(y)
pts ← sample(y, 500, prob=accept_prob)
```

The Dirichlet process object is fitted, then a posterior sample is drawn of the intensity function $\hat{\lambda}(t)$ and sample 150 new points from the full data set with probabilities proportional to $\hat{\lambda}(t)$. The Dirichlet process object is then modified with the new data and the process is repeated.

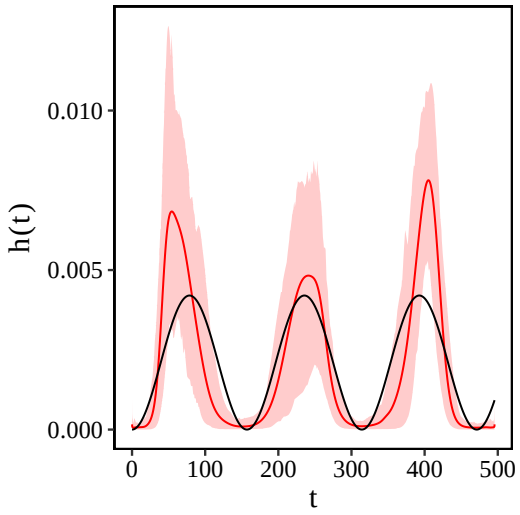


Figure 3.10: Estimation of the inhomogeneous Poisson process using stick breaking.

```
# Create the object and initially fit it
dp ← DirichletProcessBeta(sample(pts, 100),
  maxY = max(pts)*1.01,
  alphaPrior = c(2, 0.01))

dp ← Fit(dp, 100, TRUE)

for(i in seq_len(2000)){
  # For each iteration sample the function
  lambdaHat ← PosteriorFunction(dp)
  # Sample a new dataset
  newPts ← sample(pts, 150, prob=lambdaHat(pts))
  newPts[is.infinite(newPts)] ← 1
  newPts[is.na(newPts)] ← 0
  # Update the object
  dp ← ChangeObservations(dp, newPts)
  dp ← Fit(dp, 2, TRUE)
}
```

Figure 3.10 shows the true intensity function is being recovered even though the full dataset is never observed.

3.7 Final Remarks

This package provides the foundation for the nonparametric work in the thesis. It has taken the standard algorithms for fitting a Dirichlet process model from Neal (2000) and produced a comprehensive interface for anyone to use. In the next two chapters of this thesis multiple sections of this chapter will be referred to and explicitly used with within a Hawkes process.

Chapter 4

Bayesian Nonparametric Hawkes Processes with Application to Extreme Values

In this thesis the first application of the nonparametric Hawkes process is the modelling of the occurrence of extreme events in a time series. In time series data, any event that is larger than a predetermined threshold is regarded as extreme and the occurrence of such an event is believed to be sufficiently rare. In this chapter a Hawkes process will be used to model the extreme events as self-exciting such that the occurrence of an extreme event increases the probability of further extreme events, which in turn leads to clusters in time of when these extreme events occur.

Effective risk management often requires an estimate of the probability that large events will occur during a given time frame. For example, suppose that m terrorist attacks have previously occurred over a period of T years, at times $t_1, t_2, \ldots, t_m$. For each attack time t_i, let r_i be a mark denoting the corresponding number of fatalities. Based on this historical data, it may be important for insurance or disaster response purposes to have a probabilistic estimate for the chance of a terrorist attack causing more than z fatalities in the next 5 years for a given value of z. Similar problems are also often considered in other fields such as natural hazard modelling where the events correspond

to earthquakes and the marks correspond to earthquake magnitudes (Bray and Schoenberg, 2013), and in finance where the events are the times at which the increase or decrease in a company's stock price are observed, and the marks represent the size of the change (Chavez-Demoulin and McGill, 2012).

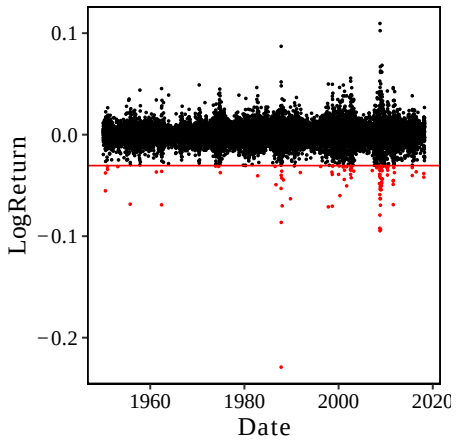
The estimation task can be broken down into two parts. First, a prediction is made for the number of large events which are likely to occur in a given period along with their occurrence times. Second, for each predicted event r_i, the probability $p(r_i > z)$ is estimated. Direct estimation of this latter quantity is difficult since it usually involves inference about extreme quantiles of the distribution governing r_i. This can be highly sensitive to the parametric assumptions made about the distribution, where the danger of misspecification can invalidate parameter inference results (White, 1982). As such, it is usual to instead rely on asymptotic results from the field of extreme value theory (EVT) to avoid the need to make strong parametric assumptions.

For this purpose, the well known Pickands-Balkema-de Haan (PBH) from EVT states that as long as z is sufficiently large, the distribution $p(r_i > z)$ can be approximated by a Generalised Pareto Distribution (GPD) as long as the distribution satisfies certain regularity conditions (Balkema and de Haan, 1974). This means that the distribution of the extreme marks, denoted by r_i^*, can be modelled without the need for parametric assumptions of the full distribution. This has led to the widely-used Peaks-Over-Threshold (PoT) approach for modelling extreme values where the occurrence times t_i^* of extremes is assumed to follow a Poisson process, with the corresponding marks r_i^* following a GPD. However the direct application of this methodology to real-world problems is limited by the strong stationarity assumptions that it requires. Specifically, a straightforwards EVT analysis is only possible if: a) the time occurrences t_i^* of large values can be modelled by a (possibly inhomogeneous) Poisson process and b) the conditional distribution of the magnitude r_i^* exceeding some threshold z are independent draws from a GPD with constant parameters.

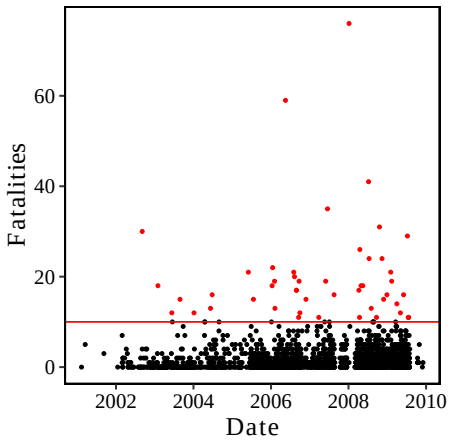
Unfortunately these assumptions often do not hold. Figure 4.1a shows the daily log returns of the S&P 500 stock index. This index is designed to provide an easily calculable measure of the US stock market and an indicator of the general outlook of the economy. Many risk measures are based on the S&P 500 daily closing price therefore it is of great interest to predict when the next large drop could occur and how severe it could be. In Figure 4.1a notable clusters of large movements can be observed; Black Monday in 1987, the dot-com bubble era at the turn of the 21st century and finally the 2008 financial crisis. All three periods are easily identified by the larger values of log return, both positive and negative. This in turn shows that the occurrence of these large events appears to be inhomogeneously distributed and clustered. Furthermore, there appears to be structural change in the distribution of r_i^*, with the losses during the 2008 financial crisis being larger and a higher frequency than previously seen in other downturns. It would hence be unwise to fit single stationary GPD to this data

Similarly, Figure 4.1b shows the time series of fatalities that occurred in terrorist attacks in Afghanistan during the 21st century, taken from the widely studied RAND database of Worldwide Terrorism Incidents. The number of fatalities has been plotted over time. Again, there are clear increases over time in both the frequency and magnitude of the attacks over the period, with both variables seeming to violate the assumptions of classic EVT.

This chapter develops a novel approach to extreme value theory that is suitable for making predictions about future extreme events in situations where the exceedances are nonstationary in both the time and mark domain. Specifically, the self-exciting Hawkes process is shown to be a good model for the time-domain which can reproduce clustering patterns such as those seen in Figure 4.1. A previous frequentist application of the Hawkes process to EVT was carried out by Chavez-Demoulin et al. (2005), however their approach is limited by the need to make strong parametric assumptions about the form of the Hawkes kernel, specifically that extreme values occur in a similar man-



(a) Daily log returns of the S&P 500 stock index. The red points and line indicate decreases larger than 3%.



(b) Terror attacks in Afghanistan from the RAND Database of Worldwide Terrorism Incidents (RDWTI). The dotted line indicates the threshold and triangular points are those attacks greater than the threshold.

Figure 4.1: Examples of extreme events occurring over a threshold.

ner to earthquakes with a power law kernel. Instead, nonparametric Bayesian techniques are used to avoid the need for these strong assumptions and the form of the kernel is learnt from the data rather than being prespecified.

The Hawkes formulation allows for self-excitation in the arrival events compared to using a regression model for the process intensity. The self-excitation can then be used as a mechanism to describe how clusters have formed in the event arrival times. These clusters are also used to show how hierarchical Bayesian modelling can be used to handle nonstationarity in the mark distribution in a natural manner, with the clusters produced by the Hawkes process assigned different parameters of the GPD distribution, with hierarchical pooling used to allow more accurate inference.

The literature is discussed in Section 4.1 before reviewing the traditional methods of EVT for estimating $p(r_i > z)$ when the event process is stationary in Section 4.2. Section 4.3 builds on the Hawkes process from Chapter 2

and show how it can be used with nonparametric methods which is suitable for when there is no strong theoretical motivation for particular parametric assumptions. In Section 4.4 the Bayesian algorithm for sampling the full posterior distribution of our model parameters is developed and explained. Synthetic data is then used to demonstrate the posterior sampling in Section 4.5. Finally, the new framework is applied to a dataset of terror attacks in Section 4.6

4.1 Literature Review

The time varying nature of extreme events in real world data has lead to numerous different approaches to modelling this nonstationarity. An early influential paper suggested a parametric regression framework for the GPD parameters to allow variation over time (Davison and Smith, 1990) and this idea has been extended in several ways (Northrop and Jonathan, 2011; Chavez-Demoulin and Davison, 2005). Related work has proposed various more sophisticated models for the point process governing the occurrence of extremes, for example (Gyarmati-Szabo, 2011). A partial review of the literature can be found in Coles (2001).

As mentioned earlier Hawkes processes have been used for the extreme values in financial time series both on the daily time scale (Chavez-Demoulin et al., 2005) and on a high frequency intraday time scale (Chavez-Demoulin and McGill, 2012). In both cases where the occurrence over the threshold of the extreme event is modeled using the ETAS form of the Hawkes process, then using the conditional intensity function a risk metric is updated based on the last extreme event. For multivariate extreme values, (Grothe et al., 2014) use a two dimensional Hawkes process to study the clustering and dependence of returns in US and European stock markets.

Hierarchical extreme value models have also been used to model the effects of climate change. Cooley et al. (2006) built such a model to study the effect of the 'Little Ice Age' that was well recorded and observed in Europe. From

this model they wished to understand whether the subsequent effects of this weather were also felt in South America. A hierarchical model is useful in this case as the information from a particular set of observations is able to influence a different sample were such data may be less informative. This is the pooling effect in hierarchical modelling and produced a better model for the data in studying the effects of climate changes geological effects.

Extreme value theory has been applied to financial markets in Gilli and Këllezi (2006) they introduce and apply extreme value theory to the extreme values found in the Credit Suisse General index from 1969 to 1999. They use the sample mean excess function to select the appropriate threshold before using maximum likelihood estimation to fit the GPD to the extreme values. This is a similar approach used in this thesis but instead the models in this work are fitted in a Bayesian manner and developed further than just a singular GPD combined with a Hawkes process for the time component.

A recent series of innovative papers (Kottas and Sansó, 2007; Wang et al., 2011; Kottas et al., 2012) introduced a novel Bayesian approach for modelling the occurrence of extreme values in nonstationary series. Their insight was that the pairs (t_i^*, r_i^*) can be viewed as observations from a bivariate Poisson process, and that the intensity function associated with this process can be estimated using Bayesian nonparametric methods based on the Dirichlet process. The flexibility of nonparametric estimation allows nonstationary behaviour in both the time and mark domains to be easily handled. However while their framework is well-suited to modelling historical data, it is less useful for making predictions about the occurrence of extremes in the future. This is because their point process representation effectively smooths out the historical data rather than explicitly modelling the conditional intensity function of the point process, which makes it difficult to make predictions based on recent process behaviour.

4.2 Extreme Value Theory

Suppose that $r_1, \ldots, r_m \sim F$ are a sequence of independent and identically distributed observations, and that interest lies in the probability of large values occurring. If the functional form of F is known, then this can be computed directly after any unknown parameters have been estimated. However the functional form of F is usually unknown, and a particular parametric form will have to be chosen based on both the observed data and theoretical considerations. Unfortunately, inference for extreme quantiles of F is known to be highly sensitive to these parametric assumptions (White, 1982).

To avoid making parametric assumptions, it is common to instead use the peaks-over-threshold (POT) approach which models only the distribution $p(r_i \mid r_i > z)$ rather than $p(r_i)$, where z is a threshold parameter chosen prior to modelling. This approach is justified by the Pickands–Balkema–de Haan theorem which essentially states that regardless of the unknown form of F, 'most' probability distributions look the same as long as z is sufficiently large. More formally:

Pickands–Balkema–de Haan (PBH) Theorem: Suppose $r_1, \ldots, r_m$ are i.i.d with distribution F. Let $F_z(y) = p(r_i - z \leq y \mid r_i > z)$ denote the conditional excess distribution function which describes the behaviour of F above a given threshold z. Then, assuming that F satisfies certain regularity conditions, F_z converges to the Generalised Pareto Distribution (GPD), i.e. $F_z(y) \to G(y \mid \alpha, \xi) \quad \text{as } u \to \infty$ where:

$$G(y \mid \alpha, \xi) = \left\{ \begin{array}{rl} 1 - (1 + \alpha y/\xi)_+^{-1/\alpha} & \text{if } \alpha \neq 0 \\ 1 - e^{-y/\xi} & \text{if } \alpha = 0, \end{array} \right\}$$

where $y > 0$ (Balkema and de Haan, 1974). The regularity conditions here essentially require that F can be normalised, and are satisfied by most non-pathological distributions. Assuming they are satisfied, the POT approach to extreme value estimation is to choose a threshold z sufficiently large to make the GPD a good approximation for F, estimate the GPD parameters (α, ξ),

and then approximate $p(r_i > z \mid r_i > u)$ by $1 - G(r - z \mid \alpha, \xi)$.

The PBH theorem is closely related to another theorem in EVT which states that the occurrence of extreme values can be treated as observations from a two-dimensional Poisson process, where one dimension denotes time, while the other represents the marks. First, write the original data as ordered pairs (t_i, r_i) for $i = 1, 2, \ldots, m$. Next, delete the pairs where $r_i < z$. Suppose n pairs remain, and for notational convenience write these as (t_i^*, r_i^*) for $i = 1, \ldots, n$ where r_i^* denotes an extreme value. Then, $\{(t_i^*, r_i^*)\}$ can be viewed as observations from a Poisson process with intensity function:

$$\frac{1}{\xi}\left(1 + \alpha\frac{r^*}{\xi}\right)^{-1/\alpha - 1},$$

note that this process is stationary in the time-domain so that extreme values are equally likely to occur at any point in the sequence, while the nonstationarity reflects the fact that the exceedances follow the GPD. As such, an alternative way to view this result is to consider the observations $\{(t_i^*, r_i^*)\}$ as being drawn from a homogenous **marked** point process. In this case, the exceedance times $t_1, \ldots, t_n$ are governed by a homogenous Poisson process, while the marks r_i^* are i.i.d draws from a GPD(α, ξ).

4.3 Nonstationarity of The Exceedance Process

For the rest of this chapter, we drop the * denoting an extreme event for notional convenience and thus extreme events are now defined as (t_i, r_i) given that r_i is larger than some threshold z.

In many applications the i.i.d assumptions on the marks $r_1, \ldots, r_n$ above the threshold that are required by both the PBH theorem and the above point process representation will not be satisfied. This can occur for two reasons:

1. The point process governing the times at which the exceedances occur can be nonstationary. This can be seen in Figures 4.1a and 4.1b discussed earlier where the exceedances fall into clusters, with no exceedances occurring for long periods of time followed by many occurring close together.

2. The distribution $p(r)$ of the exceedances may also change over time. This can be seen in Figure 4.1a where it appears that the extreme values observed during the 2008 financial crisis are systematically higher than those during the previous two decades.

There is also previous work that relaxes the need for the exceedance occurrences to be i.i.d. Given that the exceedances satisfy some mixing condition (Leadbetter, 1976), they are no longer required to be independent and instead can display some local dependance. This local dependance is expressed through the extremal index θ (Hsing et al., 1988) and can be interpreted as the average clustering effect in the extreme values.

Much of the existing literature relies on specifying parametric models for the time-evolution of both processes. Although this is a reasonable approach for modelling historical nonstationarity, it typically does not allow for inference of the **conditional** exceedance distribution $p(r_t \mid r_t > z, r_{1:(t-1)})$ which will often be the main object of interest. In many applications it will be important to assess the probability of an extreme value occurring at some particular time point t (e.g. “next week") which requires taking into account the conditional history of the process.

In this chapter a different approach is taken which focuses on modelling the conditional exceedance distribution directly. This is based on the point process representation for the marked exceedance process (t_i, r_i). However, rather than treating this process as homogenous in the time-domain with a constant mark distribution, instead a representation that allows for conditional nonstationarity in both domains is used. Specifically, (t_i, r_i) are modelled as a marked process with the following intensity function:

$$\lambda(t_i, r_i) = \lambda(t_i)\lambda(r_i \mid t_i),$$

where again r_i denotes the set of marks which have magnitude higher than the threshold z. The intensity $\lambda(t_i)$ will be modelled as a Hawkes process which allows for nonstationary and clustered behaviour to arise directly. The mark

intensity $\lambda(r_i \mid t_i)$ will be modelled using a GPD with different parameter specifications for each cluster.

4.3.1 Hawkes Process

The background outlined in Chapter 2 is built upon by taking the standard Hawkes process (Eq. (2.2)) and extending the kernel to take a nonparametric form using a Dirichlet process. This will allow the model to learn from the data, rather than prespecifying and potentially miss-specifying the form of the kernel. For notional purposes, the kernel $g(t)$ in Eq. (2.2) is relabelled as $h(t)$.

A Nonparametric Kernel

In previous applications of the Hawkes process, the kernel of the Hawkes process $h(t)$ has been specified parametrically and fitted using frequentist techniques (Porter and White, 2012; Balderama et al., 2012). However in most realistic applications it will not be obvious which sort of parametric form is most appropriate. As such, there has been recent interest in nonparametric specifications of the kernel function, typically within a frequentist framework (Mohler, 2013; Bacry et al., 2012; Lallouache and Challet, 2016).

The details of using a Dirichlet process as the basis for a nonparametric model have been previously outlined in Chapter 3. This same type of model is used to extend the kernel nonparametrically.

The general form of the Hawkes process, Equation (2.2), is modified to use a Dirichlet process as a prior to model $h(t)$. Specifically, as a mixture of some probability distribution $k(. \mid \phi)$ with parameters ϕ;

$$\begin{aligned} h(t) &= \int k(t \mid \phi) \mathrm{d}G(\phi), \\ \phi &\sim G, \\ G &\sim \mathrm{DP}(\alpha, G_0), \end{aligned}$$

where G_0 is the base distribution of the Dirichlet process and α the concentration parameter. By using a mixture of distributions the shape of $h(t)$ is flexible and adaptable to the data. Features such as heavier tails and multiple peaks

can emerge from the data that a parametric kernel would miss. Furthermore, the Dirichlet process can revert to a single mixture component if necessary.

Two forms of k will be investigated; the exponential distribution and lognormal distribution. Exponential kernels are commonly used in Hawkes process models, both parametrically and as finite mixture models (Lallouache and Challet, 2016). This previous work is extended to develop an infinite mixture model that learns the most suitable amount of components. The exponential has one free parameter $\phi = \beta$ and probability density function $k(t \mid \phi) = \beta \exp(-\beta t)$, with mean β. For the base measure G_0 the gamma distribution is used $G_0 = \text{Gamma}(\beta \mid \alpha_0, \beta_0)$, where α_0 and β_0 are the prior parameters. The gamma distribution is a conjugate prior for the exponential distribution used as the mixture kernel.

The lognormal distribution has two free parameters $\phi = \{\mu, \sigma^2\}$ and density function $k(y \mid \phi) = \frac{1}{\sqrt{2\pi}\sigma t} \exp\left(-\frac{(\log y - \mu)^2}{2\sigma^2}\right)$, with mean of $\exp(\mu + \frac{1}{2}\sigma^2)$. The lognormal distribution is a good choice for mixing kernel as using the transformation $y' = \log(y)$ means that the data can be modelled using a mixture of Gaussian distributions and thus a conjugate prior distribution can be used. The base measure choice exploits this with a normal distribution on μ and inverse gamma distribution on σ: $G_0 = N(\mu \mid \mu_0, \frac{\sigma^2}{k_0})\text{Inv-Gamma}(\sigma^2 \mid \alpha_0, \beta_0)$, where the prior parameters $\{\mu_0 = 0, k_0 = 1, \alpha_0 = 1, \beta_0 = 1\}$ are set to be uninformative.

As both the exponential and lognormal mixture models have conjugate base measures Algorithm 4 of Neal (2000) is used when performing posterior sampling. As explained in Chapter 3, this sampling will be handled by the `dirichletprocess` package.

4.3.2 Extreme Value Theory and the Generalised Pareto Distribution

Extreme events consist of a pair of values (t_i, r_i) where t_i is the time at which at the ith extreme event occurs and $r_i > z$ is the magnitude of the events that are above the given threshold z. The Hawkes process is used to model the

occurrence of the extreme events, i.e. the t_i value. The dynamic model for the magnitudes r_i is now discussed. In the simplest application of the PBH theorem it can be assumed that for some threshold z

$$p(r_i \mid t_i, r_i > z) \sim \text{GPD}(\alpha, \xi),$$

with i.i.d r_i. The parameters α, ξ can be estimated and used to compute the future probability of large r_i occurring. However, this assumes that the distribution of the r_i values is constant over time, which is not the case in many real world applications. In the traditional extreme value literature these problems are overcome by allowing the parameters of the GPD to depend on time and applying common regression estimation techniques (Coles, 2001). However, there are a number of problems with this approach. Firstly, one must assume a particular parametric form for the time dependence, which can lead to under fitting or over fitting depending on the number of parameters chosen. Secondly, this does not provide a suitable method for accounting for the clusters in the data. Instead, a new method is proposed that groups the data into appropriate clusters and fits a GPD to each cluster individually. This hierarchical model is then able to pool the individual data points across the different clusters and thus allows the time varying nature of the r_i values to be accounted for.

Recall from Section 2.2.2 that sampling the exceedance times t_i from the Hawkes process naturally produces clusters of events, which are represented by the branching variables $\mathbf{B} = (B_1, \ldots, B_m)$, where $B_i = j$ if (t_i, r_i) was generated by the Poisson process spawned by the event that occurred at time t_j, and $B_i = 0$ if (t_i, r_i) was generated by the background process. Consider the set of background events, i.e those for which $B_i = 0$. Since these events are produced by the background process μ, the formation of clusters is governed by a Poisson process.

For the magnitude of the extreme values we propose two different clustering models:

1. Each event is classified as a background or child event based on their

corresponding parent label B_i,

2. A new cluster is formed with each background event.

For the first model two clusters are defined mathematically as

$$\begin{aligned} C_{\text{background}} &= \{r_i : B_i = 0\}, \\ C_{\text{child}} &= \{r_i : B_i \neq 0\}, \end{aligned} \tag{4.1}$$

this model is referred to as the 'background and child' model. It allows for different size of events depending on whether the event is a background event or a child event and would describe behaviour where the background events are always larger than the child events. Essentially, this model is learning two different distributions for the different types of events.

For the second model, events are separated into clusters as follows. First, for each event t_j where $B_j = 0$ define the corresponding cluster C_j to be the set of events that occurred between t_j and the next background event. More formally, suppose $t_{(j+1)}$ is the next background event after t_j. Then:

$$C_j = \{r_i : t_j \leq t_i \leq t_{(j+1)}\}. \tag{4.2}$$

For both models a structure in which each cluster has its own GPD with distinct parameters ξ_j, α_j is used,

$$\begin{aligned} r &\sim \text{GPD}(\alpha, \xi), \\ \alpha &= \alpha_j \quad \text{if} \quad r_i \in C_j, \\ \xi &= \xi_j \quad \text{if} \quad r_i \in C_j, \end{aligned}$$

where the clusters are assumed to be exchangeable. The models no longer requires the r_is to be from a stationary distribution and also does not require an assumption of the time dependence of r_is. It is possible that some clusters consist of just one observation which would reduce the accuracy of the estimates of the parameters for that cluster. The introduction of this clustering scheme exchanges simplicity in the model for flexibility to adapt to the data. Fortunately this trade-off can be remedied by using a Bayesian hierarchical

model which allows for partial pooling across the clusters. Smaller clusters are no longer treated individually and will be influenced by the prior distribution that is fit from the pooling of all the data. The full hierarchical model with priors can be written as

$$\begin{aligned} \xi_j &\sim \text{Gamma}(a_\xi, b_\xi), \quad \alpha_j \sim \text{Gamma}(a_\alpha, b_\alpha), \\ a_\xi &\sim \text{Gamma}(2,2), \quad b_\xi \sim \text{Gamma}(2,2), \\ a_\alpha &\sim \text{Gamma}(2,2), \quad b_\alpha \sim \text{Gamma}(2,2). \end{aligned} \tag{4.3}$$

This use of hyper-priors allows for the clusters with fewer observations to be influenced by the larger clusters and each set of cluster parameters is drawn from a common distribution but with enough flexibility to account for the changes between clusters. The Gamma$(2,2)$ distribution is chosen to ensure the stability of the posterior inference and any distribution with larger variance causes the posterior sampling to diverge.

4.4 Posterior Inference

Posterior inference for the model involves sequentially sampling the parameters $\Theta_{HP} = \{\lambda_0, \kappa, h(t), \mathbf{B}\}$ of the Hawkes process which governs the extreme value occurrence times $t_1, \ldots$, and the parameters $\Theta_{GPD} = \{\alpha_1, \xi_1, \alpha_2, \xi_2, \ldots\}$ which govern the distribution of the extremes r_i, along with all associated hyperparameters. This separation of the model into two separate inference schemes allows for a more convenient computational approach however, the parameters of the GPD could easily be sampled alongside the parameters of the Hawkes process with no changes in the results.

In this application the background rate, λ_0, is assumed constant, in the next chapter this assumption is relaxed to allow for nonstationary background rates to which can account for seasonality in the occurrence of events.

4.4.1 Sampling for the Hawkes Process

By interpreting the Hawkes process as a branching process a computationally efficient posterior sampling method is developed. At any time t the Hawkes

intensity Eq. (2.2) can be viewed as a superposition of multiple Poisson processes; a homogeneous Poisson process with intensity $\mu(t) = \frac{\lambda_0}{T}$ and multiple inhomogeneous Poisson processes (one for each previous event that has occurred $t_i < t$). For each event t_i the intensity of this inhomogeneous Poisson process is $\kappa h(t - t_i)$. Therefore for each event, if it can be identified which Poisson process was responsible then the parameters of its generating intensity can be inferred. This information is contained in the latent branching structure $\mathbf{B}$ which allow the process to be decomposed into the events that are from the background and the events that are caused by other previous events. Therefore by using this latent variable the event times can be partitioned into appropriate sets $S_0, \ldots, S_n$

$$S_j = \{t_i; B_i = j\}, \quad 0 \leq j < n, \tag{4.4}$$

where the set S_0 contains all the events that are caused from the background intensity rate $\frac{\lambda_0}{T}$ and S_i where contains the children of event t_i which were hence generated by a Poisson process with intensity $\kappa h(t - t_i)$. The likelihood for a Poisson process with parameters θ and intensity $\lambda(t)$ is

$$p(t_1, \ldots, t_n \mid \theta) = \prod_{i=1}^{n} \lambda(t_i \mid \theta) \exp\left(- \int_0^T \lambda(z \mid \theta) \mathrm{d}z \right). \tag{4.5}$$

Due to the high level of correlation between the parameters of the Hawkes process, standard MCMC techniques would prove difficult when using the posterior from Eq. (4.5). Instead, by introducing the branching structure variable $\mathbf{B}$ the likelihood of a Hawkes function can be written as the combination of both the background intensity and the child intensities after conditioning on the latent variable $\mathbf{B}$:

$$p(t_1, \ldots, t_n \mid \theta, \mathbf{B}) = \exp\left(-\lambda_0\right) \lambda_0^{|S_0|} \prod_{j=1}^{n} \left(\exp\left(-\kappa H(T - t_j)\right) \kappa^{|S_j|} \prod_{t_i \in S_j} h(t_i - t_j) \right), \tag{4.6}$$

where $H(z) = \int_0^z h(t)\mathrm{d}t$. As $h(t)$ is a normalised probability density $H(z)$ is the cumulative distribution of $h(t)$. If $T >> t_j \forall j$ then $H(T - t_j) \to 1$ and the coupling between the kernel $h(t)$ and κ is removed. This allows separation of

the likelihood and thus all three components of the Hawkes process $\lambda_0, \kappa, h(t)$ become independent of each other. By exploiting this conditional independence an efficient MCMC sampler for the parameters of the Hawkes process can be built.

The unknown values of the Hawkes model $\Theta_{HP} = \{\lambda_0, \kappa, h(t), \mathbf{B}\}$ are to be inferred. In all models in this chapter the background rate is assumed to be $\mu(t) = \frac{\lambda_0}{T}$ and constant. This can be easily extended without any loss of generality and will be demonstrated in Chapter 5. To sample from the posterior distribution of these parameters each parameter will be simulated from its full conditional distribution, as in the Gibbs sampler.

Recall that each event t_i has an associated B_i that indicates its parent. A value of $B_i = 0$ is indicating that the event was generated by the background rate and $B_i = j$ indicates that event t_j is the parent of event t_i. For the ith event occurring at time t_i, the (unnormalised) probability of its origin is (Ross, 2019)

$$\begin{aligned} \Pr(B_i = 0 \mid \lambda_0, \kappa, h(t), t_i, H_t) &= \frac{\frac{\lambda_0}{T}}{\lambda(t_i \mid H_t)}, \\ \Pr(B_i = j \mid \lambda_0, \kappa, h(t), t_i, H_t) &= \frac{\kappa h(t_i - t_j)}{\lambda(t_i \mid H_t)}, \quad j = \{1, 2, \ldots, i-1\}. \end{aligned} \tag{4.7}$$

These distributions can be sampled from exactly once calculated. This process allows full simulation of the branching structure of a Hawkes process from the exact posterior using values of the parameters λ_0, κ and $h(t)$. Then conditional on this branching structure, the other parameters of the model can be simulated. Conditional on $\mathbf{B}$ the sets S_i can be formed as defined in Eq. (4.4) and used to estimate the other parameters of the Hawkes model.

The number of events with $B_i = 0$ allows for a posterior sample of λ_0 using the standard sampling procedure for a homogenous Poisson process. These events are caused from the background rate λ_0 and distributed as a Poisson process as shown from the full likelihood in Equation (4.6)

$$S_0 \mid \mathbf{B} \sim \text{Poisson}(\lambda_0), \tag{4.8}$$

a conjugate Gamma(a_μ, b_μ) prior is used for λ_0 as this allows for direct samples from the posterior distribution

$$p(\lambda_0 \mid S_0, a_\mu, b_\mu, \mathbf{B}) \sim \text{Gamma}(a_\mu + \mid S_0 \mid, b_\mu + 1). \tag{4.9}$$

The expected number of children events for each event is controlled by κ. Again, using the full likelihood (Eq. 4.6) it has been shown that the number of children events from each event $S_j, j > 0$ is Poisson distributed

$$S_j \mid \mathbf{B} \sim \text{Poisson}(\kappa) \quad j > 0, \tag{4.10}$$

the value of κ can be inferred by using a conjugate gamma$(a_\kappa b_\kappa)$ distribution and sampling from the posterior distribution

$$p(\kappa \mid \mathbf{S}, a_\kappa, b_\kappa, \mathbf{B}) = \text{Gamma}\left(a_\kappa + \sum_{i=1}^{n} S_i, n + b_\kappa\right). \tag{4.11}$$

Sampling the parameters of the kernel depends on the specification of $h(t)$. For notational convenience write $\tau_j = t_j - t_{\text{parent}}$ as the rescaled event times for all events with $B_i \neq 0$. The values of τ_j are then used to sample the kernel posterior distribution. For a parametric kernel, a prior distribution for the parameters is specified and then used to sample from the posterior distribution using the appropriate method. For an exponential or lognormal kernel there are conjugate prior choices that allow direct sampling of the posterior.

To obtain posterior samples for the nonparametric kernel Algorithm 4 specified by Neal (2000) is used. The algorithm uses the Polya Urn representation of the Dirichlet process to assign the τ_js into their appropriate mixture components and then update the components appropriately. The stick-breaking construction of the Dirichlet process is used to sample from the posterior distribution of the fitted τ_js. It is this posterior draw that functions as the kernel for the next round of sampling. The parameter α of the Dirichlet process has a gamma prior (West, 1992) and sampled for each iteration of kernel fitting. The overall sampling of the Dirichlet process is achieved using the `dirichletprocess` package in R (Markwick and Ross, 2018) as detailed in Chapter 3.

Combing the structure simulations and posterior parameter samples leads to the full Gibbs sampling algorithm:

Algorithm 1: Sampling the parameters of the Hawkes model.

Data: Event times $t_1 \dots t_n$

Result: S posterior samples of λ_0^s, κ^s, kernel $h^s(t)$ and parent structure $\mathbf{B}^s$.

Chose starting values λ_0^1, κ^1 and $h^1(t)$;

for *i = 1 to S* **do**

- Sample the new parent structure $\mathbf{B}^{i+1}$ from probabilities calculated using Eq. (4.7) and λ_0^i, κ^i, kernel $h^i(t)$;
- Calculate S_j from $\mathbf{B}^{i+1}$ using (4.4) ;
- Sample λ_0^{i+1} using Eq. (4.9) ;
- Sample κ^{i+1} using Eq. (4.11) ;
- Sample $h(t)^{i+1}$ using the `dirichletprocess` R package ;

end

By exploiting the conditional independence of $\mathbf{B}$ and the explicit parameters of the Hawkes model λ_0, κ and $h(t)$ MCMC chains of the parameters conditional on the data can be built.

4.4.2 Sampling for the GPD

The parameters Θ_{GPD} are sampled from their full conditional distributions. Conditional on the branching structure $\mathbf{B}$, the extremes $r_1, \dots, r_n$ are divided into clusters based on Equation (4.2). Suppose there are K such clusters in a given iteration of the Gibbs sampler. Each cluster has a unique value for the GPD parameters (α_j, ξ_j) for $j \in 1, \dots, K$. Each of these K parameter sets are independent of the others, so the sampling procedure is equivalent to performing K completely separate sets of inference for a GPD, given independent parameters and observations. For each of the parameter sets, this sampling is performed by using Hamiltonian Monte Carlo as implemented in the Stan programming language (Carpenter et al., 2016). As such, the algorithm for sampling the GPD parameters can be written as:

Algorithm 2: Algorithm for sampling the GPD parameters.

Data: Parent structure samples **B**

Result: S samples of Θ_{GPD}

for $i = 1$ *to* S **do**

Sample the parameters Θ_{GPD} using the Stan implementation of (4.3)

end

Overall, sampling from the posterior distribution is achieved in this split manner, firstly by sampling from the Hawkes process and then using the resulting parameter chains to sample from the GPD model.

4.5 Experiments

The above method is applied to synthetic data to verify that the correct parameters of the model can be inferred.

4.5.1 Model Checking

Since the primary goal is predicting the occurrence of extreme values in the future, model performance will be assessed by dividing the data into a training and testing set. The parameters of the model will be inferred from the training set before applying the model checking on the test set and the better fitting model will have greater predictive power on the test set. For checking the validity of the Hawkes model the Deviance Information Criteria (DIC) (Section 2.3.2) will be used [1]. This is a likelihood based calculation that penalises the predictive accuracy of a model by the complexity of such model. As the Dirichlet Hawkes models can potentially use an infinite number of parameters it is necessary to correctly penalise overfitting. Furthermore, as the models are conditional on the history of the process, cross validation methods that remove events in the time domain cannot be used. The best fitting model is one with the lowest DIC value. Hawkes processes will also be simulated from the model parameters to ensure that future forecasts are consistent with what

[1] Other information criteria could have possibly been used such as the Bayesian information criteria.

is actually observed.

For the GPD part of the model leave-one-out cross validation (LOOCV) will be performed using the `loo` R package (Vehtari et al., 2017). LOOCV is a method for assessing the pointwise out-of-sample predictive accuracy of a model and outputs a metric leave-one-out information criteria (LOOIC). It is computed from the log-likelihood evaluated using the posterior samples of the parameter values and like the DIC the model with lowest LOOIC value is preferred.

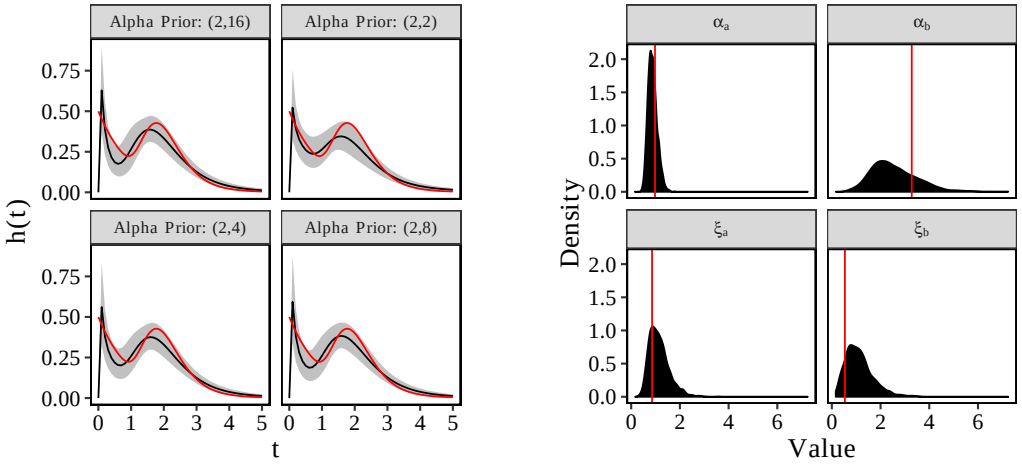
4.5.2 Synthetic Data

To demonstrate the outlined methodology a synthetic data set from a Hawkes process with a complex kernel is simulated and used to show that the new method is able to extract the correct kernel shape. A data set with $\lambda_0 = 0.01$, $\kappa = 0.8$, $h(t) = \frac{1}{2}\text{Log-normal}(0.7, 0.3) + \frac{1}{2}\text{Exp}(1)$ and $T = 5000$ is simulated from to form the synthetic data. This resulted in 319 events.

A Hawkes process with Dirichlet mixture of the lognormal distribution kernel is fitted to the data and the resulting kernel is compared to the true kernel function. To check for prior sensitivity, four different priors on the concentration parameter α are compared. A parametric lognormal kernel is also fitted to the data to enable comparisons between the Dirichlet and single kernel models.

The resulting posterior mean and credible intervals of the kernel is shown in Figure 4.2a. The Dirichlet mixture model has correctly adapted to the more complex true kernel shape. Figure 4.2a also shows very little prior sensitivity to the Dirichlet process parameter α .

The Kullback-Leibler (KL) divergence for the Dirichlet models and a Hawkes model with a lognormal kernel can also be calculated as the true kernel shape is known. All four of the Dirichlet models have an average KL divergence of 0.06, where as the model with singular lognormal kernel has an average value of 1. This shows that the Dirichlet kernel mixtures are correctly identifying the true kernel and fitting to the data better than the single



(a) Resulting kernels from the synthetic data with complex kernel, each plot utilises a different prior parameter for α of the Dirichlet process. The true kernel shape is shown with the solid red line. The posterior mean is the solid black line with 95% credible intervals as the grey interval.

(b) Hyper parameter posterior samples of the GPD model. The solid vertical line indicates the true value.

Figure 4.2: Results from fitting the model to a synthetic dataset.

lognormal kernel.

For the GPD synthetic data, the clustering of the events is used to group them appropriately as per Eq. (4.2). Then individual α_j, ξ_j values are drawn from gamma distributions as per Eq. (4.3) with $a_\xi, b_\xi, a_\alpha, b_\alpha = 2$ and then subsequent values are drawn from the GPD with these cluster parameters for each data point. Using the model in Eq. (4.3) the unknown parameters are inferred using the synthetic data. In Figure 4.2b the density of the hyperparameter samples are within the distribution of the posterior values and closely centered around the true value for three out of the four parameters as expected.

4.6 Application

Given that the fitting procedure has been verified on synthetic data it can now be used on real data. The Hawkes and GPD model is used to study the self-exciting nature of extreme terror attacks. For the data source the RAND Database of Worldwide Terrorism Incidents (RDWTI) [2] is used which is a publicly available record of all terrorist attacks from January 1968 to December 2009. The terrorist attacks that occurred in Iraq from 20th of March 2003 onwards are considered as this is the official start date of US led invasion of Iraq. For each terrorist attack in the database only the day of the event is recorded. Therefore, on days where multiple attacks occurred the ordering is arbitrary. To overcome this issue, a random uniformly drawn number between 0 and 1 is added to the time of each of the attacks, this ensures that days with multiple attacks have unique time stamps. The modelling of terror attack fatalities has been previously studied in Porter and White (2012).

The threshold level is set at fatalities > 10 which was consistent for EVT using the mean over threshold theory (Coles, 2001). This subsample consists of 5% of all the attacks and contained 490 separate events. This is the dataset used to infer the parameters of the Hawkes and GPD models as per Section 4.4.

Four different Hawkes models are considered, two with parametric kernels and two with Dirichlet mixture kernels. For the parametric kernels the exponential and lognormal distribution are used and for the Dirichlet kernels infinite mixtures of exponentials and lognormal distributions. The prior on α is set as Gamma$(2, 4)$ as from the synthetic analysis little prior sensitivity was shown in Figure 4.2a.

The first 70% of the attacks are used as the training set with the remaining 30% used as the test set to evaluate model performance. The sampling algorithm is run for two chains of 5000 iterations, discarding the first 2500 iterations as burnin which is enough for the parameter samples to be well mixed.

[2] https://www.rand.org/nsrd/projects/terrorism-incidents/download.html

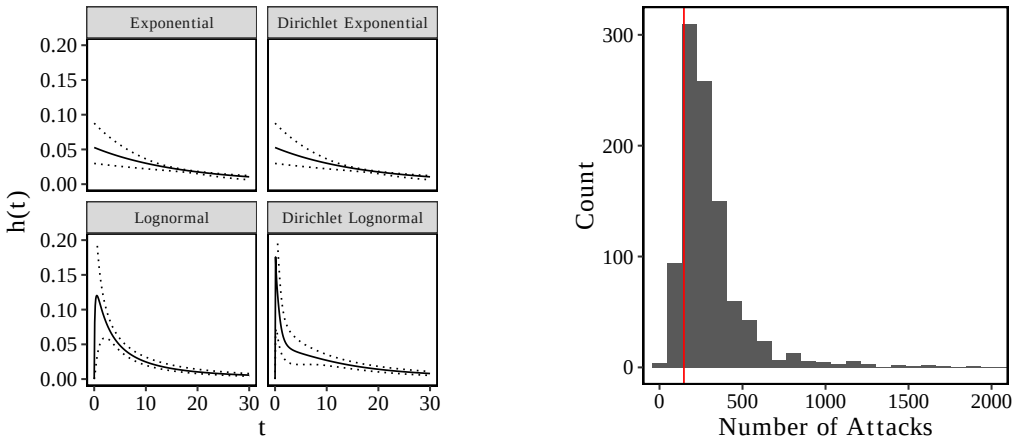
Table 4.1: Posterior means and standard deviations for the Hawkes parameter samples fitted on the terror attack data.

Model	λ_0	s_{λ_0}	κ	s_κ	DIC
Exponential	0.034	0.016	0.86	0.080	753
Lognormal	0.047	0.019	0.81	0.089	755
Dirichlet Exponential	0.032	0.014	0.87	0.074	752
Dirichlet Lognormal	0.033	0.014	0.86	0.075	745

Table 4.1 shows that both models have converged to similar parameter values for both λ_0 and κ. A κ value of 0.86 indicates that there is a high amount of self-excitation from these extreme terror attacks, on average every 1.16 ($\frac{1}{\kappa}$) attacks will see another child attack. Figure 4.3a shows the resulting kernel shapes fitted from the data and the Dirichlet lognormal kernels have a different shape to the single lognormal kernel. There is a fast decay before plateauing out which shows that the Dirichlet lognormal mixture model has found new structure in the kernel. As the time scale is in days, Figure 4.3a shows that the majority of self-excitation effect dies out 10 days after the initial attack with further continual decay until approximately 30 days after the attack. The main difference between the Dirichlet lognormal kernel and others is both the presence of the sharp increase after the initial attack followed by the power-law like decay. This combination of features cannot be achieved by the single lognormal kernel and the Dirichlet exponential kernel does not have the flexibility to provide both the peak and slow decay. Therefore this improvement in the DIC has a physical interpretation evident from comparing the kernel shapes.

From the DIC values the Dirichlet lognormal model is the best fitting model. So whilst there is potentially an infinite amount of parameters in this model, it has shown that the benefit from these parameters outweighs the increased complexity. Furthermore, as this was fitted on the test set it shows a

real predictive benefit in using the Dirichlet lognormal model over the simpler parametric models.



(a) Kernel shapes for the terrorism data. The solid black line indicates the posterior mean, with the dotted line indicating the 97% credible interval.

(b) Posterior predictive simulations of the number of events in the test set. Solid red vertical line shows the true number of events. The model is well calibrated in predicting the correct number of events.

Figure 4.3: Results from fitting the Hawkes model to the extreme terror attack dataset.

Figure 4.3b shows the distribution of simulated events across the posterior parameter samples. The vertical line indicates the true number of events in the test set and is shown close to the centre of the distribution. The model is correctly forecasting the number of events for the future, therefore can be seen as a valid model which can also be confirmed by the improved DIC values.

Given the time component of the terrorist attack has been modelled, the actual number of fatalities is now considered. The clustering of the Hawkes process is used to separate the terror attacks into groups; firstly using Eq. (4.1) and a further model using Eq. (4.2). The baseline model is one in which the number of fatalities of each event are i.i.d from a GPD. After fitting the three models the `loo` package is used to perform Bayesian leave-one-out cross

Table 4.2: LOOIC values for the GPD models of the number of fatalities.

Model	elpd_{loo}	p_{loo}	LOOIC
Baseline	-119.4	0.3	238.8
Background and Child	-103.4	0.9	206.8
Full Clustering	-343.5	215.4	687.1

validation on the held out test data.

Table 4.2 shows that the background and child hierarchical model is an improvement on the baseline model. The full clustering model is a very poor fit due to its large LOOIC value. The parameter p_{loo} is effectively a Bayesian estimate of the number of parameters in a model and shows the jump from the background and child model to the full cluster model has introduced over 200 new parameters without an improvement in predictive performance. However, the increase in predictive performance with modest increase in p_{loo} between the baseline model and background and child model shows that the hierarchical GPD model is an improvement in modelling the terror attack fatalities.

4.7 Discussion

This chapter has developed and applied a novel framework for modelling extreme events that relaxes many of the conditions of classic extreme value theory. I have produced a full posterior simulation algorithm for both the time component and value of extreme events, with specific contribution of the Bayesian algorithm for sampling from the Hawkes process. The model offers an interpretation of the time clustering behaviour of extreme values through the parameter κ and kernel $h(t)$ of the Hawkes process. By using a nonparametric parameterisation for $h(t)$ a greater degree of flexibility can be used in the modelling approach. A Dirichlet process mixture model has learnt from the data and recovered a kernel that has a fatter tail than parametric kernel specifications. The GPD model combined with this clustering from the

Hawkes process leads to a better predictive performance for the applied example of terrorist attacks. This is an improvement on previous approaches where predictive performance had not been possible (Kottas and Sansó, 2007; Wang et al., 2011; Kottas et al., 2012).

This framework has been applied to terrorist attacks however it is agnostic to any extreme event situation. There is scope for further work assessing its performance in a variety of fields such as financial risk, assessing the clustering nature of extreme falls in the stock market and the subsequent GPD model.

In conclusion, extreme value theory has been a good introduction to the application of a nonparametric Hawkes process. I have shown how the Bayesian Hawkes process can be combined with the Dirichlet process using the `dirichletprocess` software package of Chapter 3.

Chapter 5

Hierarchical Bayesian Modelling of FX Trade Arrivals Using the Nonparametric Hawkes Process

In the previous chapter there was only one time series of extreme events in consideration but in this chapter, the Hawkes process will be extended to multiple timeseries which in turn will involve extending the Bayesian Hawkes algorithm of Chapter 4 to account for the multiple timeseries. The nonparametric component will now be applied to the background rate, $\mu(t)$, of Equation (2.2) to model the seasonal pattern of background events throughout the day. In this chapter a financial dataset consisting of when trades were executed throughout the day will be used as the application. The general occurrence of trades is variable throughout the day and by using a nonparametric model to account for this variation avoids the need to assume a specific form of daily seasonality where instead it can be learnt from the data.

In recent history financial markets have made a steady march from physical trading pits to electronic exchanges (Rime and Schrimpf, 2013). Human traders are now less involved in the submission of trades and instead, algorithmic trading strategies are used to send trade orders to the market. This electrification has also introduced predatory actors in the market and there are now trading bots that are attempting to detect such algorithms and profit

off the pattern of trades other traders make. The distribution of order times are now going to be more complicated and each trade is unlikely to be independent of each other hence each order that arrives in the market will cause a reaction by other actors who will respond with their own order. This chapter shows how a nonparametric and hierarchical Hawkes model can be used to model the number of trades throughout the day and can account for the clustering of orders. Understanding the daily pattern of financial market data

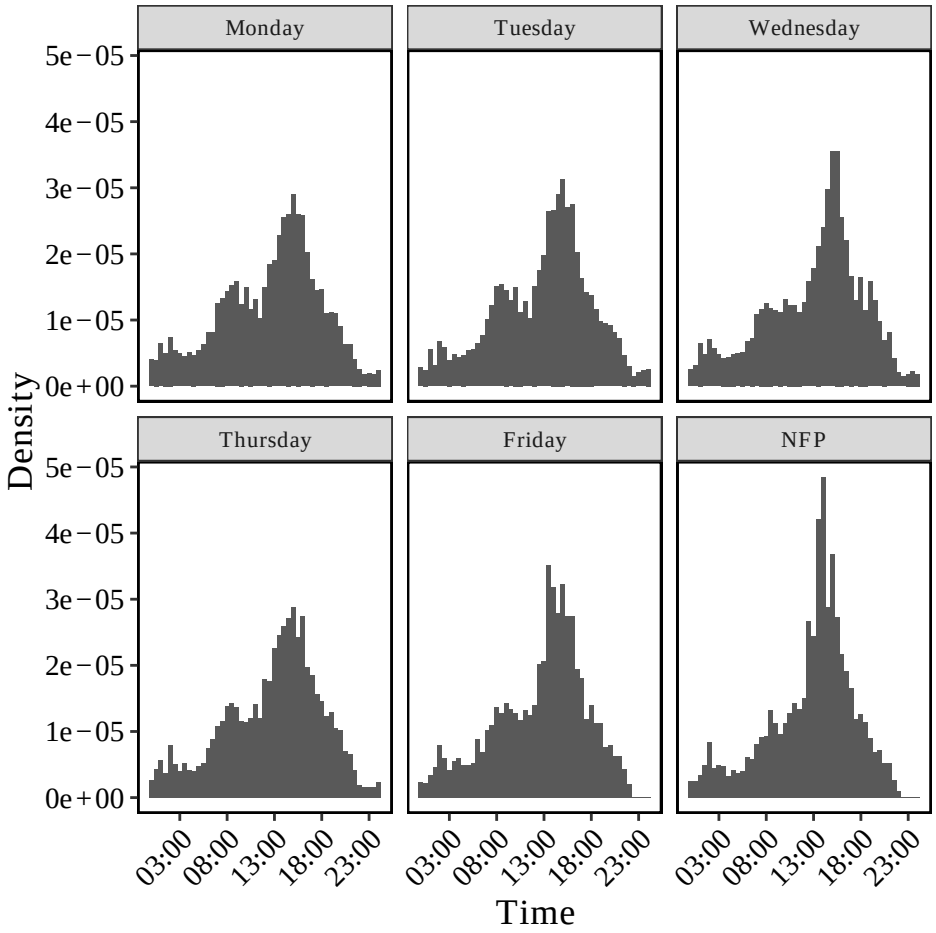


Figure 5.1: Density histogram highlighting the difference in number of trades across the weekdays and NFP days. The trades are bucketed into 30 minute groups.

brings with it a number of interesting modelling problems. The data consists of observations throughout the day (intraday) across a number of days and

it is expected to see common, reoccurring patterns - i.e. increases in activity at market opening and closing times with slight variations in this behaviour depending on the day. Figure 5.1 shows this type of pattern where there is a general increase in the number of trades around 08:00 and 14:00 that follows the opening of the European and American stock exchanges. This type of pattern is similar for each day of the week, but there are subtle differences in the peaks of the empirical distributions depending on the day of the week. For example Wednesday mornings are not as active as the other days of the week and Friday afternoons have an increased trading amount compared to other days. Furthermore, by using the Anderson-Darling test we are able to conclude that the distribution of trade times on each different day of the week in statistically different, such that this separation into different days of the week is valid.

This type of difference motivates a model where each day of the week has an individual background rate and thus a separate Hawkes process depending on the day of the week

$$\lambda_{\text{Monday}} = \mu_{\text{Monday}} + \kappa_{\text{Monday}} \sum_{t_i<t} g_{\text{Monday}}(t - t_i),$$

where each other day of the week would follow the same pattern, but with their own parameter sets, i.e. λ_{Tuesday} etc. However, modelling each day individually eliminates the possibility of the sharing data to find common features amongst the days of the week, as previously highlighted, there is a common location in the peaks that should be learnt using all the available data. A Bayesian hierarchical model, as outlined in Chapter 2, remedies this by allowing for the sharing information across days whilst retaining individual characteristics (Gelman et al., 2014).

The background component of the model will be described with a hierarchical Dirichlet process. By using such a process in a hierarchical manner the data will separated into appropriate groups and a model for each group that shares data across groups will be established. In a Dirichlet process this involves sharing clusters which has been explored with application to textual

data in Teh et al. (2005) and Zhang et al. (2010). Both works uses a hierarchical Dirichlet process to cluster the patterns in text data from different sources and by sharing the clustering across the sources both pieces of work are able to find common groups in the data. This is a similar goal for this chapter, to find similarity in diverse groups but also where the differences between the groups may lie.

The application of such a model will be focused on the times of trades in the foreign exchange (FX) market. As most financial literature typically focuses on returns or volatility in equity markets, this will be a new avenue of exploration. A point process model will be explored and extended to include both the daily seasonal patterns and clustering behaviour and this type of model will then provide an ability to forecast future seasonal behaviour and high frequency changes in market conditions from these trading patterns. In this case, seasonality refers to the diurnal pattern of market behaviour, where there is an increase of activity around stock markets opening times but periods of decreased activity outside of normal trading hours.

Such prediction of the intraday behaviour of FX markets is important for a number of reasons. Firstly, the cost of trading is directly related to the amount of liquidity at the point in time. If a trader places a large order at a time where there is a low amount of volume traded, the price of the trade will be negatively impacted and executed at a worse price. Therefore, understanding the patterns in trading behaviour is needed to obtain the best prices of an asset. Secondly, as the FX market is the largest financial market, the intraday predictions can cascade through into other markets. Asset managers buying foreign stocks must also manage their currency positions, therefore every stock trade will be followed by a FX trade to hedge the currency risk. So whilst they might not be actively trading a certain currency they can be indirectly influenced and must be aware of the behaviour of the FX market.

This will be a novel approach to point process modelling consisting of using a Hawkes process with a hierarchical Dirichlet process to learn both the

pattern of intraday trades and number of trades in each day. Using a unique dataset provided by BestX [1] the resultant model is able to predict both future numbers of trades per day and how these trades are distributed throughout the day. Once again, the approach will be explicitly Bayesian and combining the flexibility of nonparametric models with the self exciting nature of the Hawkes model.

This chapter begins by reviewing the relevant literature in Section 5.1 before exploring the available data and highlighting key features in Section 5.2. I proceed to build on and extend the mathematical details from previous chapters for both Dirichlet process and Hawkes processes in Section 5.3 before describing our full model that combines both processes. In Section 5.4 I present our inference results and analyse the fit of the models. Finally, Section 5.5 discusses these results and the impact they have on predicting the behaviour of trading a currency.

5.1 Literature Review

Financial time series modelling commonly uses autoregressive type models (Bollerslev, 1986; Engle, 1982). This framework of modelling is structured such that the future value of a time series is dependent on the previous values that have been observed. This dependance can be controlled and extended by a variety of parameters in the models and similarly, the variance in the observations can also depend on the previous values which leads to what is known as a GARCH model (Bollerslev, 1986).

Hawkes processes have also been used to model a wide variety of financial problems, notably trading activity and the clustering behaviour in markets. American future contracts trade almost constantly from Monday to Friday and as such offer a wide range of different periods of high and low trading activity. Filimonov and Sornette (2012) apply the Hawkes model to the change in mid price of E-Mini S&P 500 contracts from 1998 to 2010. By fitting the Hawkes

[1] www.bestx.co.uk

model (using maximum likelihood estimation) to 2 month intervals across the 12 years they find that the average number of branching events, i.e. events caused by other events has increased overtime from 0.3 to 0.7. This shows that the number of changes in price is more endogenous and self-excited than in the past. This has been attributed to the increase in electronic trading and rise in algorithmic trading as being able to operate quicker in markets has led to more trading around events. This shows how the parameters of a Hawkes process can quantify the changing behaviour of a phenomena and that by parameterising the Hawkes process correctly there is good interpretation of the parameters.

Similarly, currency trading is notably volatile as the 2016 EU referendum in the United Kingdom has shown. Following the result of the election, the value of the pound dropped by 10% against the dollar over the course of hours. Untangling internal and external events that drive price changes is desirable for well functioning markets and helps understand the dynamics of the various factors that control currency prices. Scheduled macroeconomic news such as GDP figures and unemployment rates of countries are examples of external events that will cause a change in volatility in the markets. Rambaldi et al. (2015) studied the effects of such news and the subsequent market reaction using the Hawkes process to model the number of market events before and after macroeconomic information was released. For their Hawkes model an additional kernel was included to account for the release and subsequent effect of news on trade intensity. Again, the model was fitted using maximum likelihood and found that the addition of a news kernel did lead to a better fit for the price changes.

There have been various other studies into volumes of trades in other financial markets. In the US equity markets, it is found that the volume of shares traded follows a power law distribution and that there are long range correlations in the total number of shares traded (Gopikrishnan et al., 2000). Similarly, Ajinkya and Jain (1989) examine the empirical distributions of daily

trading volume in stocks listed on the New York Stock Exchange and find that there is a slight correlation between consecutive days trading activity. There has been analysis into the behaviour of volume in the bond market (Alexander et al., 2000) where a linear regression is used to model the trading volume of different issues of bonds. They find a number of explanatory variables to help predict the volume traded of bonds per month. However, all three of these papers do not consider intraday prediction, they are focused with longer trends in trading rather than the finer minute to minute amount of trades. This chapter fills the gap in the literature for both the intraday modelling and application to the FX market.

There is also a variety of literature that links the price or trade intensity with volatility. In Gerhard and Hautsch (2002) they assess how the time to observe large enough price change is comparable to the volatility thus the observed rate of these price changes is directly related to the current volatility. In Taylor (2004) a similar approach is taken where they again model the time between transactions to estimate the volatility in returns. Both papers are applied to the futures market and thus show that understanding the trading intensity of a financial market can provide information on the current market conditions.

5.2 The Data

The motivation for studying the FX market is two fold. Firstly, The FX market trades on average $5.1 trillion USD a day and is the largest financial market in the world (Moore et al., 2016). However, there are no centralised exchanges where all transactions must be reported. This contrasts with equities where all trades of a stock must be reported to the exchange, such as the London Stock Exchange (LSE) or New York Stock Exchange (NYSE) depending on where the stock is listed. Secondly, there are no set hours from which you can place FX trades, instead you can trade at any point of the day regardless of your local timezone; the FX market opens late on Sunday evening, staying

open until the following Friday evening. Without daily opening and closing times, a more natural pattern of trading occurs that is closely correlated with the various stock exchange opening times around the world. For example, the currency pair EURUSD, sees large amounts of trades around the opening of the London Stock Exchange (8am GMT) and at the opening of the NYSE (0930 AM (GMT-5)) with a decrease in observed trading activity between these time. There is also a slight increase in the number of trades around midnight and the early hours of the morning (GMT) as the Asian markets start opening. It is this kind of behaviour that makes the FX market an interesting application to study and these emergent patterns must be understood by FX traders to achieve best prices for clients.

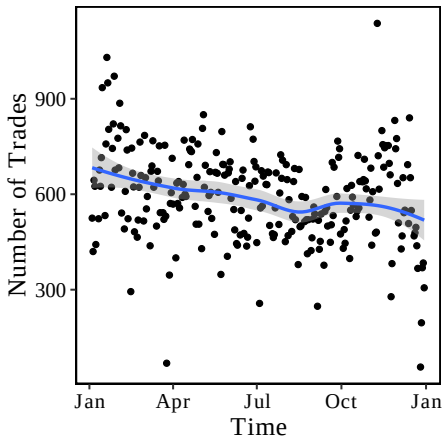
Our dataset consists of high frequency data from the 1st of January 2016 to the 31st December 2016 of the currency pair USDCAD. This single currency provides the necessary information to deduct both the self-exciting nature of trade occurrences and the seasonal patterns in the data, then using this single currency behaviour the general model could be extended to other currency pairs. As USDCAD is a liquid G10 currency, its behaviour can be naturally extended to other currency pairs, without needing to change the structure of the model and instead each currency pair would be used to obtain their own parameters. Each entry of the data contains a timestamp indicating at what time the trade occurred (rounded to the closest millisecond). As highlighted in the introduction, Figure 5.1 shows that there are subtle differences in the patterns on different weekdays. The Anderson-Darling criterion (Scholz and Stephens, 1987) is used to test whether the five samples are generated from a common function and it is found that this hypothesis is rejected. Therefore we can conclude that the intraday profile of trades for each weekday is sufficiently different for us to justify separate groups for each weekday. Furthermore, we can visually identify certain differences. Mondays and Tuesdays have a larger increase in the number of trades at roughly 08:00 (LSE opening time) and all five days have different behaviours at approximately 14:30 (NYSE opening

time). The density of trades decreases at different rates for each day after 18:00. From this it can be concluded that there is significant difference in the day to day behaviour of USDCAD. Furthermore, we would also expect other currencies to also display their own intraday differences in weekday behaviour that would be similar to USDCAD. Therefore, if this model were to be extended to other currencies, there should be sufficient flexibility in the model to allow for the variations between days of the week.

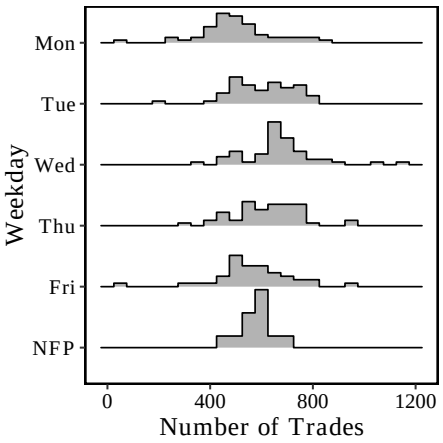
We are also interested in the market behaviour on special economic event days. As mentioned previously, the FX market is continuously open throughout the week and therefore different patterns emerge depending on the nature of the economic event. A concern of FX traders is both the pattern changes before and after the event occurs and one such event that will be used in this work is the Non Farm Payroll (NFP) day. The Non Farm Payroll Employment statistic is a comprehensive indicator of the US economy and state of the labour market. It reflects the number of jobs added or lost in the USA over the last month. The number has a large effect on the value of the dollar and therefore all currency markets feel its effect. The Non Farm Payroll number is closely watched by economist, asset managers and speculators who will react and trade based off its value immediately when it is released. The NFP number is released every 1st Friday of the month at 08:30 Eastern Time and this has a large effect on the intraday trade patterns of the majority of currencies which is shown in Figure 5.1 specifically for USDCAD. NFP days have fewer trades occurring at London open and all round fewer amount of trades. When the NFP number is released there is a massive spike in the number of trades that occur which is when the NFP value is released and reacted upon.

The NFP trading pattern is an excellent example of how daily trading patterns are similar but remain diverse. There are common features of an NFP day that is seen in the other weekdays, the London and New York opening increase but at different magnitudes. There is a large difference in behaviour but it is still resembles the other weekdays. This suggests that the data is well

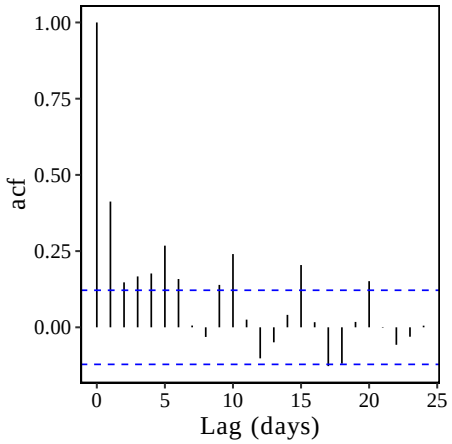
suited to a hierarchical based model.



(a) The number of trades per day with added smoothing using LOESS. The grey band indicates the standard error of the estimated smoothed line.



(b) Histograms showing the distribution of the total number of trades on each of the different days.



(c) Autocorrelation estimation of the number of trades.

Figure 5.2: Empirical trends in the total number of trades per day.

The total number of trades per day has a slight trend, there is a decline in the number of trades in the summer months that then reverse in Autumn. Figure 5.2a shows the number of trades per day with a local regression added to smooth the trend. This long term behaviour must be accounted for in the model to ensure that the total number of trades per day predicted by the

Hawkes process is accurate.

The distribution of total number of trades in each weekday is also explored. Figure 5.2b shows the distributions of the number of trades on each weekday including NFP days as separate days. Again, by using an Anderson-Darling test it is found that these distributions are statistically different and the total number of trades on each day of the week cannot be seen as being drawn from a common distribution. Mondays and Fridays generally have a lower amount of trades, whilst the other days have a larger number of trades. NFP days are less varied in the number of trades compared to the other days of the week. Therefore it can be concluded that the number of trades per day is significantly different for each day of the week and NFP day and must be accounted for in the model.

There are also autocorrelations present in the data. By computing the autocorrelation function for the number of trades there are a number of lags where the autocorrelation appears to be significant. Figure 5.2c shows that the most significant correlations occur at lags of 1 day and at multiples of 5 days. This suggests that Monday's with significant trading activity lead to the next Monday in the following week also having significant activity.

In summary, by looking at the empirical results of the dataset there have been a number features identified that are needed within the model. The model must be flexible enough to account for different intraday distributions but must also maintain consistency in peaks across all days for when the stock markets open. The variation in total trades per day of the week and the slight trend in the total number of trades must also be present in the model. Both the intraday and day-to-day predictions must be described by some model component.

5.3 The Model

From previous work in studying other financial markets (Filimonov and Sornette, 2012; Chavez-Demoulin and McGill, 2012; Rambaldi et al., 2015), it is

known that the trade arrivals are unlikely to be independent of one another. There will be clustering effects, self-exciting behaviour and other dynamics that come into play. Such effects can be modelled using a Hawkes process which as mentioned throughout, is a type of self-exciting point process with intensity function, $\lambda(t \mid H_t)$, dependent on the history of previous events.

The empirical findings from the data (Figure 5.1 and 5.2) have a number of features that must be captured: multi-modality due to the UK and USA stock markets opening, weekday differences and capturing the variance in the total number of trades per day accurately. This is achieved by using a non-parametric hierarchical model for the background rate, $\mu(t)$, of the Hawkes process intensity in Eq. (2.2).

The trade arrival times t_{di} are modelled as if they are generated from a point process. Under this notation d indexes the day and i indexes each individual trade during the day, $i = 1, \ldots, N_d$ where N_d is the total number of trades on day d. Each day consists of a window of time from midnight to midnight the next day.

The intensity function for a single day d can be written as

$$\lambda_d(t \mid H_{t_{di}}) = \mu_d(t) + \kappa_d \sum_{t_{td}<t} g_d(t - t_{t_{di}}),$$

where the background rate is decomposed into an amplitude of μ_d constant over the day d and a density $f_d(t)$ both of which depends on the day of the week

$$\mu_d(t) = \mu_d f_d(t), \tag{5.1}$$

and each day of the week has its own set of parameters.

Each day d has a variable D that indicates what day of the week d was. This variable is used to group the days into their appropriate days of the week

and thus group the background density functions

$$\begin{aligned} f_d(t) &= f_{\text{Monday}} \quad \text{if} \quad D_d = \text{Monday}, \\ f_d(t) &= f_{\text{Tuesday}} \quad \text{if} \quad D_d = \text{Tuesday}, \\ &\text{etc}, \\ f_d(t) &= f_{\text{NFP}} \quad \text{if} \quad D_d = \text{NFP}, \end{aligned}$$

therefore there are six background intensities that must be inferred, for notational convenience, these six intensities are relabelled as $f_D(t)$ where $D = 1, \dots 6$, with 1 - 5 labelling Monday to Friday and $D = 6$ labelling the NFP days. In summary the amplitude for each day μ_d controls how many background events occur on day d and the density f_d controls how they are distributed throughout the day, which depends on the day of the week. The amplitude, μ_d, will be used to capture the long-term trends and the autoregressive nature of the total number of trades on each day by including different covariates, such as indicator variables for each day of the week and autoregressive components for the previous days total number of trades. Whereas the density will describe the intraday seasonality present in the data.

A Dirichlet process mixture model is used to model the intraday behaviour of the trades. As stated in the previous chapters, a Dirichlet process mixture model provides an nonparametric method of estimating a density that requires the specification of a mixing kernel k and a base measure G_0. In this application the occurrence of trades is bounded between midnight and midnight and as such the generalised beta distribution with mean μ and scale ν is a suitable specification of k

$$k(t \mid \mu, \nu, T) = \frac{t^{\frac{\mu\nu}{T}-1}(T-t)^{\nu(1-\frac{\mu}{T})-1}}{\mathrm{B}(\frac{\mu\nu}{T}, \nu(1-\frac{\mu}{T}))T^{\nu-1}},$$

where B is the beta function.

The combination of multiple beta distributions will be able to account for the multi-modality in the data whilst remain bounded in the time window

under consideration. For the base measure G_0 a uniform distribution and inverse gamma distribution is used for μ and ν respectively. The uniform distribution is parameterised between 0 and $T = 60 * 60 * 24$ and the inverse gamma distribution has α_0 fixed at 2 whilst β_0 also has a distribution placed on it such that it can be updated during the fitting process. This provides a prior distribution on ν that has infinite variance and thus large and small values can be inferred for the scale of the separate mixture components. The full model for each background density can be written as

$$\begin{aligned} f_D(t) &= \int k(t \mid \theta) \mathrm{d}G(\theta), \\ G &\sim \mathrm{DP}(\alpha, G_0), \\ G_0 &= \mathrm{Uniform}(\mu \mid 0, T)\mathrm{Inv\text{-}Gamma}(\nu \mid \alpha_0, \beta_0), \\ \beta_0 &= \mathrm{Gamma}(1, 0.125). \end{aligned}$$

By using a Dirichlet process for the background rate, the inhomogeneous rate of background events can be learnt without placing any assumptions of behaviour on the data and instead, by using an unsupervised algorithm, the model can learn from the data. This type of specification allows for each day of the week to develop its own features, as seen in Figure 5.1.

However, treating each day of the week separately is restrictive and there will be cases where common features in the data, such as the location of the peaks in intensity, would benefit from including data from the other weekdays. Therefore it is desirable to introduce some pooling of the data, such that each background rate can benefit from an increased amount of data whilst retaining individual characteristics. This motivates the use of a hierarchical model and specifically a hierarchical Dirichlet process.

5.3.1 Hierarchical Dirichlet Processes

A hierarchical Dirichlet process is an extension to the Dirichlet process where the base measure G_0 is itself a Dirichlet process too (Teh et al., 2005). In this example, by replacing the base measure with another Dirichlet process, the individual background rates for each Hawkes model can now be inferred

together by pooling data. There is a common distribution that links the individual background rates that provides a mechanism to share data between the different weekdays. By using a hierarchical Dirichlet process the model for each weekday is able to learn its own background rate that benefits from the pooling of the data from the other weekdays which will provide a better inference of the underlying distribution than if each weekday was treated separately.

Hierarchical Dirichlet processes are well suited for data that forms natural groups. If there are j groups each with observation x_{ji} then j labels the assigned group and i indexes the observation within that group. Each group requires its own distribution that is drawn from a common distribution of all groups. This allows both flexibility and cohesion across the groups.

Such a model can be written as

$$\begin{aligned} x_{ji} &\sim F_j, \\ F_j &= \int k(x \mid \theta_j) \mathrm{d}G_j, \\ G_j &\sim \mathrm{DP}(\alpha_j, G_0), \\ G_0 &\sim \mathrm{DP}(\gamma, H), \end{aligned}$$

where α_j is the concentration parameter for each G_j, γ the global concentration parameter and H is the global base distribution. The individual distributions F_j are drawn from G_j which in turn are linked by the common base measure G_0.

The stick-breaking construction of the Dirichlet process (Sethuraman, 1994) is extended to account for the hierarchy (Teh et al., 2005). Like the non-hierarchical Dirichlet process, G_0 can be represented as an infinite sum of weights and atoms

$$\begin{aligned} G_0 &= \sum_{i=1}^{\infty} w_i \delta_{\theta_i}, \\ w &\sim \mathrm{GEM}(\gamma), \\ \theta_i &\sim H, \end{aligned}$$

as the parameters of G_j are drawn from G_0 it also can be written as a sum over these atoms with weights

$$
\begin{aligned}
G_j &= \sum_{i=1}^{\infty} \pi_{ji} \delta_{\theta_i}, \\
\pi_{ji} &= \pi'_{ji} \prod_{l=1}^{i-1} (1 - \pi'_{jl}), \\
\pi'_{ji} &\sim \text{Beta}\left(\alpha_j w_i, \alpha_j \left(1 - \sum_{l=1}^{i} w_l\right)\right),
\end{aligned}
$$

here the explicit sharing of cluster parameters θ_i can be seen where each group of data has its own independent weights π_{ji}. This allows datasets to share cluster parameters θ_i whilst learning the structure separately. Each group has its own mixture model that is linked via the global curve G_0. For further details see Teh et al. (2005).

Figure 5.3 demonstrates how a hierarchical Dirichlet process can be visualised. The global curve is G_0 and the two dataset curves, are G_1, G_2 respectively. The two child curves are composed of components of the global curve. A novel model for the arrival of FX trades can be developed by using a hierarchical Dirichlet process as the basis of the background rate $\mu_d(t)$ to control the distribution of trades over the day such that each weekday can develop its own profile using its own information and that from the other weekdays. For the total number of trades in a day, the amplitude of the background rate must be considered.

5.3.2 Hawkes with Covariates

From Fig 5.2b there is variation between the number of trades on each day and similarly there is also a local trend in the data evident in Fig 5.2a. This type of behaviour is accounted for by using covariates in the background rate $\mu(t)$ must be used. As illustrated in Equation (5.1) the background rate is decomposed into an amplitude and a density. The amplitude is modified using various covariates with both regressive and autoregressive components to model this change in the number of trades per day. As such, the specific form of μ_d is

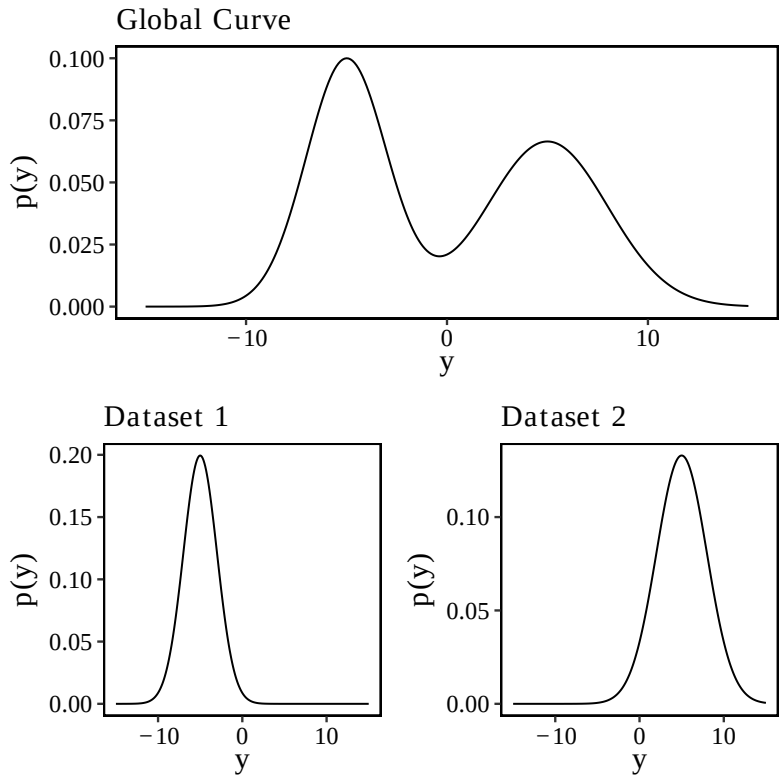


Figure 5.3: Example of a hierarchical Dirichlet process.

increased in complexity based on the features included in the model. These regression terms and autoregressive components will be added incrementally to ensure that overfitting does not occur.

The use of covariates in self exciting point processes has been studied recently in Pitkin et al. (2018). They use seasonal effects and indicator variables for various time based features in the data. In this chapter a similar approach will be used with both indicator variables for the day of the week and autoregressive terms to capture the local trend.

5.3.3 The Full Model

By combining the Hawkes process with a hierarchical Dirichlet process and a regression structure the full model can now be written.

Trades t_{di} are modelled by a Hawkes process with intensity

$$\lambda_d(t \mid H_{t_{di}}) = \mu_d(t) + \kappa \sum_{t_{di}<t} g(t - t_{di}), \tag{5.2}$$

where d indexes the day and i indexes each event during the day, $i = 1, \ldots, N_d$ where N_d is the total number of trades on day d. The kernel $g(t)$ is the exponential probability distribution function $g(t) = \eta \exp(-\eta t)$. This is the recommended kernel if there are no strong views on the decay of the intensity (Liniger, 2009) and as this work is concerned with the background seasonality we have no such views.

Both κ and the kernel $g(t)$ will remain constant across the days of the week leading to a single κ and kernel parameter to be inferred from the data. This ensures that the impact of a single trade is constant throughout the day and the main variation in trading differences is driven by the intraday seasonality rather than changes in the impact of each trade. Whilst there is potential that the impact of each trade varies throughout the day and between days of the week it is more likely that the dominant effect comes from the background rate and thus keeping these parameters constant provides a way to assess the average market impact of a trade.

The background function $\mu_d(t)$ is decomposed into two components: an amplitude μ_d that controls the total amount of background events on day d, and $f_d(t)$ the density of these background events throughout the day. As previously mentioned, a hierarchical Dirichlet process will be used to model the intraday background event density. The days will be grouped into 6 different classes, indexed by D, based on the day of the week of t_{di} including NFP days. Each weekday density will be a Dirichlet process mixture model with base measure G_D with $D = 1, \ldots, 5$ to represent Monday to Friday and $D = 6$ to represent the NFP day. As the number of background events is bounded between midnight to midnight the next day, the generalised beta distribution with mean μ and scale ν is used as the mixing kernel k. The background

density can be directly specified as

$$\begin{aligned}
\mu_d(t) &= \mu_d \cdot f_D(t), \\
f_D(t) &= \int k(t \mid \theta) \mathrm{d}G_D(\theta), \\
k(t \mid \theta = \{\mu, \nu\}) &= \mathrm{Beta}(t \mid \mu, \nu, T), \\
G_D &\sim \mathrm{DP}(\alpha_D, G_0), \\
G_0 &\sim \mathrm{DP}(\gamma, H), \\
H &= \mathrm{Uniform}(\mu \mid 0, T)\mathrm{Inv\text{-}Gamma}(\nu \mid \alpha_0, \beta_0), \\
\beta_0 &\sim \mathrm{Gamma}(1, 0.125),
\end{aligned}$$

where α_j is the concentration of each component of the hierarchical Dirichlet process, γ the concentration parameter of the global Dirichlet process, G_0 and α_0 is a prior parameter fixed at the value 2. The parameter T defines the right boundary of the beta distribution. As each day is modelled separately this is fixed at 86,400 seconds (24 hours).

5.3.4 Posterior Inference

The algorithm developed in Section 4.4 is extended to account for the hierarchical dataset. The hierarchical structure is explicit as each day is a weekday that consists of its own event times that have been generated from its own intensity function. The weekdays are grouped together and the parameters of each of these intensity functions are learnt in a hierarchical manner and for this, an extra level of notation must be introduced.

At any time t the Hawkes intensity Eq. (5.2) is a superposition of multiple Poisson processes; one process with intensity $\mu_d(t)$ and multiple inhomogeneous Poisson processes for every previous event that has occurred $t_{di} < t$. For each event t_{di} the intensity of this inhomogeneous Poisson process is $\kappa g(t - t_{di})$. Previously in Section 4.4, the latent variable $\mathbf{B}$ was introduced. Now this variable must be extended to account for the multiple days, then the event times can be partitioned into appropriate sets $S_{d0}, \ldots, S_{dn}$ for each day

$$S_{dj} = \{t_i; B_{di} = j\}, \quad 0 \leq j < n, \tag{5.3}$$

the set S_{d0} counts all the events that are caused from the background intensity rate $\mu_d(t)$ on day d and $S_{di}, i > 0$ counts the number of children each event is responsible for and thus from the Poisson process with intensity $\kappa g(t - t_{di})$ on day d.

The unknown components of the Hawkes model are $\Theta = \{\mu_d, f_d(t), \kappa, \eta, \mathbf{B}\}$ and must be inferred. Again, to sample from the posterior distribution of these parameters the branching structure between events must be simulated.

The latent structure variable $\mathbf{B}$ is not observed. It must be estimated which events were immigrants caused from the background rate, $\mu(t)$, and which events were children directly caused by other events. Each event t_{di} has an associated B_{di} that indicates its parent. A value of $B_{di} = 0$ is indicating that the event was generated by the background rate and $B_{di} = j$ indicates that event t_{dj} is the parent of event t_{di}. For the i'th event occurring at time t_i, the probability of its parent event is

$$\begin{aligned} \Pr(B_{di} = 0 \mid \mu_d(t), \kappa, \eta) &= \frac{\mu_d(t_{di})}{\lambda(t_{di} \mid H_t)}, \\ \Pr(B_{di} = j \mid \mu_d(t), \kappa, \eta) &= \frac{g(t_{di} - t_{dj})}{\lambda(t_{di} \mid H_t)} \quad j = \{1, 2, \ldots, i-1\}, \end{aligned} \tag{5.4}$$

therefore to arrive at a value of B_{di} this probability distribution must be used to generate samples. Note that only events on the same day can cause further events, there is a boundary at midnight where each process starts afresh.

This algorithm allows for full simulation of the branching structure of a Hawkes process from the exact posterior using values of the parameters $\mu_j(t), \kappa$ and η. Then conditional on this branching structure, the other parameters of the model can be estimated. This removes the need for MCMC sampling of the structure between each event as used in Rasmussen (2011). Now using a sample of the structure parameters the sets S_{di} can be formed using Eq. (5.3) and used to estimate the other parameters of the Hawkes model.

The events with $B_{di} = 0$ provide the information to perform a posterior sample of μ_d. The total number of background events per day S_{d0} is used to

update amplitude

$$S_{d0} \mid \mathbf{B} \sim \text{Poisson}(\mu_d),$$

For models where μ_d is constant and has no regression components, the conjugate gamma distribution prior can be used to perform direct samples of the posterior distributions. In models where there are regression components of μ_d the probabilistic programming language Stan is used to sample from the posterior distribution (Carpenter et al., 2016).

The actual times of the background events are used to update the hierarchical Dirichlet process model. Define t_{di}^{bg} as the events with $B_{di} = 0$ and therefore generated from density of the background rate

$$t_{di}^{bg} \sim f_D(t). \tag{5.5}$$

The overall sampling of the hierarchical Dirichlet process is achieved using the `dirichletprocess` package in R as detailed in Chapter 3. It is this posterior draw of $f_D(t)$ that functions as the background density for the next round of sampling. The parameter α_D of the Dirichlet process has a gamma prior (West, 1992) and sampled for each iteration of fitting.

As before, the κ parameter can be interpreted as the expected number of children events for each event. The number of children events from each event $S_{dj}, j > 0$ has a Poisson distribution

$$S_{dj} \mid \mathbf{B} \sim \text{Poisson}(\kappa) \quad j > 0.$$

The parameter κ can be inferred by using a conjugate gamma distribution and sampling from the posterior distribution

$$\kappa \mid \mathbf{S}, a_\kappa, b_\kappa, \mathbf{B} \sim \text{Gamma}\left(a_\kappa + \sum_{d=1}^{D}\sum_{i=1}^{N_d} S_{di}, n + b_\kappa\right), \tag{5.6}$$

where $n = \sum_{d=1}^{D} n_d$ and a_κ, b_κ are the prior parameters.

Inferring the parameter η of the kernel requires the transformation $\tau_{dj} = t_{dj} - t_{d\text{parent}}$ as the re-scaled event times for all m events with $B_{di} \neq 0$. The values of τ_{dj} are then used to sample the kernel posterior distribution

$$\tau_{dj} \sim \text{Exp}(\eta),$$

this can then be sampled directly using the conjugate gamma distribution as a prior

$$\eta \mid \tau \sim \text{Gamma}\left(\alpha_\eta + n_\tau, \beta_\eta + \sum_{d=1}^{D}\sum_{j=1}^{n_d} \tau_{dj}\right), \tag{5.7}$$

where $n_\tau = \sum_{d=1}^{D} n_d$ and α_η, β_η are the prior parameters.

Combining the structure simulations and posterior parameter samples leads to the full Gibbs sampling algorithm:

Algorithm 3: Sampling the parameters of the Hawkes model.

Data: Event times t_{di}

Result: S posterior samples of μ_d^s, f_D^s, κ^s, η^s and parent structure $\mathbf{B}^s$.

Chose starting values $\mu_d^1(t), \kappa^1$ and η^1;

for *i = 1 to S* **do**

- Sample the new parent structure $\mathbf{B}_d^{i+1}$ from probabilities calculated using Eq. (5.4) and $\mu_d^i(t), \kappa^i$, kernel $h^i(t)$;
- Calculate S_{dj} from $\mathbf{B}_d^{i+1}$;
- Sample μ_d^{i+1} and $f_D(t)^{i+1}$ using $p(\mu_d \mid S_{d0})$ and `dirichletprocess` ;
- Sample κ^{i+1} using Eq. (5.6) ;
- Sample η^{i+1} using the Eq. (5.7) ;

end

This highlights how the previous algorithm for sampling the parameters of the Hawkes process in Chapter 4 can easily be extended to account for multiple timeseries. Furthermore, its modular nature has also been demonstrated, the nonparametric kernel has been replaced with a parametric form and now the background rate has been modified to a nonparametric density with a regression structure on the amplitude.

5.3.5 Model Validation

The above model will be fitted on a subset of the full dataset and the resulting parameters will be used to assess the performance of the model on days that follow the training set but were not used in the fitting process.

To estimate the suitability of the models the predictive likelihood will be calculated using days that fall outside of the training dataset. The likelihood will be averaged across the full posterior samples of the parameters to ensure that it remains a Bayesian metric and the model with the highest average predictive likelihood is the best fitting model.

A simulation based approach will also be key in validating the models. A Hawkes process will be simulated using the posterior samples for each of the models described above. By comparing these simulations to the true data a strong check can be performed to ensure that the model produces realistic realisations with interest paid to both the number of events that arise from the simulations and the distribution of these events over the days. Any model that does not resemble the out-of-sample data can be discarded.

A key driver in using a Hawkes model is the conditional intensity function. Predictions can be updated using recent market trading activity, such that if there is a surge in the number of trades in a short period, the Hawkes process can adapt to this and revise predictions. To demonstrate this, a short interval will be shown where changes in observed trading intensity will have an effect on the prediction which is realised in the actual number of trades.

5.4 Results

All models were fitted on 40 days of data: from the 11th of January to the 4th of March 2016. This subsample of data includes 2 NFP days. The sampling algorithm used 2,000 iterations with 2 different chains each with dispersed starting values. The first 1,000 iterations are removed as burnin which is the required amount of iterations to ensure suitable mixing of the parameters and the sampling algorithm converges.

5.4.1 Inference

The first model fitted is a Poisson model with a Dirichlet process intensity function, a variety of Hawkes models are then fitted with each one introducing a new feature to improve the suitability of such a model. The different Hawkes

models, each with increasing complexity, are fitted to arrive at a well calibrated and accurate model. By incrementing the complexities of the model the added components can be examined to improve the forecast and ensure that there is not an overfitting of the dataset.

A Poisson Model

The most basic model that can be used for this data is a Poisson process with inhomogeneous rate. The rate is modelled using a Dirichlet process mixture model of beta distributions which can be written as:

$$\begin{aligned} N &\sim \text{Poisson}(\lambda(t)), \\ \lambda(t) &= \int k(t \mid \theta) \mathrm{d}G(\theta), \\ k(t \mid \theta = \{\mu, \nu\}) &= \text{Beta}(t \mid \mu, \nu, T), \\ G &\sim \text{DP}(\alpha, G_0), \\ G_0 &= \text{Uniform}(\mu \mid 0, T)\text{Inv-Gamma}(\nu \mid \alpha_0, \beta_0). \end{aligned} \tag{5.8}$$

Without the clustering behaviour this model fits poorly as Table 5.1 shows that is has the worst predictive likelihood value and all the Hawkes models improve upon it. However, as it has been highlighted in the literature review, trades are unlikely to be independent and thus this Poisson model is unlikely to be a fair baseline.

A Hawkes Model with Constant Parameters

The second model fitted to the data is a Hawkes model with constant parameters that does not distinguish between different days of the week. This acts as a better baseline model but is still expected to fit badly due to the intraday pattern in trades that are highlighted in Figure 5.1 and the variations between days of the week as indicated. Constant parameters cannot account for the dynamic patterns in the data, but, it is important to show that each layer of complexity added to the model improves on this baseline. Furthermore, this is the simplest model that can be considered realistic with self-exciting behaviour.

This model can be written as

$$\begin{aligned} \lambda(t \mid H_t) &= \mu + \kappa \sum_{t_i<t} g(t - t_i), \\ g(t) &= \eta \exp(-\eta t), \end{aligned} \tag{5.9}$$

where μ, κ and η are constants across each day of the week and there is no mechanism to distinguish different days of the week. The algorithm described in Section 5.3.4 is used to obtain posterior samples of the parameters, μ, κ and η where uninformative prior distributions are used for all these parameters. Table 5.1 shows the posterior means of the parameters and the out-of-sample predictive likelihood value. Again, it is expected that all further models with nonparametric and regression components will improve on this value.

A Hawkes Model with Dirichlet Background Rate

The above model is extended to account for the inhomogeneous event intensity by decomposing the background rate $\mu(t)$ into an amplitude and density. The amplitude remains constant and the density is a Dirichlet process mixture model of beta distributions much like the Poisson model in Eq. (5.8). This model can now be written as

$$\begin{aligned} \mu(t) &= \mu_0 f(t), \\ f(t) &= \int k(t \mid \theta) \mathrm{d}G(\theta), \\ G &\sim \mathrm{DP}(\alpha, G_0), \end{aligned}$$

where $k(t \mid \theta)$ is the beta distribution bounded on $[0, T]$ and μ_0 is a constant.

This type of model accounts for the natural trend in trade behaviour throughout the day, but generalises to each day behaving the same. It is expected to improve on the basic Hawkes model Eq. (5.9), but fail on days where the structure differs from the average behaviour, such as NFP days.

A Hierarchical Dirichlet Hawkes Model

For the next model the background rate is extended to allow for structural differences between weekdays. From the empirical evidence in Figure 5.1 it is necessary to account for a change in intensity throughout the day with a

different rate for each day of the week. The advantage of using a hierarchical Dirichlet process means that extra groups can easily be incorporated. In this case the data can be partitioned into separate weekdays and an extra group for NFP days. This leads to a background rate with separate densities for each day of the week and NFP day leading to six groups in total. The amplitude of the background rate, μ_0 is constant for all days.

Using a hierarchical Dirichlet process, each day of the week and NFP day has its own density f_D and can be written as

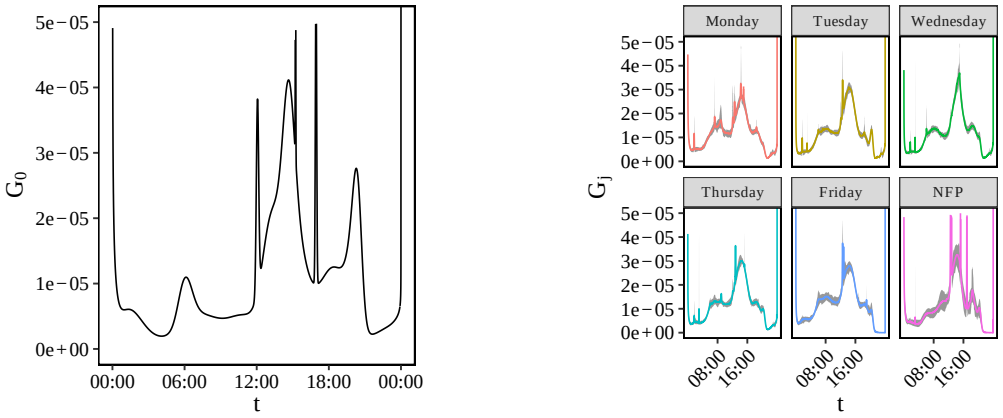
$$\begin{aligned} \mu_d(t) &= \mu_0 f_D(t), \\ f_D(t) &= \int k(t \mid \theta) \mathrm{d}G_D(\theta), \\ G_D &\sim \mathrm{DP}(\alpha_D, G_0), \\ G_0 &\sim \mathrm{DP}(\gamma, H), \end{aligned} \tag{5.10}$$

where D labels whether the day is Monday, Tuesday etc. The κ and η parameters remain constant. There are now 6 individual Dirichlet processes G_D and a global Dirichlet process G_0 that must be learnt from the data. Figure 5.4 shows the components of the Dirichlet processes inferred from the data.

From Table 5.1 we can see that the estimated posterior mean of κ decreases as more of the events are generated from the background rate. The parameter of the kernel has also increased, which leads to a decrease in the average impact time scale of the event. Furthermore, Table 5.1 shows an increase in the predictive likelihood, therefore this is an improvement on the baseline model.

A Hierarchical Dirichlet Hawkes Model with Regression Covariates

So far the parameter μ_0 has remained constant. This parameter controls the number of background trades in each day and as per Figure 5.2b it is known that the number of trades can vary significantly based on the day of the week. Therefore to account for this variation a regression component in the background amplitude is included and the exponential function is used to ensure



(a) Global curve G_0.

(b) Individual curves G_D. Each cluster in each of these curves comes from the global curve G_0.

Figure 5.4: Hierarchical Dirichlet process results from the model defined by Eq. (5.10).

positivity for the intensity function.

$$\begin{aligned} \mu_d(t) &= \mu_d f_D(t), \\ \mu_d &= \exp(\beta_0 + \beta_d x_d), \end{aligned} \tag{5.11}$$

where x_d is an indicator variable highlighting whether the day is a Monday, Tuesday etc. The parameters β_d are given an uninformative prior. The hierarchical Dirichlet process for $f_D(t)$ remains the same as the previous model.

Figure 5.5a shows the distribution of the posterior samples for each β_d. A larger average value of β_d (Wednesday's and Thursday's) indicates that these days have a greater number of background events and therefore a greater number of trades in total. This agrees with our empirical findings from Figure 5.2b. Namely, the number of trades on Wednesday is higher than average and the number of trades on Mondays and NFP days is lower.

Table 5.1 shows a further increase in the predictive likelihood on both the baseline model and the previous hierarchical Dirichlet model. Therefore this addition of regression covariates is a further improvement to the model.

A Hierarchical Dirichlet Hawkes Model with Regression and Autoregression Covariates

The previous models have been static with regards to seasonality in the data. Figure 5.2a shows that there is a slight trend in the number of trades per day. To account for this an autoregressive component is included based on the total number of trades on the previous day

$$\begin{aligned} \mu_d(t) &= \mu_d f_D(t), \\ \mu_d &= \exp(\beta_0 + \beta_d x_d + \beta_{\text{AR}} N_{d-1}), \end{aligned} \tag{5.12}$$

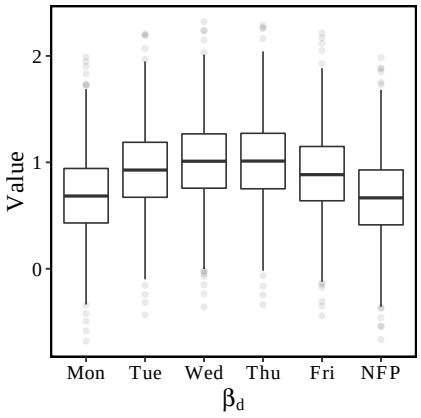
where N_{d-1} is the total amount of trades in the previous day. The total number of trades is normalised by the mean and standard deviation of the training set to ensure that the samples of β_{AR} are on a similar scale to the β_d values.

Figure 5.5b shows that the AR component of the model has an established effect as the posterior samples are clearly greater than 0. This means that days with increased trading activity follow other days of increased activity and likewise, lower trading levels leads to lower trading levels the next day. However, when consulting Table 5.1 it is found that this does not improve the likelihood. The autoregressive effect is not pronounced enough to improve the out-of-sample criticism.

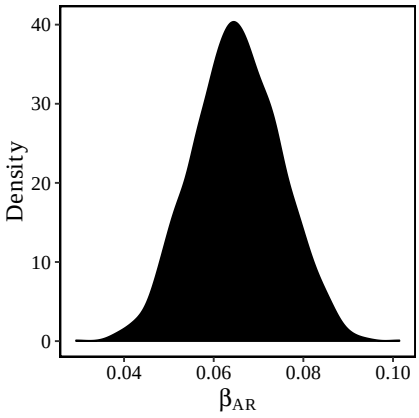
In the previous analysis of the autocorrelations between the number of trades per day (Figure 5.2c) there is a peak on multiples of 5 days. The effect of a large total number of trades on any day increases the total number of trades for the next week, i.e. a large amount of trades on a Monday also effects the next Monday. Therefore, it is intuitive to replace the previous autoregressive component with one dependent on the 5th previous day

$$\begin{aligned} \mu_d(t) &= \mu_d f_D(t), \\ \mu_d &= \exp(\beta_0 + \beta_d x_d + \beta_{\text{AR5}} N_{d-5}). \end{aligned} \tag{5.13}$$

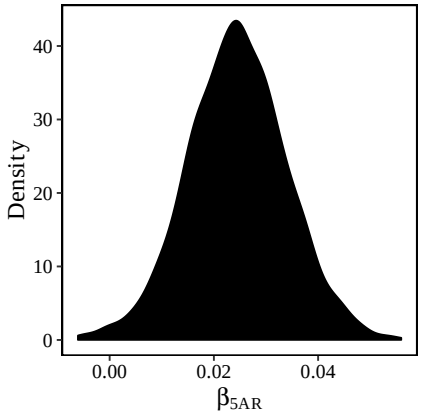
After fitting this model it is found that this leads to the largest and therefore best predictive likelihood value out of all the models considered. This is the best model suited to the data at hand.



(a) Box plot of the posterior samples for the β_d's.



(b) Density estimation of the posterior samples for the β_{AR}'s.



(c) Density estimation of the posterior samples for the β_{AR5}'s.

Figure 5.5: Posterior sample distributions of the regression parameters.

When comparing the inferred parameters for all models, Table 5.1, the value of κ for the best fitting all model is centered around ≈ 0.3. This can be interpreted as that every trade has on average ≈ 0.3 child events, or alternatively every 3 trades leads to another child trade on average. Similarly, the value of η is centered around ≈ 0.05, which means that each events impact on the intensity lasts for $\frac{1}{\eta} = 20$ seconds on average. Table 5.1 also shows how once the background rate is allowed to vary throughout the day more events are background events and reduces the size of κ. This confirms our previous assumption that the variation in events through the day is driven by a daily

Table 5.1: Posterior means and out of sample likelihood values. The parameter values are inferred from the training data and the likelihood values are calculated on the test data.

Model	μ	κ	η	Likelihood
Poisson	-	-	-	-44619
Constant Hawkes	234	0.655	0.0103	-42747
Dirichlet Hawkes	447	0.342	0.0479	-42246
Day Dirichlet Hawkes	456	0.328	0.0510	-42190
Day Dirichlet Regression		0.322	0.0525	-42159
Day Dirichlet Regression AR		0.321	0.0528	-42165
Day Dirichlet Regression 5 AR		0.317	0.0540	-42148

pattern rather than the κ and kernel parameters. Again, these differences in log likelihoods are also confirmed with by using out-of-sample predictions using simulations of Hawkes processes.

5.4.2 Posterior Simulations

Simulating Hawkes processes using the MCMC samples of parameters allows for the assessment of model suitability. For a good model, the distribution of simulated events should match the observed events. This is the posterior p-values approach of model checking as detailed in Section 2.3.2

To highlight the lack of fit of the baseline model, detailed in Eq. (5.9), Hawkes processes are simulated using the posterior samples of the constant parameters. Figure 5.6a shows that the occurrence of events is constant over the time period, which contradicts the observations from Figure 5.1. Therefore it can be concluded that a Hawkes process with all constant parameters is unsuitable for the data.

Figure 5.6b is an improvement over the base model (Figure 5.6a), as the variance in the number of events over the course of one day has now been captured. However, this is still not sufficient as it is known that there is a

difference in behaviour across different weekdays (Figure 5.1).

Figure 5.6c shows the resulting Hawkes simulations from the posterior samples of the model in Eq. (5.10). The different daily empirical pattern is replicated successfully and shows that the hierarchical density component of the background rate is sufficient to replicate the intraday behaviour of the trade times. For the hierarchical Dirichlet process Hawkes models the background rates in Figure 5.4 show potentially pathological spikes in intensity which arise from Dirichlet process assigning a cluster with vanishing variance to this particular time of day because of a large amount of trades timestamped at that particular time. However, after simulating Hawkes processes with these background intensities, as shown in Figure 5.6c, theses spikes do not affect the actual event generation negatively, so whilst the background rate can have an awkward shape, generating enough simulations remedies this behaviour.

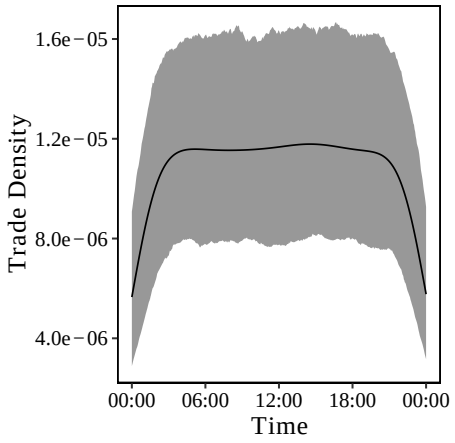
The further models, Eq. (5.11), (5.12) and (5.13), do not effect the intraday distribution of the trades and it can be safely assumed that their intraday distribution is comparable to that of Figure 5.6c. Predictive performance can now be assessed.

5.4.3 Daily Forecasts

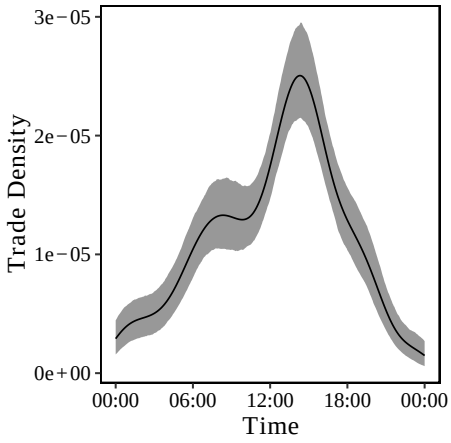
The inferred parameters are used to simulate the next week of data which can then be compared to the true but unobserved data. Both aspects of the true data are assessed, the intraday distribution and the total amount of trades per day. This is both a visual check on model performance and an example of using posterior p-values as outlined in Chapter 2.

The model with highest predictive likelihood (Eq. (5.13)) is used to simulate the next week of trading and the next NFP day.

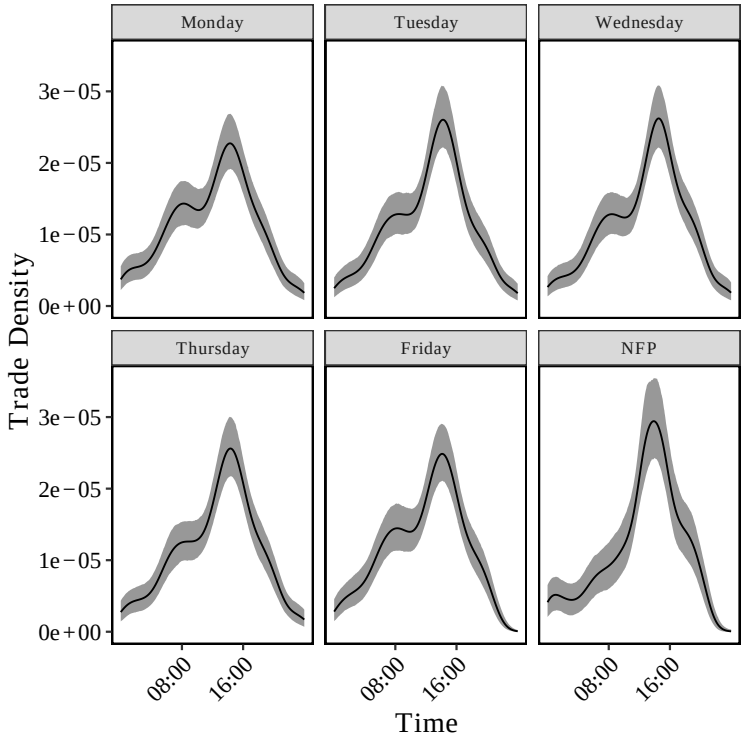
Figure 5.7a presents the predicted intraday behaviour of the number of trades. By comparing this prediction with the true result (the red line), it is evident that the predictions from the Hawkes model performs well and rarely does the true density fall outside the credible interval. Most importantly though, the structural differences between normal weekdays and NFP days has been



(a) A background rate with constant parameters.

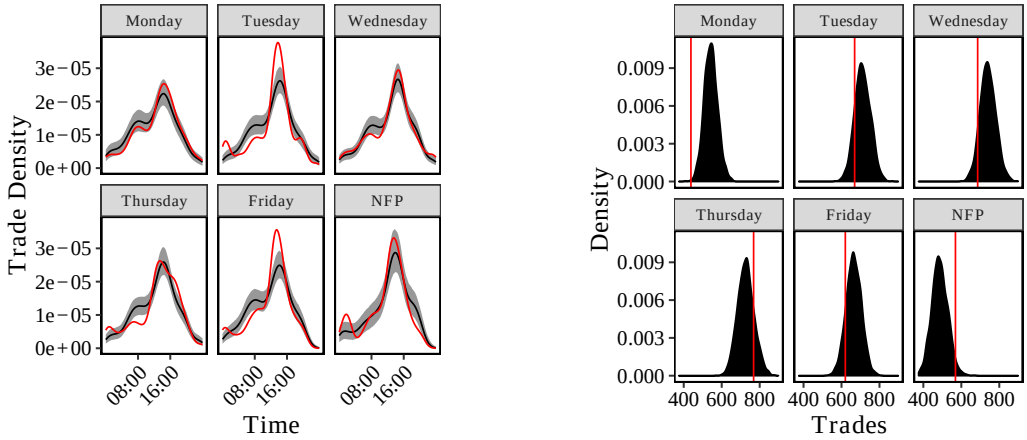


(b) A single Dirichlet process as the background rate model.



(c) A Hierarchical Dirichlet process as the background rate model.

Figure 5.6: Events densities from Hawkes process simulated with posterior samples of the model parameters. The solid black line indicates the posterior mean, the shaded region is the 95% credible region.



(a) Forecasted intraday density of trades for the next week and next NFP day. The red line indicates the true event density.

(b) Forecasted number of trades for the next week and next NFP day. Red line indicates the actual number of trades.

Figure 5.7: Forecasts for the next week and next NFP day.

correctly predicted. The dataset only features two NFP days compared to the five other weekday groups, therefore, the information sharing ability of the hierarchical Dirichlet process has enhanced the ability of the model to adapt to different patterns.

Furthermore, in Figure 5.7b the predictions for the total amount of trades per day all fall within the predicted distributions of trades. For all days (apart from Monday) the true number of trades falls within the 95% credible interval. For the Monday, it was off by 10 trades. This shows that our model has predictive power in forecasting the number of trades in a day.

5.4.4 Intraday Forecasts

The advantage of a Hawkes model over a standard Poisson point process is its ability to adapt and respond to sudden bursts in event numbers. The intensity function of the Hawkes provides a way for the occurrence of future events to be influenced by the past history of the same process. This conditional intensity function provides a direct prediction of the total number of future events. If a large number of events suddenly occur, the conditional intensity function of the Hawkes process is able to accommodate this information in predicting the future number of events. So given all the information up to time t a future prediction about $\hat{t}$ some time in the future can be made using the integrated intensity function from Eq. (2.1) where the time interval is $\left[t, \hat{t}\right]$.

To demonstrate this behaviour it will be shown how a Hawkes model can forecast trades based on the observed number of trades earlier in the day and then update a prediction throughout the day. An NFP day will be used to indicate how the model is able to adapt to perturbations in normal behaviour. For a set interval throughout the day, the number of trades will be forecasted using the history of the process so far. The forecast will be repeated for each posterior sample of the parameters to provide an average and credible interval to the forecast.

For five minute intervals the Hawkes process is simulated using what was actual observed in the market as the previous history of the process. This

produces updated predictions of the number of trades expected in the next five minutes. Here Figure 5.8 shows that all the actual number of trades fall

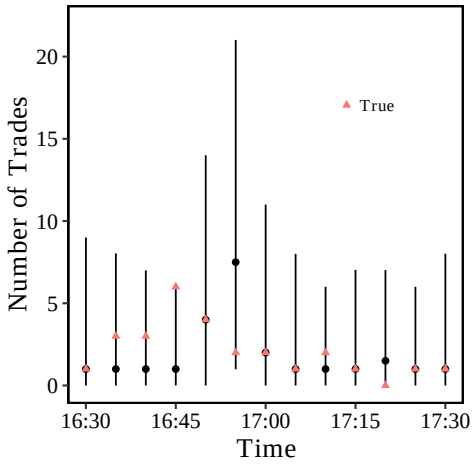


Figure 5.8: Five minute interval on the out-of-sample NFP day. The number of trades is bucketed into 5 minute intervals.

within the credible intervals predicted. A particular area of note is from 16:30, the next 3 true amount of trades are higher than the average expected. This causes the model to forecast a higher amount of trades for the next 3 intervals, which is then observed. This increase in market activity then reverts, which causes the Hawkes model to return to the background rate. Given that this data is part of the test set and unseen in the inference process, it shows that the Hawkes model is correctly adapting to the history of the process as the day progresses.

This ability to update predictions based on what is happening in the market is a key feature of the Hawkes process and by confirming that the predictions are grounded in what is actually observed makes it a very practical model for predicting current and future market conditions.

5.5 Discussion and Further Work

We have shown how a nonparametric Hawkes processes can provide good models for FX trades. The empirical differences in day to day behaviour of trading can be adequately described using a combination of hierarchical Dirichlet pro-

cesses and Hawkes processes. Clustering behaviour in the occurrences of trades can be explained by using a Hawkes process and offers a direct improvement over a standard Poisson process.

The hierarchical Dirichlet process has provided a method of separately modelling the days of the week in the data whilst sharing the data amongst all the groups. This has helped ensure coherence between the predicted trade densities, i.e. the peaks in intensities are shown to occur at roughly the same time, whilst the magnitude of the peaks is individual to each day of the week. This sharing of information is more pronounced in the modelling of the NFP day where less data is available.

By introducing a regression structure an autoregressive nature has been shown to exist in the number of days. The number of trades per day is mostly effected with a term that depended on the previous week activity. This was shown to have a positive coefficient, therefore, days with high trading activity lead to further days of high activity and likewise, quiet days follow other quiet days.

The benefits of this model are twofold. Firstly, a conditional intensity function has been introduced that can be used to forecast the time of trades given the history of the trades so far. This can be used to make predictions about the number of trades likely to happen at whatever timescale necessary. For practical applications, this means that the market view of the number of trades can be updated in real time with dynamic forecasts. Secondly, this model also provides forecasts for the total number of trades per day. Therefore traders can assess the likely trading behaviour for the next week and plan accordingly. Traders can improve the prices that they trade on by incorporating this volume information into their execution strategies.

Further investigation into the variability of κ and the kernel $g(t)$ is also an avenue to explore as in this work these variables have remained constant. In practise, trades that happen outside of typical business hours are likely to have a different impact due to changes in liquidity conditions.

This chapter has also highlighted how the Bayesian Hawkes process can be easily adapted to a new application. Chapter 4 used a nonparametric kernel, this chapter has now used a nonparametric background rate. The process of sampling from the Hawkes model was not changed significantly despite the change in nonparametric component. Also, the introduction of multiple timeseries and hierarchy in the model did not drastically change the algorithm.

Chapter 6

Bayesian Multivariate Hawkes Processes with Applications to Soccer Goals

In both previous chapters the Hawkes process has been applied to a single time dimension where the events have occurred. In this chapter the Hawkes process is extended to multiple dimensions where the events can occur and excitation can exist both mutually and across these dimensions. This type of model is then applied to the occurrence of goals in a soccer match where the home and away team occupy their own time dimension.

The Poisson distribution has long been a favoured model for predicting the number of goals scored in a soccer match. The final number of goals in a match closely follows a Poisson distribution (Heuer et al., 2010), but the actual goal times throughout a game is an underexplored topic compared to the final scorings of a match. The goal scoring rate for both teams in a match is unlikely to be constant over the time the match is played and there is a high degree of interaction between both competitors, leading to variable scoring rates over time. Therefore, the Poisson process needs to be adapted to account for the correlation between when the goals are scored by the home team and when the goals are scored by the away team. In this chapter a bivariate self-exciting process is proposed which provides a conditional scoring rate for both the home

and away teams in a soccer match.

Understanding the dynamics of goals in a soccer match is an area of great importance. Goals win games and understanding how a team reacts both when they score a goal and when they conceded a goal can help shed some light on a teams strengths and weaknesses. For coaches, identifying periods of vulnerability after scoring can indicate where training efforts or tactical improvements can be made. For sport bettors, understanding how the betting market responds to goals can help traders judge whether price movements are reasonable and then take advantages of any mispricing.

As shown in the previous chapters, the Hawkes process provides a time varying intensity function that is conditional on the history of the process. This function can then be used to explain clustering between events as the multivariate Hawkes process is an extension where there are multiple dimensions that cause excitations both in their own dimensions and across the other dimensions. Multivariate Hawkes processes have been used in finance (Embrechts et al., 2011) and molecular biology (Carstensen et al., 2010) where both works examine the influence of events on the further occurrence of more events of both the same and different types. In Embrechts et al. (2011) two dimensions are used to look at the interaction between +10% and -10% returns in stock indexes. The positive and negative returns form two time series where an event occurs when there is a large price movement. The multivariate Hawkes process allows for a large price movement that can influence the probability of both another movement in the same direction and an extreme movement is the opposite direction thus allowing for the interaction between these two type of events. Similarly, Carstensen et al. (2010) use eleven dimensions to model the interaction between different elements in a gene. The interaction between different elements in a gene can be difficult to untangle and deduce what specific element lead to a certain result. The Hawkes process helps separate out the elements and build a better picture of their interactions and subsequent occurrences. Both of these works show how the multivariate

Hawkes process is a very general model for the case of interacting events and therefore it is logical to take a similar approach to modelling soccer goals.

This chapter begins by reviewing the available literature before proceeding to explore the data and highlighting the need for a point process model. Then the background from Chapter 2 is extended to account for multiple variables in the Hawkes process. The new model is then applied to a large dataset of soccer matches from a wide range of European competitions. After training the model and assessing model performance it is demonstrated how the model could be used in a live context by comparing the changing predictions of a match to the predictions derived from a betting market.

6.1 Literature Review

Sports betting is estimated to be worth approximately £500 billion a year, of which 70% is believed to be wagered on soccer matches (Keogh and Rose, 2013). A bookmakers success and profitability depends on their ability to correctly price a soccer match and adjust these prices as necessary throughout the match. If the bookmakers odds are incorrect they might find themselves on the wrong side of a bet and risk losing money depending on the outcome of the match. Alternately, the odds that the bookmaker sets could be uncompetitive and a better price could be found elsewhere, therefore they lose market share to the better bookmakers. A bookmaker must offer correct and competitive odds so that they can minimise their risk whilst still making a profit.

The seminal work in soccer modelling is Dixon and Coles (1997) which analysed three years of English league and cup football data. They use a modified Poisson distribution to model the goals scored by each team, where each team scores at a rate equal to their attack parameter minus the opposition defense parameter. The Poisson distribution is then modified to account for low-scoring matches. Under this model they are able to show that a profitable strategy can be executed in the betting markets when comparing their calculated probabilities compared to the bookmakers implied probabilities. How-

ever, this work is focused on the total number of goals in a game and is not concerned with when the goals occur.

The occurrence of goal times has been studied empirically in Armatas et al. (2007) where it was found that goals are not uniformly distributed throughout the match and that more goals are scored both in the second half and in the last 15 minutes of a match. In this study the data consisted of the 192 matches of the 1998, 2002 and 2006 World Cups and shows that both the total number of goals and when the goals are being scored has to be taken into consideration.

A survival analysis approach has also been applied to the effect of goals. In Nevo and Ritov (2013) they examine the relationship between when the first and second goal occur in a soccer match and conclude that there is a such a relationship that is time dependent. It is also found that goals are self-exciting which is further evidence that the Hawkes process is well suited for both modelling and predicting soccer goals.

In Heuer et al. (2010) the suitability of a Poisson process is used to assess the occurrence of goals. They find that a teams strength remains constant over a season and conclude that the number of goals scored by each team is mostly Poisson - with a slight disagreement around the 0-0 score. Again, this provides more evidence that when goals are scored be modelled with a point process.

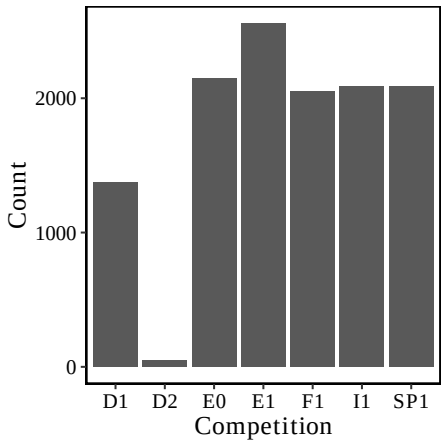
In terms of multivariate approaches, a composite Poisson model has also been used in Everson and Goldsmith-Pinkham (2008). They decompose a teams goal scoring rate by both their ability to score and concede goals plus the effect of playing at home or in a neutral stadium. Then conditional on the total number of goals scored, they are able to infer these latent parameters. A bivariate Poisson model has also been studied by Karlis and Ntzoufras (2003). The model allows for correlation between the two scores and thus they are no longer independent and this model includes parameters such as team strength and the home effect. The correlation parameter depends on each individual team playing the match. Each of these papers find that their specified multi-

variate models improve on the modelling of the final scores of the match, but none provide in-play predictions of the scoring rate of a team.

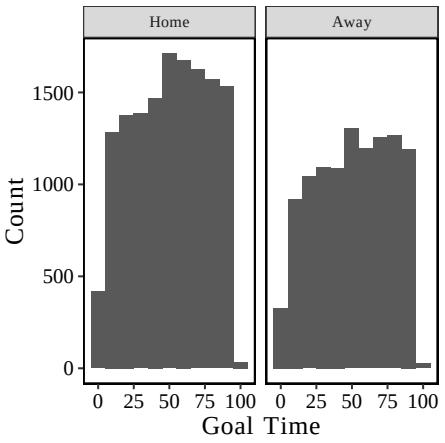
Overall, there is an opportunity to marry two themes of the literature; predicting the total number of a goals in a match and when these goals will occur. This chapter demonstrates how a multivariate Hawkes process can be used to achieve this goal.

6.2 The Dataset

The data consists of 12,347 soccer matches from multiple European competitions from 2012 to 2018. The competitions include: both German professional leagues, the Bundesliga (D1) and Bundesliga 2 (D2), the top two English competitions, the Premier League (E0) and Championship (E1), the top French League, Ligue 1 (F1), the top Italian league, Serie A (I1), and the top Spanish league, La Liga Primera (SP1).



(a) Frequency of competitions in the dataset. The letter corresponds to the country and the number the competition level.



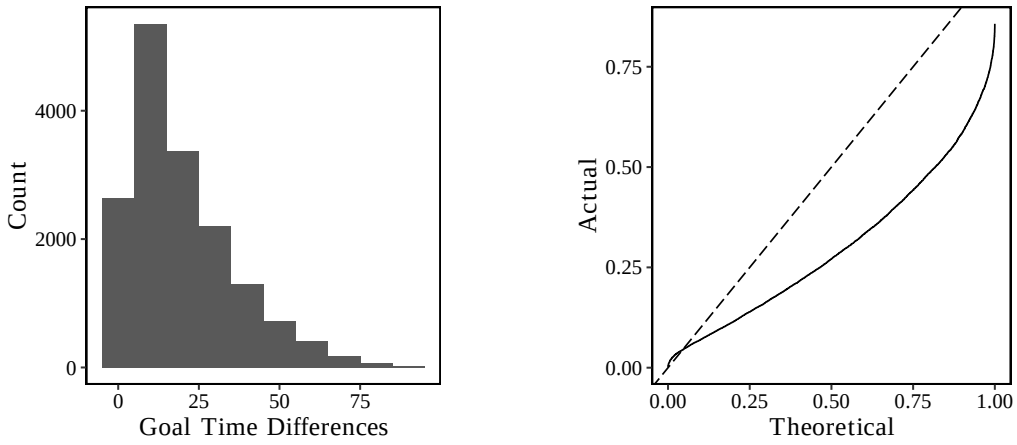
(b) Distribution of goal scoring time for both the home and away team.

Figure 6.1: Empirical distributions of variables in the dataset.

Figure 6.1a shows the occurrences of the competitions in the dataset. Each competition has a similar amount of game data except for the Bundesliga 2

matches.

The distribution of goal times is also shown in Figure 6.1b, it extends beyond 90 minutes due to stoppage times for each match. More home goals are scored than away goals which is a known phenomena and has been well studied previously (Pollard, 1986). Similarly, there are more goals scored in the second half of the match compared to the first.



(a) Histogram showing the difference in times between goals.

(b) Residual analysis of the difference between goal times.

Figure 6.2: Time differences between goals.

Figure 6.2a shows that most goals occur 10-20 minutes after a previous goal has been scored. If the time at which goals were scored follows a Poisson process then the difference between goal times would be exponentially distributed. This can be quantified by using the time-rescaling theorem (Brown et al., 2002) to check the if the goals are from an independent Poisson process. If they were from such a process, the actual quantiles would fall on the straight line indicated in Figure 6.2b. This is clearly not the case and suggests the goals are not generated i.i.d from a Poisson process which motivates the use of a more flexible process.

The number of goals a specific team will score in a match will also be directly influenced by their strength on that day and their opponents strength. Rather than model this strength directly for each match a proxy for their

strength can be derived from the match outcome probabilities. Therefore, the times of goals in each match are also joined with the Pinnacle Sports closing odds of the match. This is the last recorded price before a match starts for the 1X2 market, i.e. whether the home team will win (1), the draw (X), or the away team will win (2). As Pinnacle Sports is a trading exchange, bettors are matched with each other, rather than against a bookmaker. This means that there is a market clearing mechanism in place and bettors with better knowledge and more information on the match will find odds that are mispriced, trade these odds, causing them to move to the true price that represents the new information. Therefore, the last traded price before the match starts is likely to represent the true probability of the outcome of the match. This has been studied systematically in Franck et al. (2010) and it was found that a positive return could be extracted by exploiting the difference between exchange and bookmarker prices. Similarly, the in-play odds move towards to the true outcome efficiently based on the in game events (Debnath et al., 2003).

6.3 Method

The background outlined in Chapter 2 is built upon, again taking the standard Hawkes process and extending it to multiple variables, one for the occurrence of home goals and another for the occurrence of goals scored by the away team.

6.3.1 The Bivariate Hawkes Process

Equation (2.3) is used with $m = 2$ as we are modelling two types of event; the goals scored by the home team and the goals scored by the away team. This leads to a 2×2 matrix K for the κ_{ij} parameters and a 2×2 matrix G for the kernel parameters. Therefore, for two dimensions of the Hawkes process there are four κ values and four kernel parameters that can be assigned.

In this chapter both types of excitations are allowed in the K matrix and have directly interpretable effects. Firstly, self-excitations (κ_{ii} for $i = 1, 2$) represent the 'momentum' a team can experience after scoring a goal where

they go on to score another goal. Secondly, the cross-excitation terms (κ_{ij} for $i \neq j$) represents a teams attempt to reply to the conceded goal, either by working harder to score a goal or the other team relaxing after scoring and subsequently committing errors leading to conceding a goal.

The kernel functions $g_{ij}(t)$ of the matrix G follow the same interpretation, diagonal elements control the decay of the self-excitation and off diagonal elements control the decay of the mutual excitations.

The matrix K and its associated κ_{ij} values will be the main focus of this chapter. For the kernel matrix G an exponential distribution will be used, where each element is independent of each other

$$g_{ij}(t) = \beta_{ij} \exp(-\beta_{ij} t),$$

where β_{ij} is unknown and must be estimated. This is the standard kernel specification of the Hawkes process and provides a decreasing impact over time for each goal that is scored.

Bivariate Hawkes processes have been previously used to study the buy and sell orders of stocks in Muni Toke and Pomponio (2011) where there is excitations between the buy and sell orders as well as self-excitation. Likewise in Bowsher (2003) a bivariate Hawkes model is used to model both the timing of executed trades and the change in mid price of a stock.

The general intensity is given by Eq. (2.3) and a number of extensions are proposed to account for the dynamics of a soccer match. The algorithm used in Chapters 4 and 5 for sampling from the Hawkes process is updated to account for multiple event types.

6.3.2 Extending κ

Due to the nature of football, the impact of scoring a goal is unlikely to be constant over time. If a team scores early on in the match there is still the majority of the game left to play in which time the other team could score. In contrast, scoring close to the end of a match means that that the opponent will be unable to fight back due to the finite length of the match. There is also the

effect of scoring close to halftime, football folklore would have you believe that this is the best time to score a goal due to the psychological boost of being in the lead at the break.

To account for this variation a time component is used for the κ parameter in the K matrix

$$\log \kappa(t) \propto f(t),$$

where f is some function of t. The degree of dependence of t will be explored in this work and the choices for $f(t)$ will be considered later. The logarithm is taken to ensure positivity of the $\kappa(t)$ function which is necessary for posterior inference.

6.3.3 Accounting for Team Strengths

It is inaccurate to assume that each team will score a goal at the same rate. The total expected number of goals in a match will be dependent on the strength different between the two teams. This information needs to be incorporated into the model to account for what team is likely to score more goals and win the match. To account for this, the odds of each team winning the match and the odds of the draw will be used as covariates in the background rate.

The Pinnacle Sports closing odds for each match taken from `www.football-data.co.uk` will be used. The closing odds are believed to be the markets true opinion on the probabilities of the outcome of the match and ultimately a model-free estimate of the match outcome probabilities. This strategy also appeals to the efficiency of betting markets which has been studied in Franck et al. (2010) and Debnath et al. (2003). The changes in the betting odds will reflect the current information available about the outcome in the match and thus the last price before the match starts indicates the markets and thus best opinion on the outcome of the match.

For each team, the background rate depends on the odds of themselves winning, the other team winning, the odds of a draw and an indicator showing whether they are at home. For team i playing team j the background rate can

be written as

$$\log \mu_{ij} = \mu_0 + \alpha x_i + \beta x_j + \delta x_{\text{draw}} + \gamma x_{i\text{home}}, \tag{6.1}$$

where μ_0 is the intercept, x_i is the odds of team i winning, x_j is the odds of j winning, x_{draw} is the draw odds and $x_{i\text{home}}$ is the indictor parameter for team i playing at home. The parameters $\mu_0, \alpha, \beta, \delta$ and γ are unknown and must be estimated.

6.3.4 Posterior Inference

By interpreting the Hawkes process as a branching process a computationally efficient algorithm for sampling from the posterior distribution of the parameters can be derived. This algorithm has been detailed in the Section 4.4 and it is now extended further to account for multiple dimensions.

For a total of N_d events in d dimensions each event is labelled by t_{di}. The first index indicating what dimension the event occurs and second index indicates the ith event in that dimension. As previously stated, the background rate is as a $d \times 1$ vector of parameters, K and G are $d \times d$ matrices. Using the clustering representation each event can be assigned a tuple which indicates the parent event and the dimension of the parent. For example, if event t_{di} has parent t_{ej} then

$$B_{di} = (e, j).$$

If the event is caused by the background rate and has no identifiable parents then $B_{di} = (d, 0)$ as a background event in dimension d can only be caused by its own background rate. By using the parent labels, the branching structure is simulated and the unknown parameters of the Hawkes process inferred, namely the background rates μ_i, κ matrix K and kernel matrix G.

To simulate the branching structure, the probability of each events parent

event must be calculated

$$\Pr(B_{di} = (d, 0)) = \frac{\mu_d(t_{di})}{\int_t \lambda(t)\mathrm{d}\mathbf{t}},$$
$$\Pr(B_{di} = (d, j)) = \frac{\kappa_{dd} g_{dd}(t_{di} - t_{dj})}{\int_t \lambda(t)\mathrm{d}\mathbf{t}},$$
$$\Pr(B_{di} = (e, j)) = \frac{\kappa_{de} g_{de}(t_{di} - t_{ej})}{\int_t \lambda(t)\mathrm{d}\mathbf{t}},$$

then by sampling from these probabilities each event can be attributed with a parent and then given the parent labels, B_{di}, the rest of the Hawkes parameters can be sampled.

The algorithm must also be applied hierarchically as there are multiple timeseries that are being used to infer the unknown parameters. As such, another index m is included to account for each time series. Within the dataset there are M matches and thus an additional index for both the event times, t_{di}^m, and the parent labels B_{di}^m. By using the parent labels across all the matches, the posterior distributions of the parameters can be simulated from.

The number of child events that each event is responsible for and what the child event belonged to is used to infer the κ parameter. For this calculation we define a matrix N_{ij}^{child} of the same dimension as K where each element of N_{ij}^{child} is a row vector of length equal to the number of events in dimension i. Each element of the row vector counts the number of children events event t_{jl} is responsible for

$$N_{ijl}^{\text{child}} = \sum_{k=1}^{N_i} \mathbb{1}\left(B_{ik} = (j, l)\right),$$
$$N_{ijl}^{\text{child}} \sim \text{Poisson}(\kappa_{ij}),$$

this likelihood depends on the form of κ. In the case of a constant κ model, this is a simple posterior sample step. A conjugate prior is used to arrive at the posterior distribution

$$\kappa_{ij} \sim \text{Gamma}(\alpha_0^\kappa, \beta_0^\kappa),$$
$$\kappa_{ij} \mid N_{ijl}^{\text{child}} \sim \text{Gamma}\left(\alpha_0^\kappa + \sum_{l=1}^{N_i} N_{ijl}^{\text{child}}, \beta_0^\kappa + N_i\right),$$

which allows for direct sampling from the posterior distribution. For more complicated forms of κ the R package `rstanarm` is used. This allows for different forms of κ and then sampling the resulting posterior distribution numerically using Hamiltonian Monte Carlo (Goodrich et al., 2018).

A similar structure is used for the kernel update procedure. For each event with a parent event, define the shifted event time as

$$\tau_{ij} = t_{il} - t_{jk} \quad \text{if} \quad B_{il} = (j, k),$$

where τ_{ij} is the element of the matrix that contains the values used to update the kernel parameters in $g_{ij}(t)$. This means that each τ_{ij} is a vector of length p. In this work the exponential kernel is used for $g(t)$ which allows for a conjugate sampling step

$$\begin{aligned}
\tau_{ij} &\sim \text{Exp}(\beta_{ij}), \\
\beta_{ij} &\sim \text{Gamma}(\alpha_0^\beta, \beta_0^\beta), \\
\beta_{ij} \mid \tau_{ij} &\sim \text{Gamma}\left(\alpha_0^\beta + p, \beta_0^\beta + \sum_p \tau_{ij}\right),
\end{aligned}$$

this posterior distribution is easily sampled from given the parent labels B.

6.3.5 Background Covariates

The background events are used to infer the background rate parameters

$$\begin{aligned}
N_d^{\text{bg}} &= \sum_{m=1}^{M} \mathbb{1}(B_{d,i=0}^m), \\
N_d^{\text{bg}} &\sim \text{Poisson}(\mu_d).
\end{aligned} \tag{6.2}$$

If μ_d is constant, then a new sample can be taken by directly drawing from the posterior where μ_d has a $\text{Gamma}(\alpha_0^{\text{bg}}, \beta_0^{\text{bg}})$ prior distribution.

$$\mu_d \mid N_d^{\text{bg}}, B_{di} \sim \text{Gamma}(N_d^{\text{bg}} + \alpha_0^{\text{bg}}, M + \beta_0^{\text{bg}}).$$

In reality the background rate μ_d is used with a regression structure as specified in Eq. (6.1) in which case sampling from the posterior is more involved. This regression is estimated using `rstanarm` which provides simple Bayesian inference for the parameters in question and thus allows easy updating of the background rate for each match.

6.4 Results

The data provides the times of goals of both the home and away teams. As the goals are recorded to the closest minute, we modulate each goal occurrence with a randomly drawn uniform value between 0 and 1 which prevents any goal from occurring on the same time stamp.

The model is trained on 9264 matches and uses 3083 matches as the test set. The training data is randomly selected from all the data and consists of a varied mixture of European competitions: the top divisions in England, Germany, Spain, Italy and France. The second divisions of England and Germany are also included. The time range is from the 12/13 season up to the 17/18 season. The remaining data that is not used in the training set makes up the test set.

For all the fitted models the posterior samples of the parameters will be used to calculate the predictive likelihood of the matches in the test set and the best performing model will have the largest value of the likelihood. Also, simulations of a Hawkes process with samples of the parameters from the posterior distributions will be used to further check the models validity. Firstly, full matches, from start to finish, will be simulated to ensure that the final scores are inline with true match outcomes. This provides a check to ensure that our model is correctly specified. Then a toy match will be simulated to demonstrate how live information from when a team scores can be used to forecast the future scoring rates of teams in the match. Finally, the Hawkes model will follow a real match, updating the predictions based on when the goals are scored. These predictions will be compared to the live in-play odds of that match taken from a bookmaker.

6.4.1 Null Model

For the null model it will be assumed that both the home and away goals are generated from a Poisson process, each with independent parameters. These rates will be structurally identical to the background rates of the Hawkes process - taking into account the strength of the teams by using the Pinnacle

Sports closing odds. In each match team i is at home playing team j and the number of goals is Poisson distributed

$$N_{ij} = \text{Poisson}(\mu_{ij}),$$
$$N_{ji} = \text{Poisson}(\mu_{ji}),$$

where μ_i is defined at Eq. (6.1).

This model has no interaction between the teams or even time varying behaviour over the course of the match, therefore it is expected to perform poorly. From Table 6.1 the out-of-sample likelihood is much smaller than the Hawkes models and thus shows that the Hawkes models improve on this model. This type of baseline model is common to the previous works (Everson and Goldsmith-Pinkham, 2008; Karlis and Ntzoufras, 2003).

6.4.2 Constant κ

For the most basic Hawkes model the κ and kernel matrices are kept constant. This can be interpreted as a model where the impact of scoring a goal is constant throughout a match. This model is the equivalent to the null model but with an added clustering mechanism.

From the κ parameters in Table 6.1 it suggests that the self-exciting impact is less than that of the cross excitations therefore, a goal being scored increases the probability of the other team scoring more than their own scoring rate. Both κ_{12} and κ_{21} are very similar in value which shows that there is an equal cross-excitation effect for both the home and away team. The average impact of any goal is about 11 ($\frac{1}{\beta_{ij}}$) minutes and there is very little difference in the length of time the impact lasts based upon who scored it.

6.4.3 Linear κ

The time of the goals is now used as a covariate in the number of children events.

$$N^{\text{child}} \sim \text{Poisson}(\kappa(t)),$$
$$\log \kappa(t) = \beta_0 + \beta_1 t.$$

Table 6.1: Posterior means of the unknown parameters in the models. The training set 9264 matches and the test set is 3083 matches.

Parameter	Constant κ	Linear κ	Quadratic κ	Null
μ_0	0.240	0.237	0.227	0.234
α	-0.120	-0.129	-0.131	-0.100
β	0.080	0.074	0.073	0.085
δ	0.007	0.009	0.009	0.005
γ	0.048	0.025	0.013	0.045
κ_{11}	0.032	-	-	
κ_{12}	0.038	-	-	
κ_{21}	0.040	-	-	
κ_{22}	0.029	-	-	
β_{11}	0.086	0.052	0.058	
β_{12}	0.085	0.046	0.052	
β_{21}	0.082	0.048	0.050	
β_{22}	0.089	0.052	0.059	
Likelihood Out of Sample	-20529.97	-12261.99	-7792.331	-45326.11

This now allows for the impact of a goal on the intensity function to change over time. The logarithm is taken to ensure that $\kappa(t)$ remains positive.

Figure 6.3 shows that the impact of scoring a goal decreases with time for each element of K. There is a fairly large difference in the responses early on in the match, until roughly 50 minutes in, where the impacts converge to the same rate. This suggests that there is different behaviour in the response to home and away goals in the first half of the match. The largest impact is from the κ_{12} element which indicates that after the away team scores the home team is likely to score. In contrast, the smallest impact is the self-exciting of away team goals κ_{22}, therefore away teams do not capitalise on their scoring by scoring further goals. However, forcing the form of $\kappa(t)$ to be linear is

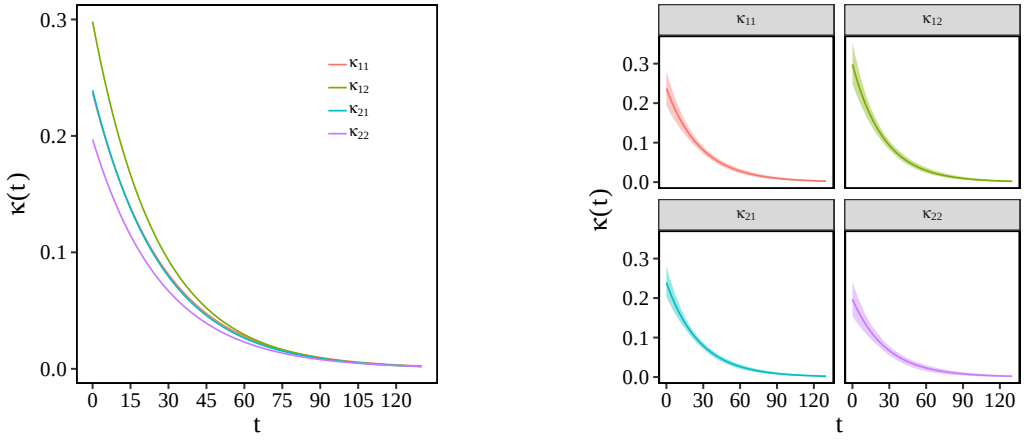


Figure 6.3: Linear $\kappa(t)$ results.

overly restrictive, instead more degrees of freedom are needed to show how this impact changes over time.

6.4.4 Quadratic κ

The complexity of κ is increased by including a second order term which allows for a more flexible shape in $\kappa(t)$. The linear model is extended with just the one additional term

$$N^{\text{child}} \sim \text{Poisson}(\kappa(t)),$$
$$\log \kappa(t) = \beta_0 + \beta_1 t + \beta_2 t^2,$$

again, the logarithm is taken to ensure positivity.

Figure 6.4 shows that there is a more complicated behaviour with $\kappa(t)$ and each term peaks close to halftime ($t = 45$). Interestingly, more independent behaviour is observed for each of the terms, each peak at slightly different locations and at different values. The κ_{12} parameter remains the largest as seen in the linear results and κ_{22} the smallest.

All three Hawkes models have shown that there is self-exciting behaviour and cross excitations between the goals scored by the home and away teams. When comparing the out-of-sample log likelihood values in Table 6.1 all three Hawkes models outperform the null model and the best fitting Hawkes model is that with quadratic dependence in the form of κ. The shapes of the quadratic

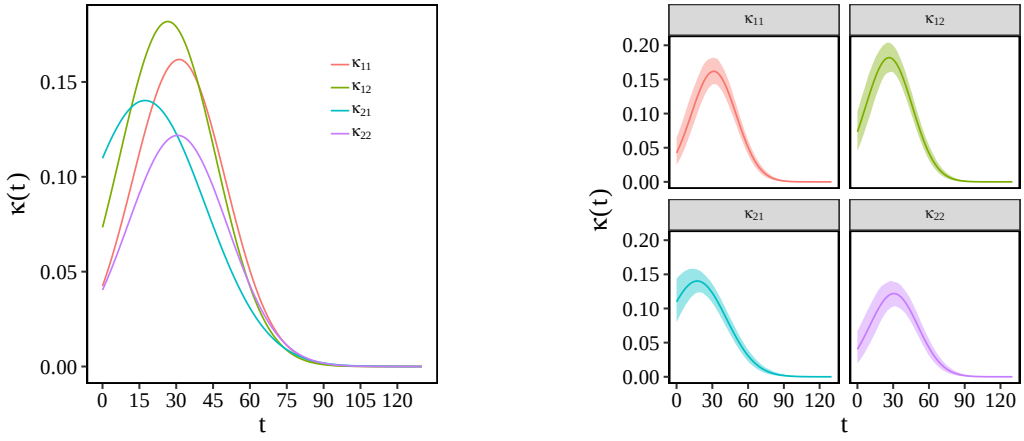


Figure 6.4: Quadratic $\kappa(t)$ results.

κ models agree with intuition that goals scored shortly before half time have the greatest impact on the total amount of goals in the game.

6.4.5 Posterior Simulations

To check the validity of our model a Hawkes process is simulated for each match in the test set. By comparing the resulting scores to the true scores the suitability of the model can be assessed.

From Figure 6.5 is can be seen that the model is allocating sufficient mass around the correct scores (indicated by the red dot). Even match 5, a high scoring 3-2, falls within the predicted scores of the model. However, the Hawkes model is not primarily designed to predict the final score of a match, instead, it provides a way of predicting the future response of teams after goals are scored. This allows for a 'live' model, where the subsequent changes in intensities can be updated as a goal is scored.

When comparing the Hawkes intensities for the different three models over a fictional game subtle differences emerge in the jumps of intensity due to the goals. Figure 6.6 shows the simulated intensities for the second half of a match where both teams scored in the first half. The home team scored early and the away team scored just before half time. From the constant model it can be seen that there is little difference in intensity spikes and the resulting simulated events in the second half are fairly uniform until the end of the match, the

intensity decays back to the base rate. For the linear model a large spike is observed due to the early goal, and a much smaller spike due to the second. The simulated intensity shows a heightened intensity early in the second half but steadily decays as no other goal is scored. For the quadratic model in Figure 6.6 there is a larger spike in intensity for the away team than the home team when the home team scores a goal and this type of difference is not seen in the other models. For the second goal, both increases in intensity are larger as expected from the curves found in Fig 6.4 as the κ parameter is larger at the 45 minute mark. For the simulated intensities, there is an expected increase in activity at the start of the half, with a larger increase for the home team, but again this decays back down to the end of the match. Overall, Figure 6.6 highlights the differences in the models and how the structure of κ can change the overall behaviour of the system.

6.5 In-play Odds

It has been demonstrated how the Hawkes model responds to goals and the probability of a team scoring will change on the previous goal times. This can now be compared to live betting odds using historical data provided by Betfair. By observing the change in odds throughout a match the probability of a team winning a match can be calculated and a good model will produce similar odds to the market.

The match in question is a Champions League group stage on 01-11-2017 between Tottenham Hotspur (Spurs) and Real Madrid. Notably, this match is very different from our training set. Tottenham were playing at Wembley whilst their new home stadium was built, therefore their home effect was uncertain. The match was also a group game in the Champions League contrasting the training set games as they were normal league competitions.

However, despite these differences, the Hawkes model performs well in replicating the behaviour of the match odds. In the match there were 4 goals, Tottenham scored three goals at 24, 56 and 65 minutes. Real Madrid scored

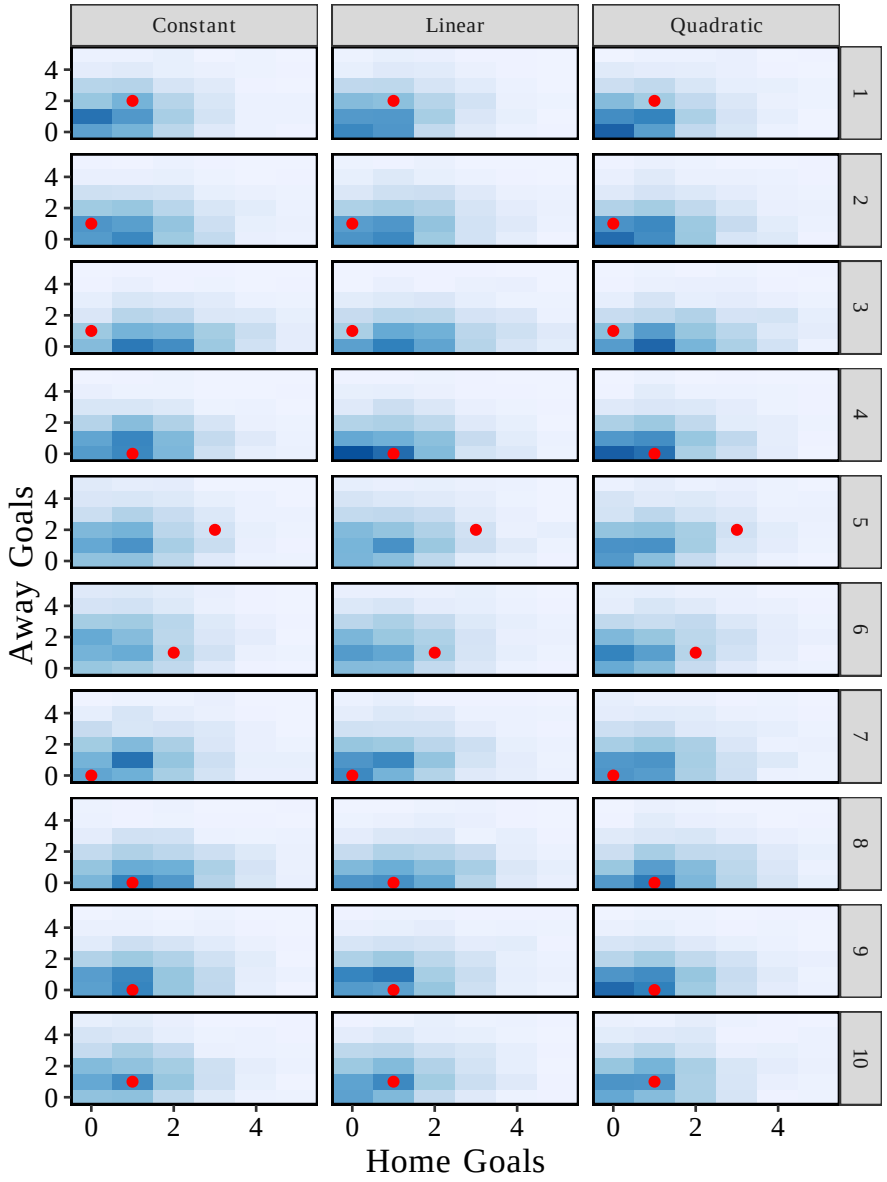


Figure 6.5: The density of scores for 10 matches in the test set. The red dots indicate the true outcome of the match.

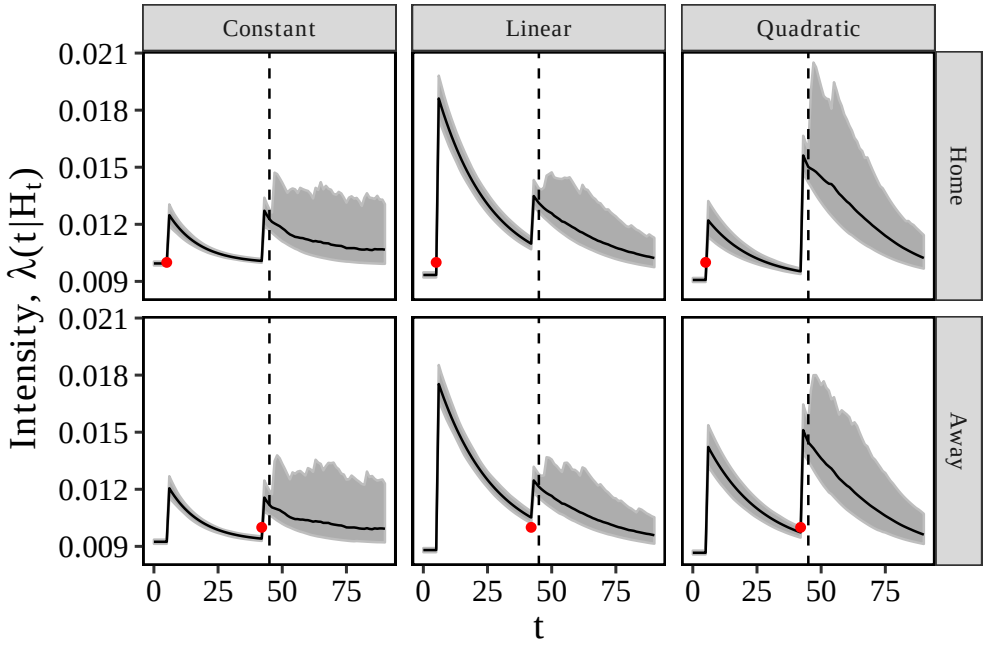


Figure 6.6: Live forecasting the intensity for the second half of a toy match where the home team scores one goal early in the first half and the away team scores a goal just before half time. The goals are indicated by the red dots.

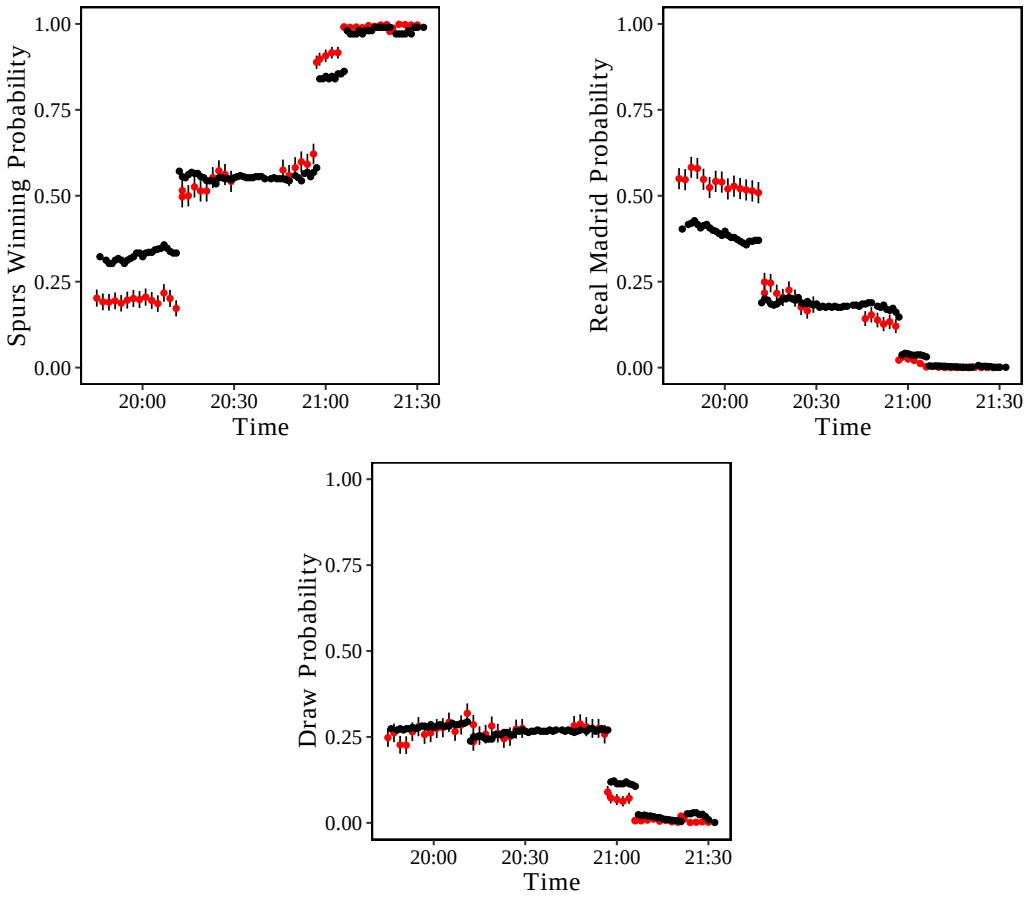


Figure 6.7: In-play odds. The black dots indicate the last traded price on Betfair which have been converted into a probability. The red dots are the estimated event probabilities from the Hawkes model. The lack of red dots betweeen 20:30 and 20:45 is due to half-time where prices are still traded but the Hawkes model prediction does not change.

a single goal in the 80th minute. At each of these goals the Hawkes model produces jumps that are comparable to the real jumps in the betting market.

To forecast the game probabilities the quadratic Hawkes process is used to simulate the match. Each simulation is computed with a different posterior sample of the parameters from the fitted model. Using the multiple simulations a probability that either team won or the draw occurred could be calculated with standard errors. The game is broken into 2 minute intervals and each probability is reported every two minutes. When a goal is scored, the simulations incorporate this new information by including the time of the goal. The outcome of the match is now forecasted including the goal and subsequent self-exciting effects from the Hawkes process. In the 90th minute it was revealed that 4 minutes of extra time would be played and thus the simulation was continued until the match was completed.

From Figure 6.7 we can see that after each Spurs goal there is a jump in the probability of them winning. Likewise, Real Madrid's probability of winning decreased with each goal conceded. There was a slight increase in the 80th minute when they scored a consolation goal. Similarly for the draw, the probability was increasing as the score remained 0-0, seeing a decrease with each subsequent goal being scored, but increased slowly in the periods between goals up until Spurs' third goal. This increase in draw probability during this period reflects the higher chance of a draw in a low scoring game. Interestingly though, there was a 3% increase when Real Madrid scored compared to the Hawkes 1% increase. This quickly decreased to zero as the match reached its conclusion.

It should be noted that the starting odds for the match winner being either team are quite different from the Betfair model in Figure 6.7. This is expected, as the prices are continually moving from when the market became available before the match to trade. Predicting how the market will move from the market opening to the match starting is another problem in itself and not within the scope of this chapter.

6.6 Conclusion

Overall a generative model has been built for the occurrence of goals in a soccer match. The multivariate Hawkes process is used such that goals can excite both their own scoring rate and the opposing teams scoring rate. This leads to behaviour where a team scoring a goal can cause a temporary increase in the probability that they will score again, plus there is also an increase in the probability that the opposing team will score. Both the size of these increases and the length of time they last for are free components of the model. Multiple models were explored where the actual impact of a goal being scored is a function of what time in the match that the goal is scored.

The model was fitted on a dataset of football matches across Europe. The best fitting model had a quadratic form of κ over time which produced a general shape where the maximum impact of a goal occurs around 45 minutes and each impact of a goal lasts for roughly 12 minutes.

To assess model performance, the log likelihood was calculated on a test set of data that was unseen in the training process. The quadratic model outperforms other Hawkes models and also improves on a null model without self-exciting behaviour. Using the posterior samples of the estimated parameters of the model, unseen matches are simulated and it is found that they produce sensible predictions for the final score.

It is also demonstrated that the model can be used to produce 'live' predictions of a match as it is being played. For this a Champion's League match between Tottenham Hotspur and Real Madrid is examined and the probability of either team winning, plus the draw is forecasted. After each goal is scored, the forecast is updated and it is shown that the Hawkes model is able to replicate market behaviour.

In conclusion, this work provides a Bayesian model that agrees with market behaviour. The multivariate Hawkes process shows that there is a change in team scoring rates both as goals are scored and conceded. Again, using the same type of algorithm for the previous chapters, the flexibility and extensi-

bility of the Hawkes approach has shown an improvement on modelling a new application.

Chapter 7

Discussion

This thesis has proposed and demonstrated a Bayesian method for estimating the parameters of a general Hawkes process. This algorithm exploits the conditional structure of the process to arrive at an efficient implementation that is independent of the form of the Hawkes process. From this algorithm, numerous extensions have been made to the standard Hawkes process including a nonparametric method for both the background rate and the kernel plus an extension into multiple variables. This work has improved both the statistical methodology in parameter inference of the Hawkes process and how using such inferences can be applied to a wide variety of situations.

Chapters 4 and 5 both used nonparametric forms of the kernel and background rate respectively which required different specifications of the Dirichlet process. This led to the development of the `dirichletprocess` package and Chapter 3 outlined the features of this R package, showing how it can accomplish many different nonparametric tasks. The main aim of the package is to provide a interface for users to build their own Dirichlet process models without needing to understand the mathematics or algorithms needed to fit such models. This has been achieved and the package enjoys a moderate success, receiving a citation in Koenker and Gu (2019) and roughly 15 downloads a month.

For the first application, Chapter 4, the Hawkes process was used to account for nonstationarity in the occurrence of extreme events. This type of

model provided an easy method for clusters to appear in the timings of extreme events by using the self-exciting property of the Hawkes process. This was extended further to include a nonparametric kernel which, by using a Dirichlet process, was an easy incorporation to the Bayesian estimation framework. When used on synthetic data it was able to recover a difficult predetermined shape and then when applied to a real dataset of extreme terror attacks the estimated kernel was particularly heavy tailed. Furthermore, the number of fatalities of these extreme terror attacks (the magnitude of the extreme event) was modelled using a GPD hierarchically based on their cluster allocation from the Hawkes process. It was found that partitioning the main and child events into different groups that were modelled hierarchically using a GPD provides a better model than one where all the fatalities are from a common GPD. Overall, the Hawkes process was able to show that terror attacks do display clustering and self-exciting behaviour.

The Hawkes process was then applied to predict trades in the foreign exchange market in Chapter 5. In the trading of currencies there are a number of phenomena that can be explained using a Hawkes process that is combined with a nonparametric model of the background rate. This time, by using a hierarchical Dirichlet process, the background rate of the Hawkes process was modelled nonparametrically across the different days of the week. The clear grouping in the data by day of the week meant that a hierarchical approach worked well as different days of the week were able to develop their own shape whilst pooling information to aid the inference of days where there was less data. The trades were also found to be self-exciting and the Hawkes process with a number of regression and autoregressive components in the background rate was the best performing model when fitted to the high frequency dataset. The conditional structure of the Hawkes process was able to adapt to changing market conditions and this was demonstrated when predicting the number of trades in five minute intervals on a selected day.

Chapter 6 involved extending the Hawkes process to include multiple vari-

ables and analyse how events occurring in a dimension can influence the events in both their own and other dimensions. This was another example of how the Hawkes inference algorithm can be used and flexibly extended into more than one dimension. This was applied to goals in a soccer match and in this case there are two dimensions, one for the home goals and one for the away goals. A multivariate Hawkes process allowed for the occurrence of a home goal to also affect the probability of an away goal. Different types of κ functions were explored which allowed the impact of a goal being scored by either team to vary over time. After fitting the model to a large data set of soccer matches it was found that a quadratic dependence on time for κ was the best fitting model. This specification indicated that the largest impact of a goal occurred around the halftime mark (after 45 minutes have been played) for both the home and away teams. The suitability of the model was also assessed by comparing the live probability of a team winning as the match was played. The true probability was calculated from the available betting odds and the predicted probability was calculated by simulating a Hawkes process and updating the prediction after each goal was scored. The Hawkes model was found to move in agreement with the market probability and suggests that the model is well suited in explaining the occurrences of goals.

The key advantage of these Hawkes models has been their flexibility. In each case, the model has started from the general intensity function, Eq. (2.2), before modifying a different component depending on the problem at hand. Then given this new intensity function, the inference algorithm is modified to account for these new components. Likewise, using the inferred parameters to make predictions is also consistent across the applications. Simulating forward and updating predictions based on the occurrences of events is the designed behaviour of a Hawkes process. Each prediction comes from a posterior distribution of parameter samples and therefore uncertainty around the prediction is obtained without any further work necessary highlighting the advantage of using a Bayesian model.

7.1 Future Work

Throughout this thesis the possible extensions and further investigation pathways have been highlighted in each chapter. These are now summarised.

The work in this thesis has demonstrated how the driving component of a Hawkes process can differ from application to application. In Chapter 4 it was the kernel that controlled the expressiveness of the model, in Chapter 5 is was the background rate controlling the variation due to the intraday patterns in the data and finally in Chapter 6 the κ parameter was explored due to the low individual activities of each match. There is further work to be done in understanding what types of data benefit most from the different variations in the Hawkes process and how the best forecasts can be obtained from selecting the correct model with the correctly adjusted component. This would then pave the way to a systematic model selection process that could correctly select the both the correct model specification of the individual components and the combination of the Hawkes components.

There is still much to be explored for Hawkes based models with multiple research areas where a Hawkes process could explain the clustering behaviour between events. One particular area of interest is cyber security. Internet services are built up of networks and consist of information requests being passed between networks and therefore a point process is well suited for modelling the arrival times between connection events. All three extensions to the Hawkes process discussed in this thesis could be applied to cyber security data to give insight into the clustering nature of attacks. Specifically, by using a multivariate Hawkes process to represent different network areas and the large amount of interacting processes the contagion between components of a network could be investigated and used to predict the likely spread of attacks.

The method of Hawkes process inference used in this thesis is also well positioned to benefit from the new frontiers of statistical and machine learning techniques. As it has been demonstrated, each iteration of inference involves partitioning the event times into parent and child, calculating the number

of children events and the relative time of events. The parameters of the model are then inferred from this partitioned data. It has been shown how both simple parametric models and more complicated Dirichlet process models can be used but this could be extended even further. Methods like Gaussian processes, neural networks or even deep learning could be used to infer the necessary component structure which makes the Hawkes process a model that can easily be updated to move with the direction of research.

The `dirichletprocess` package is at an advantage where future development directions can be aided by users and current discussions indicate that Indian buffet processes and their applications are a required feature. The Indian buffet process (IBP) is an extension to the Dirichlet process where the datapoints can belong to multiple clusters. It is a natural inclusion to the software package and thus makes it an easy goal for future development. In a wider context IBPs allows for more complex hierarchical models and feature based approaches to modelling.

Finally, the majority of the code used to perform the analysis in each chapter can be released as a general Hawkes package. It would compliment the `dirichletprocess` package and provide functionality to fit both the simple Hawkes models used and the more complex nonparametric models. There is also a gap in the market for a general package for performing Bayesian analysis of the Hawkes process as the current offering implement frequentist techniques.

Bibliography

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, and others. Tensorflow: A system for large-scale machine learning. In *12th Symposium on Operating Systems Design and Implementation*, pages 265–283, 2016.

L. Adamopoulos. Cluster models for earthquakes: Regional comparisons. *Journal of the International Association for Mathematical Geology*, 8(4):463–475, August 1976.

Ryan Prescott Adams, Iain Murray, and David JC MacKay. Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 9–16. ACM, 2009.

Bipin B. Ajinkya and Prem C. Jain. The behavior of daily stock market trading volume. *Journal of Accounting and Economics*, 11(4):331–359, November 1989.

Gordon J. Alexander, Amy K. Edwards, and Michael G. Ferri. The determinants of trading volume of high-yield corporate bonds. *Journal of Financial Markets*, 3(2):177–204, May 2000.

Charles E. Antoniak. Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

V. Armatas, A. Yiannakos, and P. Sileloglou. Relationship between time and goal scoring in soccer games: Analysis of three World Cups. *International Journal of Performance Analysis in Sport*, 7(2):48–58, May 2007.

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), May 2012. arXiv: 1112.1838.

Earvin Balderama, Frederic Paik Schoenberg, Erin Murray, and Philip W. Rundel. Application of Branching Models in the Study of Invasive Species. *Journal of the American Statistical Association*, 107(498):467–476, June 2012.

A. A. Balkema and L. de Haan. Residual Life Time at Great Age. *The Annals of Probability*, 2(5):792–804, 1974.

Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, April 1986.

Clive G. Bowsher. Modelling Security Market Events in Continuous Time: Intensity Based, Multivariate Point Process Models. SSRN Scholarly Paper ID 343020, Social Science Research Network, Rochester, NY, January 2003.

Andrew Bray and Frederic Paik Schoenberg. Assessment of point process models for earthquake forecasting. *Statistical science*, pages 510–520, 2013.

Emery N. Brown, Riccardo Barbieri, Valérie Ventura, Robert E. Kass, and Loren M. Frank. The Time-Rescaling Theorem and Its Application to Neural Spike Train Data Analysis. *Neural Computation*, 14(2):325–346, February 2002.

Bob Carpenter, Gelman, Andrew, Hoffman, Matt, Lee, Daniel, Goodrich Ben, Betancourt Michael, Brubaker Michael A, Michael, Guo Jiqiang, Li Peter, and Riddell Allen. Stan: A probabilistic programming language. *J Stat Softw*, 2016.

Lisbeth Carstensen, Albin Sandelin, Ole Winther, and Niels R. Hansen. Multivariate Hawkes process models of the occurrence of regulatory elements. *BMC Bioinformatics*, 11(1):456, September 2010.

V. Chavez-Demoulin and J. A. McGill. High-frequency financial data modeling using Hawkes processes. *Journal of Banking & Finance*, 36(12):3415–3426, December 2012.

V. Chavez-Demoulin, A. C. Davison, and A. J. McNeil. Estimating value-at-risk: a point process approach. *Quantitative Finance*, 5(2):227–234, April 2005.

V. Chavez-Demoulin and A. C. Davison. Generalized additive modelling of sample extremes. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 54(1):207–222, 2005.

Stuart Coles. *An Introduction to Statistical Modeling of Extreme Values*. Springer Series in Statistics. Springer London, London, 2001. ISBN 978-1-84996-874-4 978-1-4471-3675-0.

Daniel Cooley, Philippe Naveau, Vincent Jomelli, Antoine Rabatel, and Delphine Grancher. A Bayesian hierarchical extreme value model for lichenometry. *Environmetrics*, 17(6):555–574, September 2006.

D.J Daley and D Vere-Jones. *An Introduction to the Theory of Point Processes*. Probability and its Applications. Springer-Verlag, New York, 2003. ISBN 978-0-387-95541-4.

A. C. Davison and R. L. Smith. Models for Exceedances over High Thresholds. *Journal of the Royal Statistical Society. Series B (Methodological)*, 52(3): 393–442, 1990.

Sandip Debnath, David M. Pennock, C. Lee Giles, and Steve Lawrence. Information Incorporation in Online in-Game Sports Betting Markets. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, EC '03, pages

258–259, New York, NY, USA, 2003. ACM. ISBN 978-1-58113-679-1. event-place: San Diego, CA, USA.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977. Publisher: Wiley Online Library.

Mark J Dixon and Stuart G Coles. Modelling association football scores and inefficiencies in the football betting market. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 46(2):265–280, 1997.

Paul Embrechts, Thomas Liniger, and Lu Lin. Multivariate Hawkes processes: an application to financial data. *Journal of Applied Probability*, 48(A):367–378, August 2011.

Robert F. Engle. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 50(4): 987–1007, 1982.

Agner Erlang. The Theory of Probabilities and Telephone Conversations. *Nyt Tidsskrift for Matematik*, 20(B):33–39, 1909.

Michael D. Escobar and Mike West. Bayesian Density Estimation and Inference Using Mixtures. *Journal of the American Statistical Association*, 90(430): 577–588, June 1995.

Phil Everson and Paul S Goldsmith-Pinkham. Composite Poisson Models for Goal Scoring. *Journal of Quantitative Analysis in Sports*, 4(2), 2008.

Thomas S. Ferguson. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209–230, 1973.

Vladimir Filimonov and Didier Sornette. Quantifying reflexivity in financial markets: Toward a prediction of flash crashes. *Physical Review E*, 85(5): 056108, May 2012.

Egon Franck, Erwin Verbeek, and Stephan Nüesch. Prediction accuracy of different market structures — bookmakers versus a betting exchange. *International Journal of Forecasting*, 26(3):448–459, July 2010.

Andrew Gelman, Gareth O Roberts, Walter R Gilks, and others. Efficient Metropolis jumping rules. 1996.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian Data Analysis*, volume 2. Chapman & Hall/CRC Boca Raton, FL, USA, 2014.

S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, November 1984.

Frank Gerhard and Nikolaus Hautsch. Volatility estimation on the basis of price intensities. *Journal of Empirical Finance*, 9(1):57–89, 2002. Publisher: Elsevier.

Manfred Gilli and Evis Këllezi. An Application of Extreme Value Theory for Measuring Financial Risk. *Computational Economics*, 27(2-3):207–228, May 2006.

Ben Goodrich, Jonah Gabry, Imad Ali, and Sam Brilleman. *rstanarm: Bayesian applied regression modeling via Stan.* 2018.

Parameswaran Gopikrishnan, Vasiliki Plerou, Xavier Gabaix, and H. Eugene Stanley. Statistical properties of share volume traded in financial markets. *Physical Review E*, 62(4):R4493–R4496, October 2000.

John Graunt. Observations on the London Bills of Mortality. *Natural and Political Observations Mentioned in a Following Index, and Made Upon the Bills of Mortality. London: John Martyn and James Allestry*, 1973.

Oliver Grothe, Volodymyr Korniichuk, and Hans Manner. Modeling multivariate extreme events using self-exciting point processes. *Journal of Econometrics*, 182(2):269–289, October 2014.

Janos Gyarmati-Szabo. *Statistical extreme value modelling to study roadside air pollution episodes*. Ph.D., University of Leeds, 2011.

W. K. Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, 1970.

Alan G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, January 1971.

Alan G Hawkes. Hawkes processes and their applications to finance: a review. *Quantitative Finance*, 18(2):193–198, 2018. Publisher: Taylor & Francis.

Alan G. Hawkes and David Oakes. A Cluster Process Representation of a Self-Exciting Process. *Journal of Applied Probability*, 11(3):493–503, 1974.

Andreas Heuer, Christian Mueller, and Oliver Rubner. Soccer: Is scoring goals a predictable Poissonian process? *EPL (Europhysics Letters)*, 89(3):38007, 2010.

T Hsing, J Hüsler, and MR Leadbetter. On the exceedance point process for a stationary sequence. *Probability Theory and Related Fields*, 78(1):97–112, 1988.

Alejandro Jara, Timothy E Hanson, Fernando A Quintana, Peter Müller, and Gary L Rosner. DPpackage: Bayesian semi-and nonparametric modeling in R. *Journal of statistical software*, 40(5):1, 2011.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Dimitris Karlis and Ioannis Ntzoufras. Analysis of sports data by using bivariate Poisson models. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52(3):381–393, 2003.

Frank Keogh and Gary Rose. Football betting - the global gambling industry worth billions. *BBC*, October 2013.

Sinae Kim, Mahlet G Tadesse, and Marina Vannucci. Variable selection in clustering via Dirichlet process mixture models. *Biometrika*, 93(4):877–893, 2006.

J. F. C. Kingman. *Poisson Processes*. Clarendon Press, December 1992. ISBN 978-0-19-159124-2.

Roger Koenker and Jiaying Gu. Minimalist G-modelling: A comment on Efron, April 2019.

Athanasios Kottas. Dirichlet process mixtures of beta distributions, with applications to density and intensity estimation. In *Workshop on Learning with Nonparametric Bayesian Methods, 23rd International Conference on Machine Learning (ICML)*, 2006a.

Athanasios Kottas. Nonparametric Bayesian survival analysis using mixtures of Weibull distributions. *Journal of Statistical Planning and Inference*, 136 (3):578–596, March 2006b.

Athanasios Kottas and Bruno Sansó. Bayesian mixture modeling for spatial Poisson process intensities, with applications to extreme value analysis. *Journal of Statistical Planning and Inference*, 137(10):3151–3163, October 2007.

Athanasios Kottas, Ziwei Wang, and Abel Rodríguez. Spatial modeling for risk assessment of extreme values from environmental time series: a Bayesian nonparametric approach. *Environmetrics*, 23(8):649–662, December 2012.

Mehdi Lallouache and Damien Challet. The limits of statistical significance of Hawkes processes fitted to financial data. *Quantitative Finance*, 16(1):1–11, January 2016.

Jerald F Lawless. *Statistical models and methods for lifetime data*, volume 362. John Wiley & Sons, 2011.

MR Leadbetter. Weak convergence of high level exceedances by a stationary sequence. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 34(1):11–15, 1976.

Erik Lewis and George Mohler. A nonparametric EM algorithm for multiscale Hawkes processes. *preprint*, pages 1–16, 2011.

P. a. W. Lewis and G. S. Shedler. Simulation of nonhomogeneous poisson processes by thinning. *Naval Research Logistics Quarterly*, 26(3):403–413, 1979.

Thomas Josef Liniger. *Multivariate hawkes processes*. PhD Thesis, ETH Zurich, 2009.

David Lunn, David Spiegelhalter, Andrew Thomas, and Nicky Best. The BUGS project: Evolution, critique and future directions. *Statistics in Medicine*, 28(25):3049–3067, November 2009.

Steven N. Maceachern and Peter Müller. Estimating Mixture of Dirichlet Process Models. *Journal of Computational and Graphical Statistics*, 7(2): 223–238, June 1998.

Dean Markwick and Gordon J Ross. dirichletprocess: Build Dirichlet Process Objects for Bayesian Modelling, 2018.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, pages 6754–6764, 2017.

Xiao-Li Meng. Posterior Predictive p-Values. *The Annals of Statistics*, 22(3): 1142–1160, 1994.

George Mohler. Modeling and estimation of multi-source clustering in crime and security data. *The Annals of Applied Statistics*, 7(3):1525–1539, September 2013.

Michael Moore, Andreas Schrimpf, and Vladyslav Sushko. BIS Quarterly Review - December 2016, November 2016.

Ioane Muni Toke and Fabrizio Pomponio. Modelling Trades-Through in a Limited Order Book Using Hawkes Processes. SSRN Scholarly Paper ID 1973856, Social Science Research Network, Rochester, NY, 2011.

Radford M. Neal. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2): 249–265, June 2000.

John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965. Publisher: Oxford University Press.

Daniel Nevo and Ya'acov Ritov. Around the goal: Examining the effect of the first goal on the second goal in soccer using survival analysis methods. *Journal of Quantitative Analysis in Sports*, 9(2):165–177, 2013.

Paul J. Northrop and Philip Jonathan. Threshold modelling of spatially dependent non-stationary extremes with application to hurricane-induced wave heights. *Environmetrics*, 22(7):799–809, 2011.

Yosihiko Ogata. Statistical Models for Earthquake Occurrences and Residual Analysis for Point Processes. *Journal of the American Statistical Association*, 83(401):9–27, 1988.

Bernhard Pfaff and Alexander McNeil. *evir: Extreme Values in R*. 2018.

James Pitkin, Ioanna Manolopoulou, and Gordon Ross. Bayesian hierarchical modelling of sparse count processes in retail analytics. *arXiv preprint arXiv:1805.05657*, 2018.

Martyn Plummer. *JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling*. 2003.

Richard Pollard. Home advantage in soccer: A retrospective analysis. *Journal of Sports Sciences*, 4(3):237–248, December 1986.

Michael D. Porter and Gentry White. Self-exciting hurdle models for terrorist activity. *The Annals of Applied Statistics*, 6(1):106–124, March 2012.

Marcello Rambaldi, Paris Pennesi, and Fabrizio Lillo. Modeling foreign exchange market activity around macroeconomic news: Hawkes-process approach. *Physical Review E*, 91(1):012819, January 2015.

Jakob Gulddahl Rasmussen. Bayesian Inference for Hawkes Processes. *Methodology and Computing in Applied Probability*, 15(3):623–642, December 2011.

Dagfinn Rime and Andreas Schrimpf. The anatomy of the global FX market through the lens of the 2013 Triennial Survey. 2013.

Gordon J Ross. Nonparametric Bayesian Inference for the Hawkes Process with Seasonal Event Data, 2019.

John Salvatier, Thomas V. Wiecki, and Christopher Fonnesbeck. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science*, 2:e55, April 2016.

Fritz W Scholz and Michael A Stephens. K-sample Anderson–Darling tests. *Journal of the American Statistical Association*, 82(399):918–924, 1987.

Jayaram Sethuraman. A constructive definition of Dirichlet priors. *Statistica sinica*, pages 639–650, 1994.

David J. Spiegelhalter, Nicola G. Best, Bradley P. Carlin, and Angelika Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64 (4):583–639, 2002.

Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997. Publisher: Springer.

Nicholas Taylor. Trading intensity, volatility, and arbitrage activity. *Journal of Banking & Finance*, 28(5):1137–1162, 2004. Publisher: Elsevier.

Yee W Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Sharing clusters among related groups: Hierarchical Dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392, 2005.

Dustin Tran, Alp Kucukelbir, Adji B. Dieng, Maja Rudolph, Dawen Liang, and David M. Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

Alejandro Veen and Frederic 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.

Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Statistics and Computing*, 27(5):1413–1432, September 2017. arXiv: 1507.04544.

Ziwei Wang, Abel Rodrıguez, and Athanasios Kottas. A nonparametric mixture modeling framework for extreme value analysis. Technical report, Technical Report UCSC-SOE-11-26, University of California, Santa Cruz., 2011.

Jonathan Weinberg, Lawrence D. Brown, and Jonathan R. Stroud. Bayesian Forecasting of an Inhomogeneous Poisson Process With Applications to Call

Center Data. *Journal of the American Statistical Association*, 102(480): 1185–1198, December 2007.

Thomas Weise, Michael Zapf, Raymond Chiong, and Antonio J Nebro. Why is optimization difficult? In *Nature-inspired algorithms for optimisation*, pages 1–50. Springer, 2009.

Mike West. *Hyperparameter estimation in Dirichlet process mixture models.* Duke University ISDS Discussion Paper 92-A03, 1992.

Halbert White. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the Econometric Society*, pages 1–25, 1982.

Hadley Wickham. *Advanced r.* CRC Press, 2014.

Jianwen Zhang, Yangqiu Song, Changshui Zhang, and Shixia Liu. Evolutionary Hierarchical Dirichlet Processes for Multiple Correlated Time-varying Corpora. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, pages 1079–1088, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1.

Jiancang Zhuang, Yosihiko Ogata, and David Vere-Jones. Stochastic Declustering of Space-Time Earthquake Occurrences. *Journal of the American Statistical Association*, 97(458):369–380, June 2002.