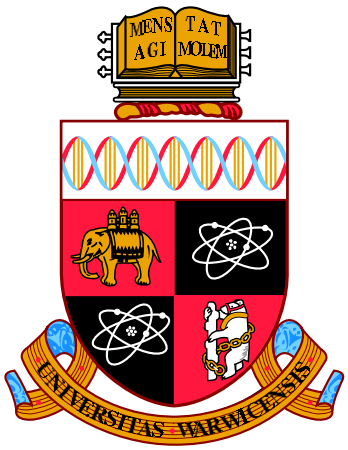
Bayesian Online Changepoint Detection

for Multivariate Point Processes

Jay Zern Ng

Supervised by Dr. Theo Damoulas

In partial fulfilment of
BSc Data Science



Department of Statistics
University of Warwick
April 2019

Dedicated to my mother and three siblings.

Acknowledgements

I wish to cordially thank my supervisor Dr. Theo Damoulas for his unwavering support, and for the valuable opportunities he has provided me throughout the year. Moreover, I wish to thank Jeremias Knoblauch for giving me access to his codebase and always helping me with any problems without hesitation. Lastly, a special thanks to Yannis Zachos and the team at Cervest for allowing me to work on their dataset.

Version Control

- *30th April 2019*: First submission.
- *8th June 2019*: Changed font size and line spacings.

Abstract

Bayesian On-line *Changepoint Detection* (CPD) is an active area of research in machine learning used as a tool to model structural changes that occur within ill-behaved, complex data generating processes. It has numerous applications in finance, health, and ecology. The goal of CPD is to detect abrupt changes in a time-series by partitioning it into identifiable sub-sequences; where the boundaries between two partitions are called *changepoints* (CP). Unlike previous Bayesian approaches in modelling CPs, *on-line* inference can be achieved by placing efficient prior beliefs over CPs to achieve a linear time and space complexity. In more recent developments of the state-of-the-art, *model selection* is proposed to solve problems such as model misspecification.

This thesis extends the Bayesian On-line CPD forefront by proposing a special model class for count data known as the Log Gaussian Cox Process, which falls under the realms of non-parametric Bayesian methods. We study the many attractive properties of this model including flexibility, and discuss various approximation methods to overcome issues in intractability. Next, we attempt to speed-up the model using sparse approximations; and extend this to multivariate streams. To benchmark the performance of this model, we apply it on two real-world and one synthetic dataset.

Keywords: Changepoint detection, Cox processes, non-parametric Bayesian statistics, Gaussian Processes, multi-task learning

Nomenclature

Standard notations apply to this thesis. Lowercase symbols x refer to scalar values. Bold symbols $\boldsymbol{x}$ refer to vectors. Uppercase bold symbols $\boldsymbol{X}$ refer to a matrix. The i-th element of a vector $\boldsymbol{x}$ is x_i, and the (i, j)-th element of a matrix $\boldsymbol{X}$ is X_{ij}.

The notation throughout this thesis varies depending on the context. For example, there may be an additional underscript or superscript to denote variables such as time t. Different symbols for probability densities $\mathbb{P}$ may be used instead such as $p(\cdot), q(\cdot), \pi(\cdot)$ for easier distinctions. Some variables have short-hard expressions such as kernel functions used in Chapters 3 and 5.

$\boldsymbol{\theta}_V$	Vector of variational parameters
$\boldsymbol{\Theta}_{m_t}$	Parameter space of model m at time t
$\boldsymbol{\theta}_{m_t}$	Parameter vector of model m at time t
$\boldsymbol{A}$	Laplace approximation precision matrix
$\boldsymbol{B}_q$	Coregionalisation matrix for group q
$\boldsymbol{f}$	Latent function
$\boldsymbol{K}_{\boldsymbol{fu}}$	Kernel function for $\boldsymbol{f}$ evaluated at $k(\boldsymbol{X}, \boldsymbol{Z})$
$\boldsymbol{K}_{\boldsymbol{f_*f}}$	Kernel function for $\boldsymbol{f}$, evaluated predictive points $\boldsymbol{X}_*$, $k(\boldsymbol{X}_*, \boldsymbol{X})$
$\boldsymbol{K}_{\boldsymbol{ff}}$	Kernel function for $\boldsymbol{f}$ evaluated at $k(\boldsymbol{X}, \boldsymbol{X})$
$\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$	Kernel function for a vector-valued function
$\boldsymbol{U}$	Set of inducing points
$\boldsymbol{u}$	r.v. for inducing points
$\boldsymbol{y}_{1:t}$	Collection of data from time 1 to t

$\boldsymbol{Z}$	Set of inducing inputs
Cov	Covariance
$\cup$	Union
ℓ	Lengthscale
$\emptyset$	Empty set
$\exists$	There exists
$\forall$	For all
$\hat{\boldsymbol{Y}}_{t+h}$	r.v. for h-step ahead prediction
$\hat{\boldsymbol{y}}_{t+h}$	h-step ahead prediction
$\hat{\boldsymbol{y}}_t^h$	h-step ahead predictive interval
$\iff$	If and only if
$\implies$	Implies
∞	Infinity
$\int$	Integral
$\mathbb{1}(\cdot)$	Indicator function
$\mathbb{E}$	Expectation
$\mathbb{N}$	Natural numbers
$\mathbb{P}$	Probability density, unless given a specific notation such as $p(\cdot)$ or $q(\cdot)$ etc.
$\mathbb{R}$	Real numbers
$\mathcal{GP}$	Gaussian Process
$\mathcal{KL}$	Kullback Leibler divergence
$\mathcal{M}$	Model universe
$\mathcal{N}$	Gaussian or Normal distributed
$\mathcal{O}(\cdot)$	Big-Oh
$\otimes$	Kronecker product

$\prod$	Product
σ^2	Noise variance
σ^2_{signal}	Signal variance
$\sim$	Distributed as
$\sum$	Sum
$\mathrm{Tr}(\cdot)$	Trace function
$\triangleq$	Equal by definition
$f_d(\boldsymbol{x})$	d-associated output of a vector-valued function
H	Hazard
m_t	Model at time t
MAP_t	Maximum A Posteriori segmentation at time t
Q	Group size of $f_d(\boldsymbol{x})$
R_q	Sample size of $f_d(\boldsymbol{x})$ at group q.
r_t	Run-length at time t
S_t	Segmentation at time t
t	Time
r.v.	Random variable

Acronyms

BOCPDMS Bayesian On-line Changepoint Detection with Model Selection.

CAVI Coordinate Ascent Variational Inference.

CP Changepoint.

CPD Changepoint Detection.

CUSUM Cumulative Sum.

ECMWF European Center for Medium Range Weather Forecasts.

ELBO Evidence Lower Bound.

EP Expectation Propagation.

GP Gaussian Process.

ICM Intrinsic Coregionalization Model.

LGCP Log Gaussian Cox Process.

LHS Left Hand Side.

LMC Linear Model of Coregionalization.

MAP Maximum A Posteriori.

MCMC Markov Chain Monte Carlo.

MD Multinomial Dirichlet.

PG Poisson Gamma.

PPM Product Partition Model.

PSD Positive semi-definite.

RBF Radial Basis Function.

RHS Right Hand Side.

SoS Sum of Separable.

SVI Sparse Variational Inference.

TSA Time-series Analysis.

VI Variational Inference.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Objectives:

- ✓ Discussing modern applications/challenges in change-point detection.
- ✓ Outlining the historical development of change-point detection, i.e. evolutions from being *off-line* to *on-line*, and contrasts between *Frequentists* and *Bayesians*.
- ✓ Motivating the need for non-parametric Bayesian methods.
- ✓ Stating the personal contributions of this thesis.

Time-series analysis (TSA) has become increasingly important in diverse fields such as finance, medicine and ecology, for which a *time-series* (TS) is a sequence of measures over time describing behaviours of systems (Aminikhanghahi and Cook, 2016). These behaviours are often complex with uncertainty - which in turn requires scientists to incorporate sophisticated dynamic requirements to try to model it (Montanez et al., 2015). Moreover, these complex behaviours can change over time due to external factors, or internal systematic changes of its distribution. Over the last decade, the pertinence of TSA has increased *even more so*, given the proliferation of large datasets and technological advancements in data mining. This necessitates the need for new state-of-the-art tools to better understand the complexities of real-world TS data in the modern world.

Unlike traditional approaches, an alternative tool in TSA that has gained particular traction in recent years (Adams and MacKay, 2007; Saatchi et al., 2010; Fearnhead and Liu, 2007) is time-series segmentation or *changepoint detection* (CPD). Specifically, partitioning a time-series into identifiable sub-sequences to reveal the underlying properties of its source - for which the boundary between two partitions are defined as a *changepoint* (CP). This is motivated by the

fact that data streams are often *non-stationary* (Barry and Hartigan, 1992); or a *stochastic process* whose joint probability is constantly changing in a dynamic environment. Rather than trying to forecast future patterns of a TS, the intuition is that CPD negates the unrealistic assumption that the behaviour of a time-series is standardized across time; and that CPs indicate structural changes that happen within a *data generating process*. According to Aminikhanghahi and Cook (2016), CPD can be thought of as an algorithm used to identify abrupt changes in a data sequence.

1.1 Applications

The best and quickest way to grasp the usefulness of CPD is through highlighting a few examples. In general, CPD is applicable in any form of time-series data. That being said some interesting use-cases include:

Climate change detection. Specifically on climate analysis and monitoring factors such as CO_2-levels that affect the weather, so that stakeholders including farmers and agricultural workers can be notified in advance to avoid disastrous effects. Prediction methods that utilize CPD has become increasingly important over the last decade to the possible occurrence of climate change as a result of an increase in greenhouse gases in the atmosphere (Aminikhanghahi and Cook, 2016).

Medical condition monitoring. Using CPD as a tool to monitor a patients' physiological health such as measuring heart rates to detect possible illnesses. Further, using an electroencephalogram (EEG) and electrocardiogram (ECG) for real time monitoring (Bodenstein and Praetorius, 1977). It may also be used to analysed other medical issues including insomnia or epilepsy using an magnetic resonance imaging (MRI), in regards to trying to understand activities in the human brain (Aminikhanghahi and Cook, 2016).

Econometrics and finance. Being able to measure the volatility of the US economy, changes to the quarterly GDP growth rate or persistence in yearly inflation rates (M. Koop and M. Potter, 2004). In more specific use-cases for example, Chen and Gupta (1997) applied CPD on stock market returns on a 1971-1974 US stock dataset by modelling returns as a chain of hypothesis tests and using different information criterions to assess structural changes. Chen and Gupta (1997) was able to map key real-world events such as strikes by the United Transportation Union in July 1971 causing US gold stocks to fall, wage-price announcements by President Nixon in August 1971, and even price increases in important industrial products in December 1971.

Apart from the above applications, CPD may also be applied on unique problems such as speech recognition, image analysis and human activity analysis (Aminikhanghahi and Cook,

2016). In the next section, we will briefly cover the history of CPD for the reason that it is important to be aware of its several key historical transitions, and the motivation behind certain dichotomies in being on-line versus off-line; Bayesians versus Frequentists.

1.2 Literature

Historically, CPD dates back to the early 1950s when it was predominantly motivated by *Frequentists* approaches. In particular, most Frequentists methods at the time were executed within an *on-line* settings which meant that new observations were used to dynamically update its predictors.

The first pioneering work by Page (1954) introduced the idea of calculating the cumulative sum (CUSUM) to gauge the distribution of some parameter of interest. By choosing some information criterion, Page (1954) first proposed the use of control charts and hypothesis tests; if the criterion exceeds some threshold then this signifies a change. Eventually, other authors took inspiration. Lorden (1971) tried to solve the problem of 'optimal stopping' in CPD by proving the asymptotic optimality of the CUSUM for a worst case detection delay. In more recent developments Desobry et al. (2005) challenged the conventional Generalized Likelihood Ratio tests and came up with the idea of using Support Vector Machines. Specifically, Desobry et al. (2005) proposed building a dissimilarity measure between past and future set of descriptors.

Contrastingly, the development of *Bayesian* CPD - which allows for principled ways of uncertainty quantification only arrived later in the mid 1960s. Unlike Frequentists methods these were executed in an *off-line*, *retrospective* or *batch-mode* setting, which meant they faced issues in scalability and practicality in real-world data.

Amongst the earliest methods, Chernoff and Zacks (1964) proposed using a priori probability distribution of change at each point in time. Smith (1975) later made further contributions by looking into the single change point problem when different knowledge of the parameters of the underlying distributions is available. In the 1940s, Stephens (1994) described a popular retrospective multiple changepoint identification technique, which utilises sampling-based techniques or the Gibbs sampler for inference. Other notable approaches that were particularly influential include Chib (1998)'s proposal of using the Hidden Markov Model to model CPs, Barry and Hartigan (1992)'s method of partitioning a data into contiguous sets with constant parameter values in each set, and Xuan and Murphy (2007)'s attempt to model the joint density of vector-valued observations using undirected Gaussian graphical models.

In more recent developments of Bayesian CPD, *on-line* approaches were finally introduced by Adams and MacKay (2007) and Fearnhead and Liu (2007) when it became especially clear that the Bayesian computational bottlenecks where indeed impractical in meeting modern de-

mands of scalability. It was the former whom proposed the idea of placing efficient priors over a parameter known as a 'run-length'. Then by deriving a recursion that can be executed using dynamic programming, Adams and MacKay (2007) demonstrated that a linear time complexity can be achieved under a Bayesian setting. This eventually inspired the works of others, such as Saatchi et al. (2010)'s proposal of using non-parametric Gaussian Process models to address time-series behaviours where temporal correlations in a regime are expected. In the latest iteration of Bayesian On-line CPD, Knoblauch and Damoulas (2018) proposed the idea of integrating model selection to address problems such as model misspecification by combining the works of Adams and MacKay (2007) and Fearnhead and Liu (2007).

1.3 Challenges

In the last section, it is evident that traditional CPD methods have constantly evolved over the last few decades in response to address new challenges in modern TS data. Here, we list down three specific challenges that are highly relevant to this thesis; and are reoccurring themes that will be revisited in Chapters 3, 4 and 5.

Multivariate streams. Modelling dependencies within multiple data streams is common in any real-world TS. In addition to the temporal correlation of data, there is the spatial correlation of data to consider which encodes an abundance of information between data generating processes. When the temporal and spatial correlations are considered simultaneously, we call this a *spatio-temporal* TS.

Robustness. Differentiating between structural changes in a data generating proccess and *outliers* or *anomalies*. Real-world time-series data is often plagued with outliers which cause traditional CPD to fail. For example, when former Prime Minister David Cameron announced the Brexit referendum in June 2016, this triggered a short-term spike in UK housing transaction volume. In Chapter 4, this is one of the case-studies covered in this thesis.

Scalability. It is clear that for certain applications such as finance and investing, speed becomes a top priority; where as for other applications such as climate change detection this may not be the case. Regardless, computational speed is still always desirable, and necessary for real-time inference in both TSA and CPD, as real-world datasets are becoming too large for traditional methods to work.

In this thesis, we attempt to showcase how to *partially* circumvent these challenges by deploying a number of techniques in existing literature.

1.4 Contributions

This thesis follows the work of Knoblauch and Damoulas (2018) which is limited to the scope of *continuous data* only. In particular, we extend the Bayesian On-line CPD framework by implementing the following routine using *count data* instead.

1. Proposed a non-parametric model used in spatio-temporal point processes.
    - Implemented in Python3.6 using external libraries GPy[1] and GPFlow[2].
    - Inference via three approximation methods, discussed in Chapter 3.
2. Reduced the computational overheads of the model using sparse approximations.
3. Extended the model from a univariate to a multivariate setting.
4. Tested the model using two real-world and one synthetic dataset.

1.5 Synopsis

This thesis covers basic concepts at the beginning of each chapter, and progresses with more advanced material later on. Chapter 2 aims to cover key concepts that help set up a foundation for the remainder of this thesis. Next, it rigorously defines important quantities within the Bayesian On-line CPD algorithm in reference to three different authors. It then draws a conclusion by discussing how these important quantities can be used to generate three different outputs. Chapter 3 implicitly introduces a specific model class for spatio-temporal data, through reviewing Point Processes and Gaussian Processes. Then, it proceeds to discuss about issues with intractability and how to overcome it using approximations. Chapter 4 showcases the strengths and weaknesses of the model by applying it on a real-world and synthetic dataset. Chapter 5 takes a detour by demonstrating multivariate extensions using a concept known as multitask-learning. Finally, Chapter 6 closes the thesis by summarising its main points and proposing future directions.

[1] GPy sourcecode available here.
[2] GPFlow sourcecode available here.

Chapter 2

Bayesian On-line Changepoint Detection

Objectives:

- ✓ Outlining key concepts in Bayesian inference and TSA.
- ✓ Understanding the implications of placing CP priors for probabilistic recursions.
- ✓ Motivating the need for model selection.
- ✓ Describing the algorithm's output and specific details of how CPs are declared.

2.1 Preliminaries

We start this chapter by recapitulating important concepts in Bayesian inference and time-series analysis (TSA).

A deep rooted history: from inception in the original paper in 1763, until the posthumous emergence of 'Bayesian' philosophy in the 1970s (Fienberg, 2006). Bayesian inference offers an elegant way to express beliefs; one that mimics the natural thought process of humans when presented with new information.

Consider a random variable (r.v.) θ on any parameter space Θ. Objectively, the goal is to learn about $\theta \in \Theta$ by accumulating information that is relevant. Given a *probability density function* $p(\cdot)$, this is achieved by encoding a *prior* distribution $p(\theta)$ to represent our initial belief. Simply, update this belief using data y in observation space $\mathcal{Y}$. This is done by computing the *likelihood* of y given θ, which verifies $\int_{\mathcal{Y}} p(y|\theta)dy = 1$. Let the joint distribution on space $\Theta \times \mathcal{Y}$ be written as products of conditional probabilities in two ways: $p(\theta, y) = p(y|\theta)p(\theta) = p(\theta|y)p(y)$.

Manipulating these terms leads to the heart of Bayesian inference, known as *Bayes' rule*. Denote the *posterior* distribution as $p(\theta|y)$; the belief after observing data y, then it follows that

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} = \frac{p(y|\theta)p(\theta)}{\int_\Theta p(y|\theta')p(\theta')d\theta'} \tag{2.1}$$

where $p(y)$ is the *evidence* or *marginal likelihood* which does not depend on θ, usually considered a normalising constant. Conventionally, Bayes rule is written in proportional terms where $p(\theta|y) \propto p(y|\theta) \times p(\theta)$.

Thus far, the posterior distribution contains an abundance of useful information, which leads to the question of how might one perform inference using this. A trivial method for capturing this information is via point estimates of the mode. This is commonly referred to as the *Maximum A Posteriori* (MAP) estimate, or in words: pick the optimal θ that maximises our posterior beliefs.

$$\theta_{\text{MAP}} = \arg\max_{\theta\in\Theta} p(y|\theta)p(\theta) \tag{2.2}$$

Principally, to obtain a *closed-form* expression for the posterior distribution without resorting to numerical integration, *conjugacy* must be exploited (Schlaifer and Raiffa, 1961). By definition, a class of prior distributions $\mathcal{P}$ is said to be conjugate for a class of likelihood functions $\mathcal{F}$ if $p(\boldsymbol{\theta}|\boldsymbol{y}) \in \mathcal{P}$ for all $p(\boldsymbol{y}|\boldsymbol{\theta}) \in \mathcal{F}$ and $p(\boldsymbol{\theta}) \in \mathcal{P}$. Conjugacy has later implications in both chapters 2 and 3.

Being Bayesian as opposed to Frequentists connotes several consequences. Probabilistically, it has advantages such as representing uncertainty better, and being less prone to overfitting due to the posterior distribution being able to capture information via parameters. However, at times, selecting a model prior might prove difficult when trying to represent all spectrums of real-world data. This motivates the need for *model selection*; one of the key themes in CPD discussed later on in this chapter.

2.2 Assumptions

For the entirety of TSA, define a time-series (TS) described by a *generative* model as $\boldsymbol{y}_{1:t} = (y_1, \ldots, y_t)$ or compactly $(\boldsymbol{y}_t)_{t\in\mathbb{N}}$ as a vector. Further, denote a spatio-temporal TS by a spatial grid $s \coloneqq (s_1 \times s_2)$ such as in Figure (2.2), where the (i, j)-th position contains a univariate TS. The edges appearing represent an association between two vertex; whereby a close vicinity implies dependencies between pairs (Zachos, 2018). This associative mapping is particularly useful in studying many real-world phenomenons.

Being *generative* as opposed to *discriminative* implies that the hidden distribution of $\boldsymbol{y}_{1:t}$ is known; whilst the parameters $\theta_m \in \Theta_m$ for some model $m \in \mathcal{M}$ are unknown.

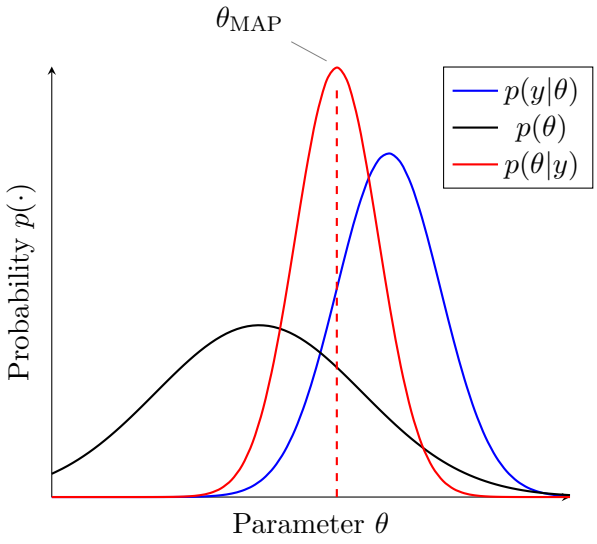


Figure 2.1: Illustration of Bayesian inference using normal distributions for all families of distributions.

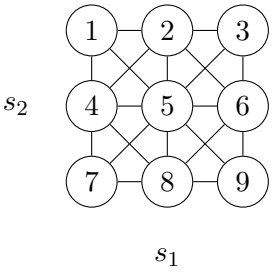


Figure 2.2: Illustration of a spatio-temporal time series with spatial dimensions $s_1 = 3$, $s_2 = 3$

As one of the core prerequisites for the CPD framework, we assume that a TS can be structured as a piecewise (weakly) stationary process. *Stationarity* which was proposed by Priestley (1981) falls under two precise definitions.

A time-series $\boldsymbol{y}_{1:t}$ is *strictly* stationary if the joint distribution verifies

$$p(y_1, \ldots, y_t) = p(y_{1+\delta}, \ldots, y_{t+\delta}) \quad \forall t \in \mathbb{N} \tag{2.3}$$

meaning that it is invariant under time shift $\forall\ \delta \in \mathbb{N}$. Implicitly, strict-sense stationarity implies the next definition: A time-series $\boldsymbol{y}_{1:t}$ is *weakly* stationary if $\mathbb{E}(\boldsymbol{y}_t)$ and $\text{Var}(\boldsymbol{y}_t)$ is constant, and the auto-covariance is

$$\text{Cov}(\boldsymbol{y}_t, \boldsymbol{y}_{t+\delta}) = \mathbb{E}([\boldsymbol{y}_t - \mathbb{E}(\boldsymbol{y}_t)][\boldsymbol{y}_{t+\delta} - \mathbb{E}(\boldsymbol{y}_{t+\delta})]) = \gamma_\delta \quad \forall t, \delta \in \mathbb{N} \tag{2.4}$$

Given a (weakly) stationary time series, it follows that the auto-correlation can also be deduced to be independent of time given by:

$$\rho_\delta = \frac{\mathbb{E}([\boldsymbol{y}_t - \mathbb{E}(\boldsymbol{y}_t)][\boldsymbol{y}_{t+\delta} - \mathbb{E}(\boldsymbol{y}_{t+\delta})])}{\sqrt{\text{Var}(\boldsymbol{y}_{k+\delta})}\sqrt{\text{Var}(\boldsymbol{y}_k)}} = \frac{\gamma_\delta}{\text{Var}(\boldsymbol{y}_0)} \tag{2.5}$$

Another important prerequisite for the CPD framework postulates that observations may be divided into non-overlapping partitions, adhering to the *product partition model* (PPM) (Barry and Hartigan, 1992). Given a segmentation $S = \{S_1, \ldots, S_b\}$ with b number of segments, the PPM asserts the following. Segmentations are independent and identically distributed (*i.i.d.*) r.v. meaning that their distributions can be expressed as:

$$\mathbb{P}(S_i, S_j) = \mathbb{P}(S_i)\mathbb{P}(S_j) \quad \forall i \neq j \quad i, j \in \{1, \ldots b\}$$

Secondly, segmentations can be written as a product of non-negative cohesions, $\mathbb{P}(S) = K \prod_{i=1}^{b} \mathbb{P}(S_i)$, with some normalizing constant K. Lastly, the transition probabilities of S can be expressed as $\mathbb{P}(S) = p_{0,i_1} p_{i_1,i_2}, \ldots, p_{i_{b-1},i_b}$ where delineations between two segments define a CP.

2.3 Probabilistic formulations

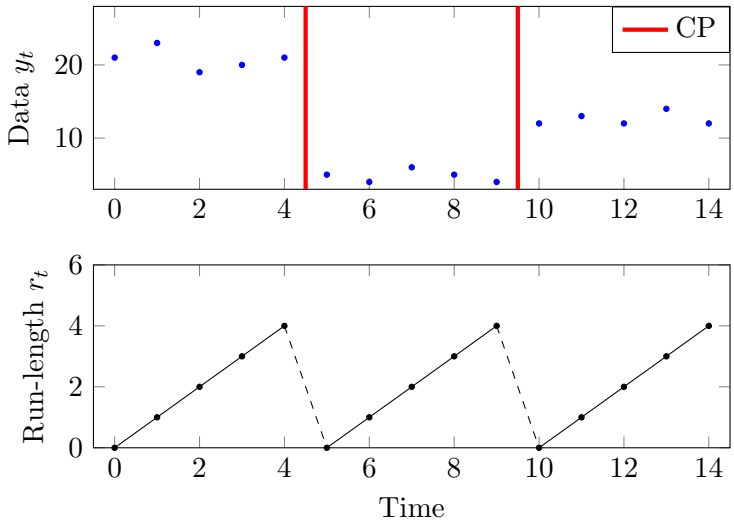


Figure 2.3: Illustration of two CPs with plots data (top) and run-lengths (bottom) versus time. Bottom plot: solid lines indicate run-lengths growing; dashed lines indicate the run-length truncates to zero

The assumptions and concepts declared in the previous section give rise to the original Bayesian CPD framework by Adams and MacKay (2007). Conceptually, the framework revolves around an approach known as *causal predictive filtering* for on-line inference rather than *retrospective segmentation*. In many applications of machine-intelligence, Adams and MacKay (2007) argues that this is a natural requirement. For example, 'agents' in any autonomous systems must navigate through environments whilst being able to quickly respond to changes dynamically.

To start, the framework introduces a new r.v. known as a *run-length* denoted by r_t at time t. A run-length either grows by one or truncate to zero $\implies \exists$ a one-to-one correspondence between run-lengths and CPs. This leads to the fundamental recursion:

$$r_t = \begin{cases} 0 & \text{a CP is declared at time } t \\ r_{t-1} + 1 & \text{otherwise} \end{cases} \tag{2.6}$$

An alternative way of making sense of Equation (2.6) is that if $t = r_t \iff$ a CP was declared at time $t - r_t$. Inference is done by tracking only the most recent CP at time $t - r_t$

to avoid retrospection - hence being coined *on-line*. The consequences of doing so serves an important role in reducing the time complexity of CPD to $\mathcal{O}(t)$. When comparing historical Bayesian approaches on CPD, the time complexity was previously factorial in $\mathcal{O}(\prod_{i=1}^{t} i)$ because all possible permutations of run-lengths must be considered in the past. This accentuates the importance of the computational gains in tracking the most recent CP.

In a familiar setup, any Bayesian framework requires specifying priors over parameters of interests to encode beliefs as outlined in the preliminaries. In this case, let H, π and f be probability densities over run-lengths, model parameters and observation density. Then construct the prior beliefs as follows:

$$r_t|r_{t-1} \sim H(r_t, r_{t-1}) \qquad \forall r_t \in \mathbb{N} \tag{2.7}$$

$$\boldsymbol{\theta} \sim \pi(\boldsymbol{\theta}) \qquad \forall \boldsymbol{\theta} \in \boldsymbol{\Theta} \tag{2.8}$$

$$\boldsymbol{y}_t|\boldsymbol{\theta} \sim f(\boldsymbol{y}_t|\boldsymbol{\theta}) \qquad \forall \boldsymbol{y}_t \in \mathbb{R}^s \tag{2.9}$$

where the prior beliefs for CPs are

$$p(r_t|r_{t-1}) = \begin{cases} 1 - H(r_{t-1}+1) & \text{if } r_t = r_{t-1}+1 \\ H(r_{t-1}+1) & \text{if } r_t = 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

for a Hazard function $H : \mathbb{N} \to [0, 1]$. A Hazard function, which falls under a branch of statistics known as Survival Analysis is ratio of discrete exponential (geometric) distribution that aims to model changepoints as a mortality rate. For brevity, it has a hyperparameter λ, which is typically set to 30 because its effect are negligible to the posterior beliefs after given a sufficiently large number of observations.

At every time t the objective is to recursively calculate the *posterior predictive* given below, which is available in closed form due to conjugate priors being assumed.

$$f(\boldsymbol{y}_t|\boldsymbol{y}_{1:(t-1)}, r_{t-1}) = \int_{\Theta} f(\boldsymbol{y}_t|\boldsymbol{\theta})\pi(\boldsymbol{\theta}|\boldsymbol{y}_{(t-r_{t-1}):(t-1)})d\boldsymbol{\theta} \tag{2.11}$$

Additionally, the joint distribution $\mathbb{P}(\boldsymbol{y}_{1:t}, r_t)$ for each datum must be computed in each cycle. The recursion can be formally derived by marginalising out discrete terms to obtain equation (2.13) by checking that

$$\begin{aligned} \mathbb{P}(\boldsymbol{y}_{1:t}, r_t) &= \sum_{r_{t-1}} \mathbb{P}(\boldsymbol{y}_{1:t}, r_t, r_{t-1}) \\ &= \sum_{r_{t-1}} \mathbb{P}(y_t, r_t|\boldsymbol{y}_{1:(t-1)}, r_{t-1})\mathbb{P}(\boldsymbol{y}_{1:(t-1)}, r_{t-1}) \\ &= \sum_{r_t-1} \mathbb{P}(y_t|\boldsymbol{y}_{1:(t-1)}, r_{t-1})\mathbb{P}(r_t|r_{t-1})\mathbb{P}(\boldsymbol{y}_{1:(t-1)}, r_{t-1}) \end{aligned}$$

Finally, the recursion for the joint distribution leads to a compact form which can be solved using dynamic programming by initialising (2.12) at $t = 1$ and storing the last term on the RHS in (2.13).

$$p(y_1, r_1) = \int_{\Theta} f(y_1|\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta} = f(y_1|y_0) \tag{2.12}$$

$$p(\boldsymbol{y}_{1:t}, r_t) = \sum_{r_t - 1} \Big\{ f(y_t|\boldsymbol{y}_{1:(t-1)}, r_{t-1}) p(r_t|r_{t-1}) \underbrace{p(\boldsymbol{y}_{1:(t-1)}, r_{t-1})}_{\text{recursive term}} \Big\} \tag{2.13}$$

To summarize the work of Adams and MacKay (2007) thus far, readers should takeaway two essential points from this section. Firstly, the standard Bayesian On-line CPD algorithm is computationally efficient by manner in which CP priors are constructed. Secondly, a CP signals that there is a shift in parameterization of the model distribution $\pi(\boldsymbol{\theta})$, due to a discrepancy between the current observations and the model trying to characterize it. In regards to the second point however, one of the main drawbacks is that this assumes a single model governs all possible segments. This can be inherently restrictive when trying to infer the best parameters for all possible observations.

2.4 Model selection

To circumvent the drawback addressed in the standard Bayesian On-line CPD framework, a recent paper by Knoblauch and Damoulas (2018) proposes the unification of two main ideas from Adams and MacKay (2007) and Fearnhead and Liu (2007).

Rather than assuming a single model, the paper generalizes this by introducing a *model universe* $\mathcal{M}$. At every time t, introduce a new r.v. $m_t \in \mathcal{M}$ to be a model describing the segment since the last CP at $\boldsymbol{y}_{(t-r_t):t}$. Now, the setup of prior beliefs previously from equations (2.7), (2.8) and (2.9) remains almost identical; the only modifications being that priors over $\boldsymbol{\theta}$ are now conditioned upon a subscript m_t because we no longer assume a single model, and priors over models are appended into the setup in equation (2.15).

$$r_t|r_{t-1} \sim H(r_t, r_{t-1}) \qquad \forall r_t \in \mathbb{N} \tag{2.14}$$

$$m_t|m_{t-1}, r_t \sim q(m_t|m_{t-1}, r_t) \qquad \forall m_t \in \mathcal{M} \tag{2.15}$$

$$\boldsymbol{\theta}_{m_t}|m_t \sim \pi_{m_t}(\boldsymbol{\theta}_{m_t}|m_t) \qquad \forall \boldsymbol{\theta}_{m_t} \in \boldsymbol{\Theta}_m \tag{2.16}$$

$$\boldsymbol{y}_t|\boldsymbol{\theta} \sim f(\boldsymbol{y}_t|\boldsymbol{\theta}) \qquad \forall \boldsymbol{y}_t \in \mathbb{R}^s \tag{2.17}$$

where the model prior $q : \mathcal{M} \to [0, 1]$ is given by

$$q(m_t|m_{t-1}, r_t) = \begin{cases} \mathbb{1}_{m_{t-1}}(m_t) & \text{if } r_t = r_{t-1} + 1 \\ q(m_t) & \text{if } r_t = 0 \end{cases} \tag{2.18}$$

The probabilistic recursions previously from the joint distribution in equations (2.12) and (2.13) also yield similar modifications. On the LHS of equations (2.19) and (2.20), the joint distribution now includes m_t; whilst on the RHS, we now have to consider the product of model distribution $q(m_t)$.

$$p(\boldsymbol{y}_1, r_1, m_1) = q(m_1) \int_{\Theta_{m_1}} f_{m_1}(y_1|\boldsymbol{\theta}_{m_1}) \pi_{m_1}(\boldsymbol{\theta}_{m_1}) d\boldsymbol{\theta}_{m_1} = \overbrace{q(m_1)}^{\text{model dist.}} \times f_{m_1}(y_1|y_0) \tag{2.19}$$

$$p(\boldsymbol{y}_{1:t}, r_t, m_t) = \sum_{m_{t-1}} \sum_{r_t - 1} \Big\{ f_{m_t}(y_t|\boldsymbol{y}_{1:(t-1)}, r_{t-1}) \overbrace{q(m_t|\boldsymbol{y}_{1:(t-1)}, r_t, m_{t-1})}^{\text{model dist.}} p(r_t|r_{t-1}) p(\boldsymbol{y}_{1:(t-1)}, r_{t-1}, m_{t-1}) \Big\} \tag{2.20}$$

It is worth mentioning that the recursion in Adams and MacKay (2007) is special case for when $|\mathcal{M}| = 1$. For $|\mathcal{M}| > 1$, the model distribution required for computing the joint distribution can be generalised as such:

$$q(m_t|m_{t-1}, r_t, \boldsymbol{y}_{1:(t-1)}) = \begin{cases} q(m_{t-1}|\boldsymbol{y}_{1:(t-1)}, r_{t-1}) & \text{if } r_t = r_{t-1} + 1 \\ q(m_t) & \text{if } r_t = 0 \end{cases} \tag{2.21}$$

where

$$q(m_{t-1}|\boldsymbol{y}_{1:(t-1)}, r_{t-1}) = \frac{p(m_{t-1}, r_{t-1}, \boldsymbol{y}_{1:(t-1)})}{\sum_{m_{t-1}} p(m_{t-1}, r_{t-1}, \boldsymbol{y}_{1:(t-1)})} \tag{2.22}$$

To interpret the statements in equations (2.21) and (2.22), we can think of this as the model posterior at time $t-1$ becoming the prior at time t given run-lengths and observations. Finally, we have to define the *growth* and *changepoint* probabilities to assess situations when a run-length grows by one or truncates to zero whilst taking the model distribution into account:

$$\begin{aligned} p(\boldsymbol{y}_{1:t}, r_t = r_{t-1} + 1, m_t) =& f_{m_t}(\boldsymbol{y}_t)|y_{1:(t-1)}, r_t) p(\boldsymbol{y}_{1:(t-1)}, r_{t-1}, m_{t-1}) \\ & \times (1 - H(r_t)) q(m_{t-1}|y_{1:(t-1)}, r_t) \end{aligned} \tag{2.23}$$

$$\begin{aligned} p(\boldsymbol{y}_{1:t}, r_t = 0, m_t) =& f_{m_t}(\boldsymbol{y}_t|\boldsymbol{y}_{1:(t-1)}, r_t) q(m_t) \\ & \times \sum_{m_{t-1}} \sum_{r_{t-1}} \Big\{ H(r_{t-1} + 1) p(\boldsymbol{y}_{1:(t-1)}, r_{t-1}, m_{t-1}) \Big\} \end{aligned} \tag{2.24}$$

The joint distribution in equation (2.20) is stored for every time t and is used to compute several other key quantities. For example, the evidence in equation (2.25) can be straightforwardly obtained by marginalising out all possible variants of m_t and r_t. With that, inference over run-lengths and models can be achieved by calculating the run-length and model posterior given in equation (2.26), which in turn can be re-used to compute the quantities in equations (2.27) to (2.29). The purpose of computing and storing these quantities is that they play a

crucial role in implementing TS segmentation and other outputs outlined in the next section.

$$p(\boldsymbol{y}_{1:t}) = \sum_{m_t} \sum_{r_t} p(\boldsymbol{y}_{1:t}, m_t, r_t) \qquad \text{[evidence]} \tag{2.25}$$

$$p(r_t, m_t|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_{1:t}, r_t, m_t)}{p(\boldsymbol{y}_{1:t})} \qquad \text{[run-length and model posterior]} \tag{2.26}$$

$$p(m_t|\boldsymbol{y}_{1:t}) = \sum_{r_t} p(r_t, m_t|\boldsymbol{y}_{1:t}) \qquad \text{[model marginal posterior]} \tag{2.27}$$

$$p(r_t|m_t, \boldsymbol{y}_{1:t}) = \frac{p(r_t, m_t|\boldsymbol{y}_{1:t})}{p(m_t|\boldsymbol{y}_{1:t})} \qquad \text{[model specific run-length dist]} \tag{2.28}$$

$$p(r_t|\boldsymbol{y}_{1:t}) = \sum_{m_t} p(r_t, m_t|\boldsymbol{y}_{1:t}) \qquad \text{[run-length marginal posterior]} \tag{2.29}$$

$$q(m_t|\boldsymbol{y}_{1:(t-1)}, r_{t-1}) = \frac{p(m_{t-1}, r_{t-1}|\boldsymbol{y}_{1:(t-1)})}{p(r_{t-1}|\boldsymbol{y}_{1:(t-1)})} \qquad \text{[conditional model posterior]} \tag{2.30}$$

2.5 Algorithm output

We now move on to specifics on how exactly CPs are constructed, and what other intriguing capabilities does the algorithm offer.

Along with CPD, it is worth mentioning that the original Bayesian On-line CPD framework by Adams and MacKay (2007) was able to perform one-step-ahead predictions only, whilst Fearnhead and Liu (2007) does not. Both papers focused on univariate streams and tracking the most recent CP posterior distribution. However, both had proposed different methods of TS segmentation, where the latter motivates a Maximum A Posteriori method for infering CPs and models.

In particular, we follow Knoblauch and Damoulas (2018)'s version of CPD which unifies the strengths of both approaches, and focused on multivariate streams in the original paper. This entails an algorithm that can deliver three different tasks: TS segmentation, h-step predictions and model selection. For the remainder of this thesis, we call this algorithm: *Bayesian On-line Changepoint Detection with Model Selection* (BOCPDMS).

2.5.1 Segmentation

Inspired by Fearnhead and Liu (2007), the intuition behind detecting CPs is that given observations $\boldsymbol{y}_{1:t}$ so far, there exists an optimal solution within a vast permutation of different possible run-lengths r_t and models m_t, given the most recent CP. Suppose an optimal solution *does exists* at some time step t. Then at the next time step $t+1$ with the next observation y_{t+1}, the next optimal solution might differ from the current state, which necessitates the need for some mechanism to reconsider optimal solutions dynamically. This is akin to dealing with

common issues in time-series behaviours such as handling anomalies or high volatility.

The core engine that conceptualizes the train of thought above is described by the recursive Maximum A Posteriori (MAP) segmentation equation. Let $MAP_0 = 1$, then $\forall t \geq 1$:

$$MAP_t = \arg\max_{r,m} \left\{ p(\boldsymbol{y}_{1:t}, r_t = r, m_t = m) MAP_{t-r-1} \right\} \tag{2.31}$$

where the optimal solution at time t is (r_t^*, m_t^*). Let S_t be the segmentation at time t, then update this using $S_t = S_{t-r_t^*-1} \cup \{(t - r_t^*, m_t^*)\}$. At the initialisation stage, denote $(0, m_0)$ to be the optimal solution and $S_0 = \emptyset$. Henceforth a solution (t^*, m_{t^*}) in $S_t \implies$ a CP has occurred at time t^* with model m_{t^*} for $\boldsymbol{y}_{t^*:t}$. Note that the CPs are not always discovered at the most recent point of time, and they can be declared in retrospection.

The joint distribution $p(\boldsymbol{y}_{1:t}, r_t = r, m_t = m)$ can also be substituted by the run-length model posterior in equation (2.26) due to proportional terms. Additionally, it is essential to realise that the term MAP_{t-r-1} on the RHS of equation (2.31) conceives the MAP segmentation the ability to reconsider run-lengths and models. Consider Figure (2.4) (for $|\mathcal{M}| = 1$), where a time-series depicts two possible outcomes: either a CP or an anomaly should be detected. After seeing a sufficient number of observations after time $t \geq 10$, the optimal solution (r_t^*, m_t^*) will eventually re-consolidate itself.

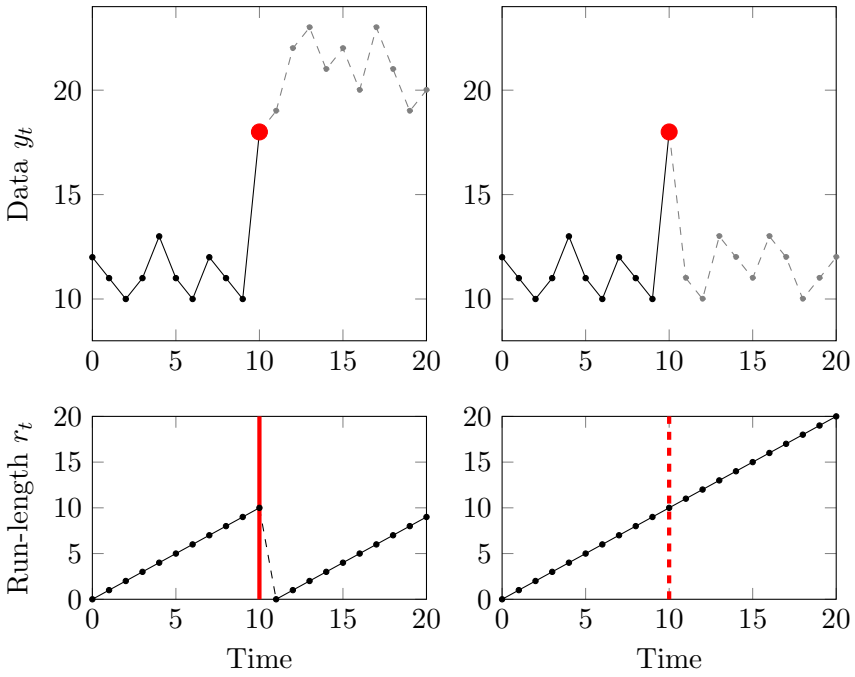


Figure 2.4: Illustration of how the recursive MAP segmentation dynamically reconsiders optimal solutions: a CP is detected (left) or an anomaly is handled (right). The red dashed line indicates that a CP was *previously* declared at that point.

2.5.2 Prediction

Apart from the recursive MAP segmentation, BOCPDMS is also capable of performing an arbitrary h-step forward prediction. Define $\boldsymbol{Y}_{t+h}$ as the r.v. that represents this prediction and $\hat{\boldsymbol{y}}_t^h$ the predictive interval between h and t. Then the recursive h-step-ahead forecast is formally defined as:

$$p(\boldsymbol{Y}_{t+h}|\boldsymbol{y}_{1:t}) = \sum_{r_t}\sum_{m_t}\left\{p(\boldsymbol{Y}_{t+h}|\boldsymbol{y}_{1:t}, \hat{\boldsymbol{y}}_t^h, r_t, m_t)p(r_t, m_t|\boldsymbol{y}_{1:t})\right\} \tag{2.32}$$

where

$$\hat{\boldsymbol{y}}_t^h = \begin{cases} \emptyset & \text{if } h = 1 \\ \hat{\boldsymbol{y}}_{(t+1):(t+h-1)} & \text{otherwise} \end{cases} \tag{2.33}$$

and $\hat{\boldsymbol{y}}_{t+h} = \mathbb{E}(\boldsymbol{Y}_{t+h}|\boldsymbol{y}_{1:t}, \hat{\boldsymbol{y}}_t^h)$ is the recursive forecast or the posterior expection of the h-step ahead prediction. The LHS of expression (2.32) is trivial in standard TSA notation for predictions; the only exception being that the RHS must consider all variants of r_t and m_t marginalised over the run-length and model posterior. As $h \to \infty$, it is intuitively easy to see that predictions become less accurate over time, because temporal dependencies between two observations are more likely to change when they are further apart. Traditionally, a one-step ahead forecast or the posterior expectation provides the most accurate forecast.

2.5.3 Model inference

One of the novel capabilities of BOCPDMS is its ability to perform model inference by tracking the model posterior $p(m_t|\boldsymbol{y}_{1:t})$ in equation (2.27). Recall that in the previous sections, discussions were made about how selecting model priors can prove difficult, and it is unrealistic to assume a single model. Tracking the model posterior solves this by two-folds. First, they provide information regarding which model captures the data better. This is attractive when structural changes in data happen slowly and are not captured well by CPs (Knoblauch and Damoulas, 2018). Second, *when* is the model useful? Many time-series models are affected by *seasonality* or variations that occur at specific regular intervals. For example, stakeholders need to know whether a certain model is more effective at a particular given month of a year. Hence, if a model is known to be good at a specific time, ideally, this can be re-used in the next cycle.

2.6 Computational costs

In the final section of Chapter 2, we end by analyzing the time and space complexity of the BOCPDMS algorithm in Algorithm (1) - by studying a single instance when a new observation or datum is fed to the algorithm from line 4 onwards.

During the 'for loop' starting from line 5, the algorithm either initialises or updates the growth and CP probabilities for all possible models, resulting in $\mathcal{O}(|\mathcal{M}|t)$ over time t assuming that the joint distribution can be calculated in $\mathcal{O}(1)$ for all models in $\mathcal{M}$. Next, the joint distribution is computed in line 11 and is stored for the next datum is used recursively, adhering to the dynamic programming paradigm. This results in a time and space complexity of $\mathcal{O}(|\mathcal{M}|t)$ for all possible models in $\mathcal{M}$, too. In line 12, segmentations and predictions requires maximising over the run-lengths r_t and model m_t. For a large t, this takes precisely $\mathcal{O}(\max(|\mathcal{M}|, t)) = \mathcal{O}(t)$. Finally the average time and space complexity overall for the algorithm is $\mathcal{O}(|\mathcal{M}|t)$. In particular, it is important to note that the main computational bottleneck of the algorithm lies in lines 4-10 because all models must be computed. One way to partially resolve this is by computing quantities for each $|\mathcal{M}|$ in parallel using multi-threads. With this, the time complexity can be reduced to $\mathcal{O}(|\mathcal{M}|/N) \cdot \max_{M \in \mathcal{M}} \text{CompTime}(M)$ (Knoblauch and Damoulas, 2018).

Naively, if all run-lengths $R(t) = \{1, \ldots, t\}$ are retained, then the complexity of BOCPDMS is linear in time. A simple method to reduce magnitude of the time complexity is via pruning $R(t)$, such that we keep the most likely run-lengths based on a set criterion and discard the rest. Adams and MacKay (2007)'s approach was to discard run-lengths whose posterior probability is $\leq 1/R_{\max}$ or only keep the $R_{\max}$ most probable ones (Adams and MacKay, 2007) - which in turn guarantees an upper bound of $\mathcal{O}(R_{\max})$. Overall, this partially reduces the time complexity for lines 11 to $\mathcal{O}(|\mathcal{M}|R_{\max})$ (Knoblauch and Damoulas, 2018) because pruning allows for $\mathcal{O}(1)$ for the retained run-lengths only.

Algorithm 1 Bayesian On-line Changepoint Detection with Model Selection

1: **Input at time 0:** model universe $\mathcal{M}$, Hazard H, prior q
2: **Input at time t:** next observation $\boldsymbol{y}_t$
3: **Output at time t:** segmentation S_t, prediction $\hat{\boldsymbol{y}}_{(t+1):(t+h_{\max})}$, model dist. $p(m_t|\boldsymbol{y}_{1:t})$
4: **for** next observation $\boldsymbol{y}_t$ at time t **do**
5: **for** $m \in \mathcal{M}$ **do**
6: **if** $t = 1$ **then**
7: Initialize $p(\boldsymbol{y}_{1:t}, r_t = 0, m_t = m)$ with prior via (2.19)
8: **else if** $t > 1$ **then**
9: Update $p(\boldsymbol{y}_{1:t}, r_t, m_t = m)$ via (2.20)
10: Prune model-specific run-length distribution
11: Obtain joint distribution over $\mathcal{M}$ via (2.25) - (2.30)
12: Compute segmentation and prediction via (2.31) and (2.32)
13: Update model parameters $\boldsymbol{\theta}_{m_t}$ for all $m_t \in \mathcal{M}$
14: **Output:** S_t, $\hat{\boldsymbol{y}}_{(t+1):(t+h_{\max})}$, $p(m_t|\boldsymbol{y}_{1:t})$

Chapter 3

Unifying Point Processes and Gaussian Processes

Objectives:

- ✓ Outlining the key difference between a Poisson Process and Cox Process.
- ✓ Introducing the Log Gaussian Cox Process model.
- ✓ Reviewing Gaussian Process Regression. Discussing issues with intractability. Overcoming it using the Laplace approximation and Variational Inference.
- ✓ Reducing the computational burden of the model using sparse approximations.

3.1 Preliminaries

In the previous chapter, the scope of statistical data in CPD has been limited to real-valued, continuous values thus far. In the scope of this thesis, we now turn to *count data* where observations can only take values non-negative integer values $\{0, 1, 2, 3, \dots\}$. In particular, we are interested in learning about a mathematical tool known as *Point Processes* which serve as a building block. Point processes can generally be thought of as random patterns of points in an arbitrary d-dimensional space (Baddeley, 2006). Many researchers who work with complex stochastic systems, especially in domains including environmental statistics, ecology and material science benefit from studying properties of Point Processes (Møller and Waagepetersen, 2007).

Point Processes are vast in their academic literature and often easy to get lost in. Rather than attempting to survey all aspects of what it entails; the focus of this section is to cover the

most common type of Point Process applicable in everyday lives, known as a *Poisson Process* (Kingman, 1992).

Poisson Processes, by inheritance of Point Processes and its properties, have a distinct notation in a one dimension. Unlike a higher order d-dimensional Point Processes, it has a natural ordering of time which is absent when $d \geq 2$ (Baddeley, 2006). This subtle difference is small enough to be ignored; yet its importance is absolutely crucial in the context of time-series data.

In order to visually understand Poisson Processes in single dimension, consider the following definition. Let $T_1 < T_2 < \ldots$ where T_i is the arrival time at the i-th point, then a one-dimensional Point Process can be modelled as in Figure (3.1).

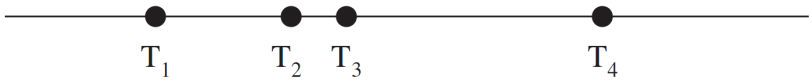


Figure 3.1: Arrival times T_i in a one-dimensional Point Process (Baddeley, 2006).

Alternatively, Point Processes can also be modelled as inter-arrival times $S_i = T_{i+1} - T_i$ in Figure (3.2). In the case of Poisson Processes, these random variables $S_1, S_2, \ldots$ are independent.

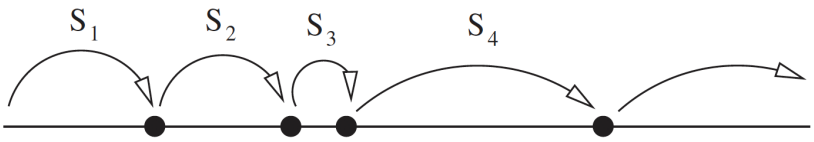


Figure 3.2: Inter-arrival times S_i in a one-dimensional Point Process (Baddeley, 2006).

Lastly, Poisson Processes can also be modelled as interval counts $N(a, b] = N_b - N_a$ for some $0 \leq a \leq b$ which counts the number of points arriving in the interval $(a, b]$ in Figure (3.3). In the case of Poisson Processes, these interval counts are disjoint intervals that are independent.

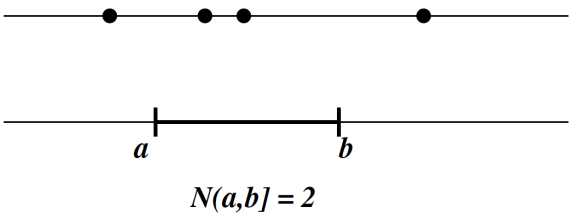


Figure 3.3: Interval count $N(a, b]$ for a Point Process (Baddeley, 2006).

Summarised together, the above definitions can be used to formally define a one-dimensional

Poisson process. Suppose we claim that there are three primary assumptions that are valid (without proof): First, the number of points which arrive in an interval has the expected value $\mathbb{E}N(a,b] = \beta(b-a)$, where β is the rate or intensity of the process. Second, assume that arrivals in disjoint intervals are independent, that $a_1 < b_1 < a_2 < b_2 < \cdots < a_m < b_m \implies N(a_1,b_1], \ldots, N(a_m,b_m]$. Finally, assume that the probability of two or more arrivals in a given time is asymptotically, uniformly smaller order than the length of the interval. According to Baddeley (2006), the three assumptions imply that the number of points arriving in a given time interval must follow a Poisson distribution:

$$N(a,b] \sim \text{Poisson}(\beta(b-a)) \tag{3.1}$$

where Poisson(μ) denotes the Poisson distribution with mean μ defined by

$$\mathbb{P}(N=k) = \exp^{-\mu} \frac{\mu^k}{k!}, \quad k = 0,1,2,\ldots \tag{3.2}$$

Intuitively, this result can be obtained by splitting $(a,b]$ into small n-number of intervals and understanding that the number of arrivals in each small interval is equal to 0 or 1. Since $N(a,b]$ is the sum of these numbers, it has an approximately Binomial distribution. Letting $n \to \infty$, then it must be true that $N(a,b]$ is a Poisson distribution.

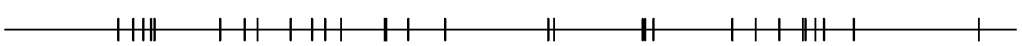


Figure 3.4: Realisations of a one dimension Poisson process with uniform intensity 1 in the time ordering interval [0, 30]. Tick marks indicate arrival times. (Baddeley, 2006).

For a general definition for higher order dimensions, Møller and Waagepetersen (2007) states that a Poisson process $\boldsymbol{X}$ defined on space S with intensity measure μ and intensity function ρ satisfies any bounded region $B \subseteq S$ with $\mu(B) > 0$ and the two conditions hold:

i $N(B)$ is Poisson distributed with mean $\mu(B)$

ii conditional on $N(B)$, points in $\boldsymbol{X}_B$ are *i.i.d.* with density proportional to $\rho(u)$, $u \in B$

It is important to note that if $\rho(u)$ is constant $\forall u \in S$, then $\boldsymbol{X}$ is described as *homogeneous* or a stationary Poisson process; it is described by complete spatial randomness or a lack of interaction. Further, stationarity means $\rho(u)$ is constant, which implies isotropy of $\boldsymbol{X}$ meaning that its distribution has rotational and translational invariance in the space that $\boldsymbol{X}$ is defined (Baddeley, 2006). In most cases, a homogeneous Poisson process is too simplistic for modelling real-world scenarios because this intensity rate is rarely 'constant'. For example in ecological

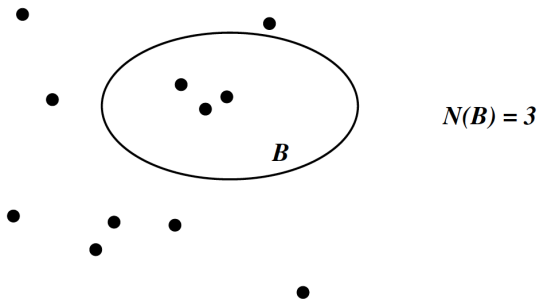
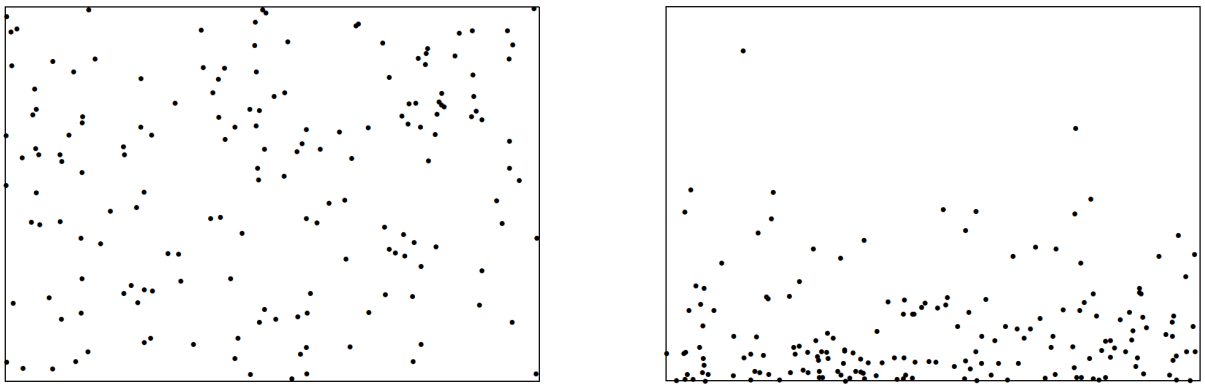


Figure 3.5: Counting variables $N(B)$ for a spatial point process with $d = 2$ (Baddeley, 2006)



(a) Homogeneous **(b)** Inhomogeneous

Figure 3.6: Realisations of a 2-dimensional spatial Poisson Process, with constant (left) and stochastic intensity rate (right)

data, a scientist might deal with clustered point patterns caused by environmental random heterogeneity. (Møller and Waagepetersen, 2007).

As opposed to a constant intensity function $\rho(u)$, a *Cox process* is a natural progression of a Poisson process X driven by a non-negative stochastic process $\mathbf{\Lambda} = (\Lambda(u))_{u \in S}$ (Møller and Waagepetersen, 2007), described as *inhomogeneous*. Because the random intensity rate of a Cox process is itself stochastic process, a Cox process also bears the name *doubly stochastic Poisson process*. The way in which $\mathbf{\Lambda}$ is constructed determines how realisations of these processes look like, which leads a question of *how* to set $\mathbf{\Lambda}$? Many Cox processes have interesting properties that have well studied effects, for example, simulating clustered points using Neyman-Scott processes or Shot Noise Cox processes (Møller, 2003).

3.2 Log Gaussian Cox Process model

The purpose of introducing Point Processes is that they set a foundation for the centrepiece of this thesis, which is to propose a specific type of model class. This model class is known as the *Log Gaussian Cox Process*, and is popular for spatial and spatio-temporal data due to its various appealing properties.

In particular, this model falls under the umbrella of Bayesian statistics known as *non-parametric* methods. The best way to understand a non-parametric model is by recalling that a *parametric* model assumes a finite set of parameters (Ghahramani, 2013). Given parameters θ, future predictions x and observed data $\mathcal{D}$, we can express the distribution of predictions given data as such:

$$p(x|\theta, \mathcal{D}) = p(x|\theta)$$

which implies parameters captures everything there is to know about the data for predictions. According to Ghahramani (2013), we can think of parametric models as *information channels* from past data $\mathcal{D}$ to future predictions x.

Contrastingly, a non-parametric model assumes that $\mathcal{D}$ cannot be defined in terms of parameters. Often, they can be defined by assuming an infinite dimensional-θ instead which conveys a misnomer by its name 'non-parametric', and they cannot be explicitly represented in terms of parameters (Ghahramani, 2013). Predictions from non-parametric models are *memory-based*; they need to store or remember a growing amount of information about $\mathcal{D}$. They are inextricably tied to the notion of *exchangeability*. A sequence is said to be exchangeable if its joint distribution is invariant under arbitrary permutation of the indices (Ghahramani, 2013). Overall, the main appeal of non-parametric models is that they provide flexibility. This is true because the complexity of the model and capacity of information channel is no longer unbounded, which means that they can capture unseemly trends not possible with parametric models.

According to Møller et al. (1998), a Log Gaussian Cox Process (LGCP) is defined by three components:

$$\begin{aligned} f &\sim \mathcal{GP}(m, K) \\ \Lambda &= \exp(f) \\ y|\Lambda &\sim \text{Poisson}(\Lambda) \end{aligned} \tag{3.3}$$

where f is a *Gaussian Process* (GP), which is a collection of random variables, any finite of which have a joint Gaussian distribution (Rasmussen and Williams, 2006). Observe that the stochastic intensity Λ takes the exponential of f because a Poisson process must have positive intensity. The next section covers Gaussian Processes in detail, for the reason they are important for determining how the stochastic intensity Λ is constructed.

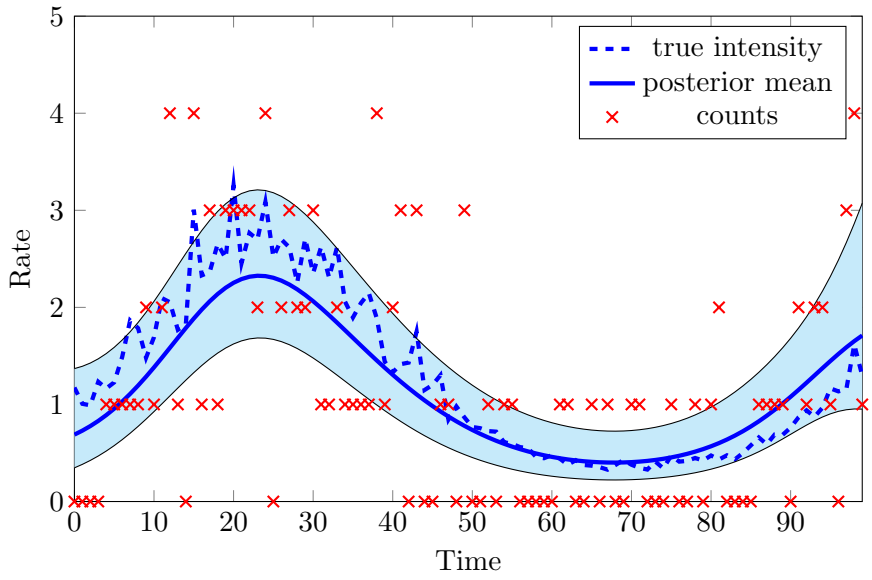


Figure 3.7: An illustration of an exponential Gaussian Process with a 95% credible interval on synthetic Poisson generated count data.

3.3 Gaussian Process Regression

The unique footprint of a GP is that unlike classical models, they can be visualized in a *function-space view* as *latent functions* f. GPs can also be thought of as projections of inputs into some feature space. In particular, this forms under the canonical basis that inner products in an input space can be lifted into some feature space, by replacing occurrences of those inner products with a kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$; known as the *kernel trick* (Rasmussen and Williams, 2006).

GPs are completely specified by two important components: the mean function $m(\boldsymbol{x})$ and a *kernel* function $k(\boldsymbol{x}, \boldsymbol{x}')$, where $k(\boldsymbol{x}, \boldsymbol{x}')$ is used to compute the covariance matrix of the joint Gaussian distribution.

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \tag{3.4}$$

In typical cases, the mean function is set to $\mathbf{0}$ unless we want to assign a prior belief, for example, if we know a linear trend exists in some generative model and we want to include this. Bayesian inference can be done using GPs by sampling from this distribution, or more specifically, assumptions about the latent functions are encoded in a GP prior followed by drawing functions from the posterior distribution. One necessary condition of a kernel function is that they must be *positive semi-definite* (PSD). In general, a PSD matrix is any $n \times n$ matrix K such that $v^\intercal K v \geq 0$ for all $v \in \mathbb{R}^n$. The *Gram matrix* is the matrix whose entries are computed by the kernel function for two inputs. A kernel is PSD if and only if any choice of input gives rise to a PSD Gram matrix (Rasmussen and Williams, 2006).

Let $\boldsymbol{y}$ be the observed data and $\boldsymbol{f} \in \mathbb{R}^{n \times 1}$ be the random functions values at locations

$\boldsymbol{X} \in \mathbb{R}^{n \times q}$ training inputs, with n number of input locations and q dimensions. Then let $\boldsymbol{f}_*$ correspond to the test or predictive outputs located at $\boldsymbol{X}_*$ test inputs. For the remainder of this chapter, denote $k(\boldsymbol{X}, \boldsymbol{X}) \triangleq \boldsymbol{K}_{\boldsymbol{ff}}$ which is the kernel function evaluated at the training inputs only for brevity. Similarly, let $k(\boldsymbol{X}_*, \boldsymbol{X}) = \boldsymbol{K}_{\boldsymbol{f}_*\boldsymbol{f}}$, $k(\boldsymbol{X}, \boldsymbol{X}_*) = \boldsymbol{K}_{\boldsymbol{f}\boldsymbol{f}_*}$ and $k(\boldsymbol{X}_*, \boldsymbol{X}_*) = \boldsymbol{K}_{\boldsymbol{f}_*\boldsymbol{f}_*}$ be the kernel function evaluated at test and training inputs.

With the specified notation above, the joint distribution over observations and predictive outputs including noise σ^2 of size $n \times n$ can be expressed below, because they are both multivariate normal.

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}_* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{ff}} + \sigma^2 I & \boldsymbol{K}_{\boldsymbol{f}\boldsymbol{f}_*} \\ \boldsymbol{K}_{\boldsymbol{f}_*\boldsymbol{f}} & \boldsymbol{K}_{\boldsymbol{f}_*\boldsymbol{f}_*} \end{bmatrix}\right) \tag{3.5}$$

The predictive equations for a GP is available because of the fact that $k(\boldsymbol{x}, \boldsymbol{x}')$ is PSD $\implies$ $\boldsymbol{K}_{\boldsymbol{ff}}$ must be invertible, in equations (3.7) and (3.8).

$$\boldsymbol{f}_*|X, \boldsymbol{y}, X_* \sim \mathcal{N}(\mu(\boldsymbol{f}_*), \mathrm{Cov}(\boldsymbol{f}_*)) \tag{3.6}$$

$$\mu(\boldsymbol{f}_*) = \mathbb{E}[\boldsymbol{f}_*|X, \boldsymbol{y}, X_*] = \boldsymbol{K}_{\boldsymbol{f}_*\boldsymbol{f}}(\boldsymbol{K}_{\boldsymbol{ff}} + \sigma^2 I)^{-1}\boldsymbol{y} \tag{3.7}$$

$$\mathrm{Cov}(\boldsymbol{f}_*) = \boldsymbol{K}_{\boldsymbol{f}_*\boldsymbol{f}_*} - \boldsymbol{K}_{\boldsymbol{f}_*\boldsymbol{f}}(\boldsymbol{K}_{\boldsymbol{ff}} + \sigma^2 I)^{-1}\boldsymbol{K}_{\boldsymbol{f}\boldsymbol{f}_*} \tag{3.8}$$

It is important to understand that a kernel function computes the 'similarity' measure between two inputs in some feature space because it computes the covariance between latent functions. Standard kernels typically contains two hyper-parameters: the *lengthscale* ℓ which determines the 'smoothness' of functions drawn; and the *signal-variance* σ^2_{signal} determines the average distance away from the mean or the 'amplitude'. For example, a commonly used kernel function is the *Radial Basis Function* (RBF) kernel in equation (3.9). A major concept within the kernel method literature is the order of *differentiability*. The RBF kernel is known to produce infinitely differentiable functions (Rasmussen and Williams, 2006). This means that if we zoom into these functions, its smoothness will always be maintained (Saul, 2016).

$$k_{RBF}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2_{signal} \exp\left(-\frac{1}{2\ell^2}(\boldsymbol{x} - \boldsymbol{x}')^2\right) \tag{3.9}$$

For completeness, some other popular kernel functions used in GPs include the Periodic (3.10) and Matern class kernels of choice $\frac{3}{2}$ (3.11) and $\frac{5}{2}$ (3.12) (which refers to their order of

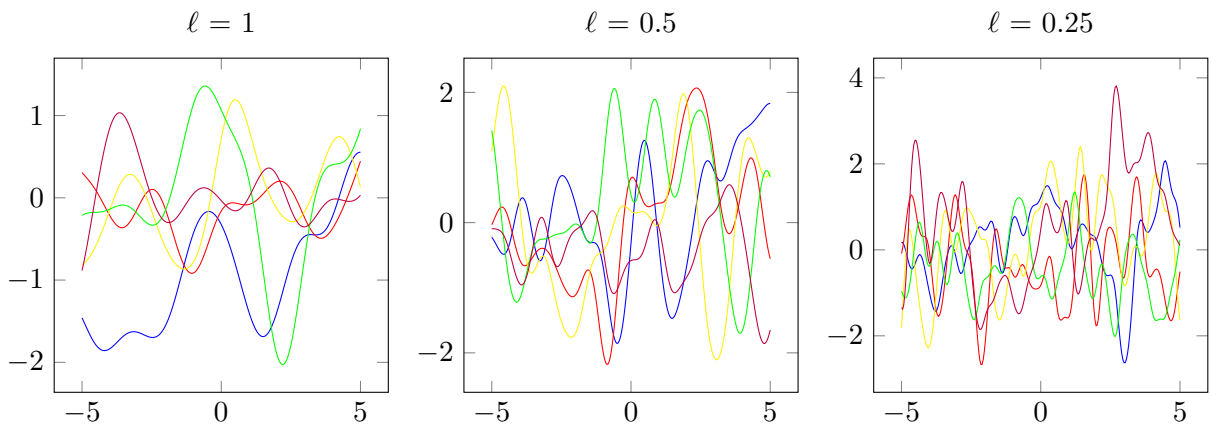


Figure 3.8: Illustration of different lengthscale values on RBF kernel samples, fixing the signal-variance to 1.

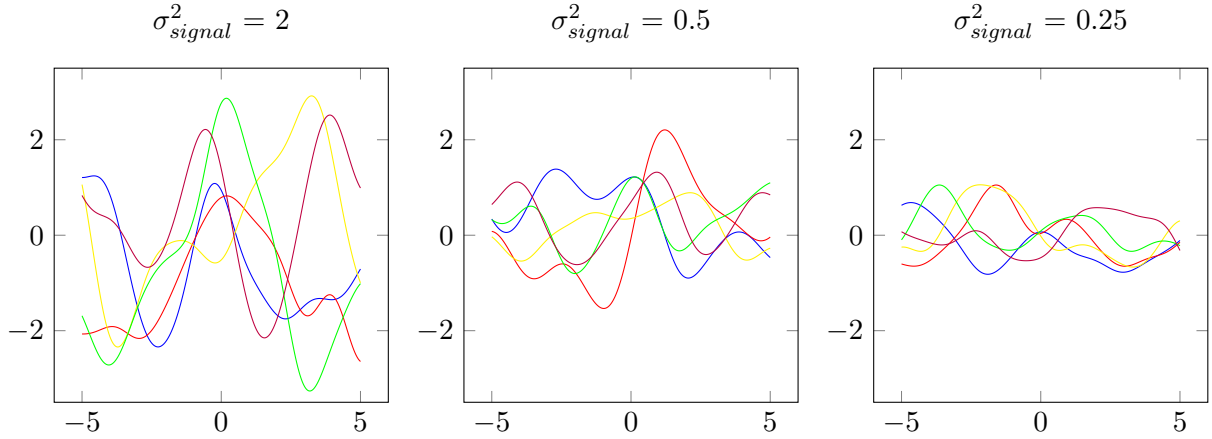


Figure 3.9: Illustration of different signal-variance values on RBF kernel samples, fixing the lengthscale to 1.

differentiability), shown in Figure (3.10) (MacKay, 1998).

$$k_{PER}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2_{signal} \exp\left(- \frac{2\sin^2((\boldsymbol{x} - \boldsymbol{x}')/2)}{\ell^2} \right) \tag{3.10}$$

$$k_{MA32}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2_{signal}(1 + \sqrt{3}r)\exp(-\sqrt{3}r) \tag{3.11}$$

$$k_{MA52}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2_{signal}(1 + \sqrt{5}r + \frac{5}{3}r^2)\exp(-\sqrt{5}r) \tag{3.12}$$

$$\text{s.t. } r = \left(\sum_{i=1}^{q} \frac{(x_i - x'_i)^2}{\ell_i^2} \right)^{1/2}$$

Finally, a kernel function can be said to be either *stationary* or *non-stationary*. A stationary kernel function is a function of $\boldsymbol{x} - \boldsymbol{x}'$, thus it is invariant to translations in the input space (Rasmussen and Williams, 2006), and otherwise it is non-stationary.

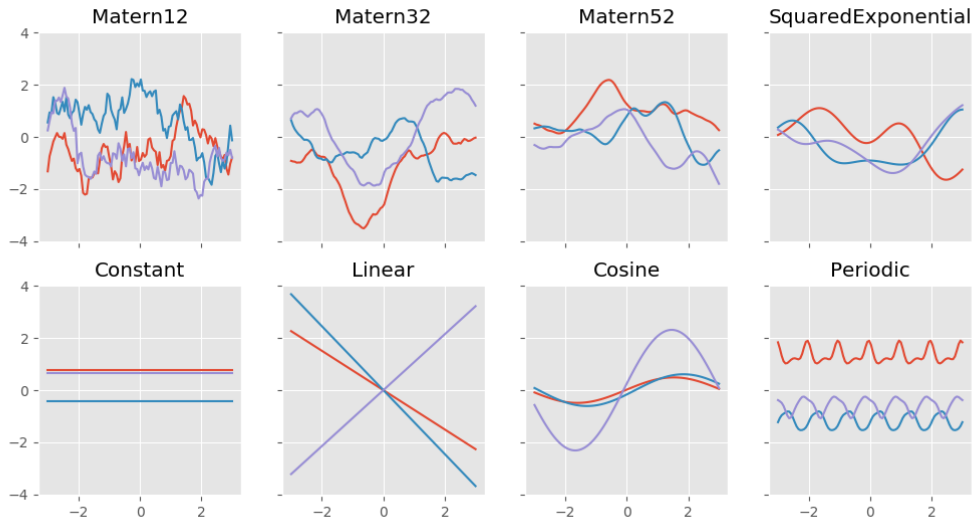


Figure 3.10: Drawing samples of GP latent functions using different kernel choices.

Although GPs are powerful, non-parametric models that take the form of latent functions; their primary drawback is that they are intrinsically slow and expensive. Naively, GPs have a time complexity of $\mathcal{O}(N^3)$ and requires a storage space of $\mathcal{O}(N^2)$. This is because in order to compute the predictive equations in (3.7) and (3.8), the covariance matrix must be inverted. Though, a practical implementation of a GP regression can be done slightly faster with the same order of magnitude, and numerically more stable (Rasmussen and Williams, 2006) using several tricks involved. Consider the implementation of a GP prediction in Algorithm (2). In line 3, the computational complexity for the operation known as the *Cholesky* decomposition is $\mathcal{O}(N^3/6)$. In lines 4 and 6 the complexity is $\mathcal{O}(N^2/2)$ to solve triangular systems. Hence the resulting complexity is $\mathcal{O}(N^3/6)$.

Algorithm 2 Implementation of GP predictions with noise

1: **Input:** data $\boldsymbol{X}$, targets $\boldsymbol{y}$, kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$, noise σ^2, test inputs $\boldsymbol{X}_*$
2: **Output:** mean $\mu(\boldsymbol{f}_*)$, covariance $\text{Cov}(\boldsymbol{f}_*)$
3: $L \coloneqq \text{Cholesky}(\boldsymbol{K}_{\boldsymbol{f}\boldsymbol{f}} + \sigma^2 I)$
4: $\boldsymbol{\alpha} \coloneqq L^\intercal \backslash (L \backslash \boldsymbol{y})$
5: $\mu(\boldsymbol{f}_*) \coloneqq \boldsymbol{K}_{\boldsymbol{f}_*\boldsymbol{f}}\boldsymbol{\alpha}$ ▷ Predictive mean eq. (3.7)
6: $\boldsymbol{v} \coloneqq L \backslash \boldsymbol{K}_{\boldsymbol{f}\boldsymbol{f}_*}$
7: $\text{Cov}(\boldsymbol{f}_*) \coloneqq \boldsymbol{K}_{\boldsymbol{f}_*\boldsymbol{f}_*} - \boldsymbol{v}^\intercal \boldsymbol{v}$ ▷ Predictive covariance eq. (3.8)
8: **Return:** $\mu(\boldsymbol{f}_*), \text{Cov}(\boldsymbol{f}_*)$

3.4 Approximations

If the likelihood function $p(\boldsymbol{y}|\boldsymbol{f})$ of a GP is assumed to be Gaussian distributed, then its integration is analytically tractable and is available because it is conjugate to itself. However, in the case of the LGCP model in equation (3.3) where the likelihood is Poisson distributed, the posterior $p(\boldsymbol{f}|\boldsymbol{y})$ is no longer analytically tractable and must be approximated instead. This is also true for any GP with non-Gaussian likelihoods (Rasmussen and Williams, 2006). Within the Bayesian community, there exists many ways to approximate the posterior $p(\boldsymbol{f}|\boldsymbol{y})$; all of which are motivated by very different approaches, and have multiple versions of each other after many iterations over the last decade (Hensman et al., 2013). Currently, some of the most popular methods include the Laplace approximation (Williams and Barber, 1998), Variational Inference (Gibbs and MacKay, 1997; Opper and Archambeau, 2008), Expectation Propagation (EP) (Minka, 2001), Markov Chain Monte Carlo (MCMC) (Neal, 1997) and more.

In this thesis, we focus on reviewing two methods only for approximation taken from Saul (2016): the Laplace approximation and Variational Inference. Generally. both methods tend to not incur large computational overheads and have good performances for certain situations. However, this is not to say that EP and MCMC are undesirable methods. It is known that MCMC can perform asymptotically exact approximations but tends to be very expensive; while EP can have matching performances with the Laplace approximation and performs well in certain tasks such as classification, but lacks guarantees of convergence.

3.4.1 Laplace Approximation

The Laplace approximation offers a simplistic method of approximating $p(\boldsymbol{f}|\boldsymbol{y})$ as a Gaussian distribution using a two-fold approach. First by attempting to find the mode of the true distribution and second by expanding around this point using a Taylor expansion.

We start by re-writing the true posterior in a different form (Bishop, 2006):

$$p(\boldsymbol{f}|\boldsymbol{y}) = \frac{1}{Z}h(\boldsymbol{f}) \tag{3.13}$$

where $Z = p(\boldsymbol{y})$ is a normalising constant in $\boldsymbol{f}$ and Z is usually difficult to compute. Note that $h(\boldsymbol{f})$ is a function of $\boldsymbol{f}$ unrelated to $\boldsymbol{y}$. The next step is to take the logarithm of the LHS and RHS of equation (3.13), followed by performing a second-order Taylor expansion around the point of $\boldsymbol{a}$ shown in equation (3.15) because we want to model a Gaussian approximation.

$$\log p(\boldsymbol{f}|\boldsymbol{y}) = \log \frac{1}{Z} + \log h(\boldsymbol{f}) \tag{3.14}$$

$$\approx \log \frac{1}{Z} + \log h(\boldsymbol{a}) + \frac{d \log h(\boldsymbol{a})}{d\boldsymbol{a}}(\boldsymbol{f}-\boldsymbol{a}) + \frac{1}{2}(\boldsymbol{f}-\boldsymbol{a})^{\intercal}\frac{d^2 \log h(\boldsymbol{a})}{d\boldsymbol{a}^2}(\boldsymbol{f}-\boldsymbol{a}) \tag{3.15}$$

The Laplace approximation performs equation (3.15) around the mode denoted by $\hat{\boldsymbol{f}}$. By definition the mode is a maximum point of $p(\boldsymbol{f}|\boldsymbol{y})$, hence the first order derivative evaluated at this point automatically becomes zero in equation (3.16). The mode is found using *Newton's* method in Rasmussen and Williams (2006), which is an iterative method for finding the roots of any differentiable function. One important aspect of the Laplace approximation is that the optimal mode is *not* always guaranteed to be found.

$$\frac{d \log h(\hat{\boldsymbol{f}})}{d\hat{\boldsymbol{f}}} = 0 \tag{3.16}$$

From the approximated equation (3.15), we then substitute all terms $\boldsymbol{a}$ by $\hat{\boldsymbol{f}}$ and taking the exponential of the LHS and RHS to yield:

$$p(\boldsymbol{f}|\boldsymbol{y}) \approx \frac{1}{Z} h(\hat{\boldsymbol{f}}) \exp\left\{ -\frac{1}{2}(\boldsymbol{f}-\hat{\boldsymbol{f}})^{\intercal} \underbrace{\left(-\frac{d^2 \log h(\hat{\boldsymbol{f}})}{d\hat{\boldsymbol{f}}^2} \right)}_{\text{denote as } \boldsymbol{A}} (\boldsymbol{f}-\hat{\boldsymbol{f}}) \right\} \tag{3.17}$$

Equation (3.17) is known as an exponentiated quadratic function of $\boldsymbol{f}$ and can be seen as an unnormalised Gaussian distribution with mean $\hat{\boldsymbol{f}}$ and a precision matrix given by negative Hessian denoted as $\boldsymbol{A} = -\frac{d^2 \log h(\hat{\boldsymbol{f}})}{d\hat{\boldsymbol{f}}^2}$. Finally, re-normalising this equation allows us to find $1/Z$ to obtain a Gaussian approximation for $p(\boldsymbol{f}|\boldsymbol{y})$ precisely as:

$$p(\boldsymbol{f}|\boldsymbol{y}) \approx \mathcal{N}(\boldsymbol{f}|\hat{\boldsymbol{f}}, \boldsymbol{A}^{-1}) \tag{3.18}$$

and it can be shown that by computing the derivatives of the log GP posterior (derivation is omitted) the approximation falls precisely under the form of:

$$\begin{aligned} p(\boldsymbol{f}|\boldsymbol{y}) &\approx \mathcal{N}(\boldsymbol{f}|\hat{\boldsymbol{f}}, (\boldsymbol{K_{ff}}^{-1} + \boldsymbol{W})^{-1}) \\ \boldsymbol{W} &= -\frac{d^2 \log p(\boldsymbol{y}|\hat{\boldsymbol{f}})}{d\hat{\boldsymbol{f}}^{\boldsymbol{2}}} \end{aligned} \tag{3.19}$$

In general, the Laplace approximation is attractive for users because it is known to be fast and simplistic amongst the two methods. Further, the Laplace approximation tends to work well for cases where the posterior is well characterized by its mode (Saul, 2016), hence for a Poisson likelihood in the LGCP model this is highly suitable; where as for other examples such as a probit function or Bernoulli likelihood, the Laplace approximation tends to be unsuited for this. A disadvantage of the Laplace approximation is that there is information loss because a Gaussian approximation clearly cannot represent non-Gaussian distributions entirely. Lastly, when working with the Laplace approximation, numerical stability must usually be taken care of (Rasmussen and Williams, 2006).

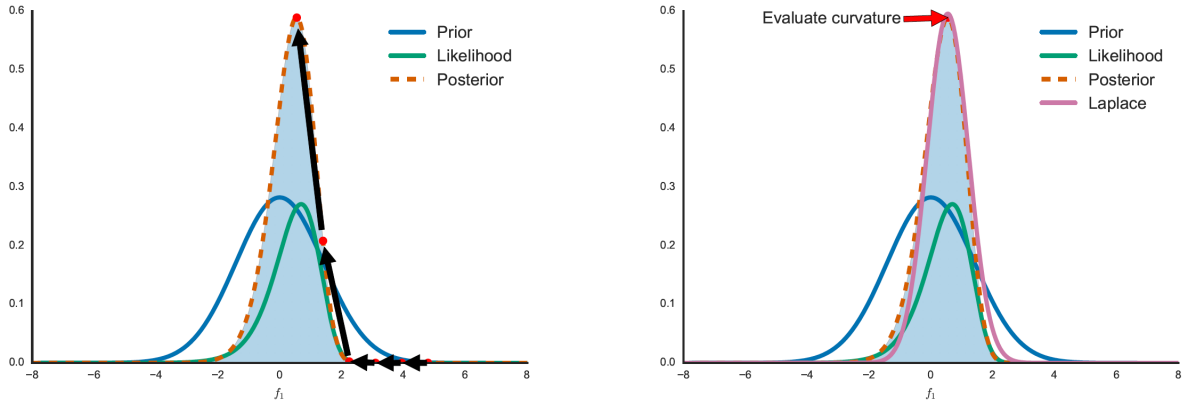


Figure 3.11: Illustration of the Laplace Approximation. Find the mode via Newton optimisation (left), evaluate curvature at mode to approximate posterior (right) (Saul, 2016)

3.4.2 Variational Inference

Unlike the Laplace approximation, Variational Inference (VI) provides a principled method for approximating posteriors for non-Gaussian likelihoods. The intuition behind VI is that we want to solve an intractability problem by turning it into an optimization problem (Blei et al., 2016). This is achieved by minimizing some *measure of closeness* between the approximated distribution denoted by $q(\boldsymbol{f}|\boldsymbol{\theta}_V)$ and the true posterior $(\boldsymbol{f}|\boldsymbol{y})$ over a set of *variational parameters* $\boldsymbol{\theta}_V$ (Nickisch and Rasmussen, 2008).

To start, we first define our measure of closeness known as the Kullback-Leibler (KL) divergence (Kullback and Leibler, 1951). Implicitly the goal of VI is to minimize the KL divergence between the approximated distribution $q(\boldsymbol{f}|\boldsymbol{\theta}_V)$ and true posterior $p(\boldsymbol{f}|\boldsymbol{y})$ (Opper and Archambeau, 2008) given by:

$$\begin{aligned}\mathcal{KL}(q(\boldsymbol{f}|\boldsymbol{\theta}_V)\|p(\boldsymbol{f}|\boldsymbol{y})) &= \int q(\boldsymbol{f}|\boldsymbol{\theta}_V)\log\frac{q(\boldsymbol{f}|\boldsymbol{\theta}_V)}{p(\boldsymbol{f}|\boldsymbol{y})}d\boldsymbol{f} \\ &= \log p(\boldsymbol{y}) + \int q(\boldsymbol{f}|\boldsymbol{\theta}_V)\log\frac{q(\boldsymbol{f}|\boldsymbol{\theta}_V)}{p(\boldsymbol{f},\boldsymbol{y})}d\boldsymbol{f}\end{aligned} \tag{3.20}$$

Doing this requires us to understand that KL-divergence contains the property that it is assymmetric and non-negative such that $\mathcal{KL}(q(\boldsymbol{x})\|p(\boldsymbol{x})) \neq \mathcal{KL}(p(\boldsymbol{x})\|q(\boldsymbol{x}))$ and $\mathcal{KL}(q(\boldsymbol{x})\|p(\boldsymbol{x})) \geq 0$ for any $\boldsymbol{x}$. We want to manipulate the expression in (3.20) by rearranging the LHS in terms of the log evidence instead, because there is no way to directly minimize the KL-divergence (Blei et al., 2016). This yields:

$$\log p(\boldsymbol{y}) = -\int q(\boldsymbol{f}|\boldsymbol{\theta}_V)\log\frac{q(\boldsymbol{f}|\boldsymbol{\theta}_V)}{p(\boldsymbol{f},\boldsymbol{y})} + \mathcal{KL}(q(\boldsymbol{f}|\boldsymbol{\theta}_V)\|p(\boldsymbol{f}|\boldsymbol{y})) \tag{3.21}$$

$$= \int q(\boldsymbol{f}|\boldsymbol{\theta}_V)\log p(\boldsymbol{y}|\boldsymbol{f}) - \mathcal{KL}(q(\boldsymbol{f}|\boldsymbol{\theta}_V)\|p(\boldsymbol{f})) + \mathcal{KL}(q(\boldsymbol{f}|\boldsymbol{\theta}_V)\|p(\boldsymbol{f}|\boldsymbol{y})) \tag{3.22}$$

Since the true posterior $p(\boldsymbol{f}|\boldsymbol{y})$ does not belong to a family of tractable distributions, then the term $\mathcal{KL}(q(\boldsymbol{f}|\boldsymbol{\theta}_V)\|p(\boldsymbol{f}|\boldsymbol{y}))$ in equation (3.22) is not computable and must handled in some way. Using the fact that KL-divergence is non-negative, we decide to remove this term to obtain the most important equation in VI given by equation (3.23).

$$\begin{aligned}\log p(\boldsymbol{y}) &\geq \int q(\boldsymbol{f}|\boldsymbol{\theta}_V)[\log p(\boldsymbol{y}|\boldsymbol{f})]d\boldsymbol{f} - \mathcal{KL}(q(\boldsymbol{f}|\boldsymbol{\theta}_V)\|p(\boldsymbol{f})) \\ &= \int q(\boldsymbol{f}|\boldsymbol{\theta}_V)[\log p(\boldsymbol{f},\boldsymbol{y}) - \log q(\boldsymbol{f}|\boldsymbol{\theta}_V)]d\boldsymbol{f} \end{aligned} \tag{3.23}$$

This is known as the *Evidence Lower Bound* or ELBO for short. In VI, minimizing the KL-divergence is essentially equivalent to maximising the ELBO (Bishop, 2006; Ranganath et al., 2013; Jordan et al., 1999). According to Ranganath et al. (2013), the first term inside the expectation can be seen to encourage parameters of the variational distribution that give high density to configurations of latent variables that explain $\boldsymbol{y}$; the second term encourages parameters that give rise to entropic variational distributions such that it spreads its mass across many configurations.

As part of the VI procedure, in order to maximise the ELBO, we are required to specify a variational family of distribution. The most popular method for this is known as the *mean-field* variational family assumption, which implies that the latent variables are known to be mutually independent and governed by a distinct factor in the variational density (Blei et al., 2016). Finally, assuming that the ELBO expression (3.23) and mean-field family assumption holds, we want to iteratively update the ELBO using an optimization procedure known as *Coordinate Ascent Variational Inference* (CAVI). The main idea behind CAVI is that it optimizes each factor of the mean-field variational density, while holding others fixed. The details and intricacies of CAVI are omitted as it is out of scope of this thesis; however, it is important to understand that the ELBO is in general a non-convex objective function, meaning that CAVI only guarantees convergence to a local optimum (Blei et al., 2016).

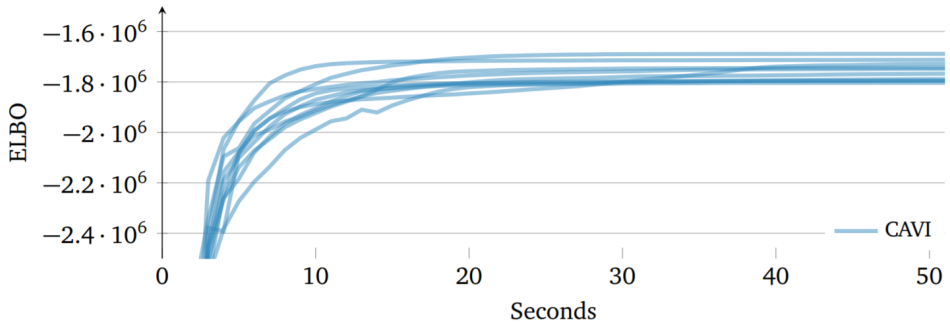


Figure 3.12: Illustration of how different initializations may lead CAVI to find different local optima of the ELBO. (Blei et al., 2016)

To summarize the above, readers should takeaway three essential points. Firstly, VI follows

a principled way of approximating the posterior by optimizing a measure of divergence between the approximation and the true posterior; hence it tends to be more accurate than the Laplace approximation because we do not explicitly find a Gaussian approximation. Secondly, the ELBO must be iteratively optimized to achieve this. This means that VI as an optimization procedure is expected to run slower in general as implicated in Figure (3.12). Third and lastly, VI is non-deterministic in the sense that CAVI does not guarantee a convergence on a global optimum at all times.

3.5 Sparse Gaussian Process

In most academic papers, many authors postulate that the upper limit for practical GPs is around $n \approx 100000$. This is reminded by the computation costs of inverting the covariance function of a GP which has a time complexity of $\mathcal{O}(N^3)$. By far, the most popular method for tackling this computational bottleneck falls under the umbrella known as *sparse approximation* methods (Quiñonero-Candela et al., 2007; Titsias, 2009; Seeger et al., 2014). Specifically, sparse approximations aim to represent the covariance matrix $\boldsymbol{K_{ff}}$ as a low rank approximation. This is done by introducing a set of *inducing points* $\boldsymbol{U} \in \mathbb{R}^{m \times p}$ that live in the same space as $\boldsymbol{X}$ and *inducing inputs* $\boldsymbol{Z} \in \mathbb{R}^{m \times q}$ of size m. These inducing points can often be chosen as a subset of the training data, or even in between points of the observations.

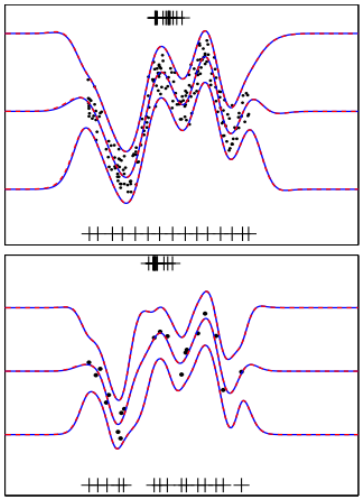


Figure 3.13: Sparse GP with 15 pseudo-inputs (ticks) with 200 (top) and 20 (bottom) training points. (Titsias, 2009)

Let the covariance matrix $\boldsymbol{K_{uu}}$ be the covariance matrix evaluated at the inducing points as before, and similarly for $\boldsymbol{K_{uf}}$ and $\boldsymbol{K_{fu}}$. Then joint distribution over the latent functions $\boldsymbol{f}$ and inducing variables $\boldsymbol{u}$ can be written as

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{u} \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K_{ff}} & \boldsymbol{K_{fu}} \\ \boldsymbol{K_{uf}} & \boldsymbol{K_{uu}} \end{bmatrix}\right) \tag{3.24}$$

and the conditional distribution $p(\boldsymbol{f}|\boldsymbol{u}, X)$ by virtue of a Gaussian distribution can also be written as

$$p(\boldsymbol{f}|\boldsymbol{u}, X) = \mathcal{N}(\boldsymbol{K_{fu}}\boldsymbol{K_{uu}}^{-1}\boldsymbol{u}, \tilde{\boldsymbol{K}})$$

with $\tilde{\boldsymbol{K}} = \boldsymbol{K_{ff}} - \boldsymbol{K_{fu}}\boldsymbol{K_{uu}}^{-1}\boldsymbol{K_{uf}}$.

The trick to reduce the computation burden of the inversion is as follows. If we replace $\tilde{\boldsymbol{K}}$ by some alternative covariance matrix that induces similar covariances between latent functions without taking $\mathcal{O}(N^3)$, then we can reduce its complexity burden. Typically, this cost is reduced from the original $\mathcal{O}(N^3)$ to $\mathcal{O}(NM^2)$ such that $M \ll N$, and the storage complexity falls from $\mathcal{O}(N^2)$ to $\mathcal{O}(NM)$, too. Amongst the many exisiting methods proposed to approximate $\tilde{\boldsymbol{K}}$, some popular methods include the deteministic training conditional (DTC) (Seeger et al., 2014), fully independent training conditional (FITC) (Snelson and Ghahramani, 2006) and variational sparse GP (Titsias, 2009).

3.5.1 Sparse Variational Inference

Variational sparse GP (SVI) as proposed by Titsias (2009) follows the same Variational Inference procedure described in the previous section; it attempts to minimize the KL-divergence between two distributions using a measure of closeness. The KL-divergence between the approximated and true posterior is similar to equations (3.20), except that we introduce a new r.v. to include inducing points $\boldsymbol{u}$ given by:

$$\mathcal{KL}(q(\boldsymbol{f}|\boldsymbol{\theta}_V)\|p(\boldsymbol{f}|\boldsymbol{u},\boldsymbol{y})) = \int p(\boldsymbol{f}|\boldsymbol{u})q(\boldsymbol{u}|\boldsymbol{\theta}_V)\log\frac{q(\boldsymbol{f}|\boldsymbol{\theta}_V)}{p(\boldsymbol{f}|\boldsymbol{y})}d\boldsymbol{u}d\boldsymbol{f} \tag{3.25}$$

The next steps are similar, too. Expand the terms in the RHS in (3.25) as done previously and rearrange everything in terms of the log evidence on the LHS. Following this sketch, it can be shown that the log evidence can be written in expression (3.26). Dropping the term that is incomputable on the far RHS and taking advantage of the fact that the KL-divergence is non-negative and assymmetric, we obtain the final ELBO equation in (3.27).

$$\begin{aligned}\log p(\boldsymbol{y}) = &\int q(\boldsymbol{u}|\boldsymbol{\theta}_V)\left[\int p(\boldsymbol{f}|\boldsymbol{u})\log p(\boldsymbol{y}|\boldsymbol{f})d\boldsymbol{f}\right]d\boldsymbol{u} \\ &- \mathcal{KL}(q(\boldsymbol{u}|\boldsymbol{\theta}_V)\|p(\boldsymbol{u})) + \mathcal{KL}(p(\boldsymbol{f}|\boldsymbol{u})q(\boldsymbol{u}|\boldsymbol{\theta}_V)\|p(\boldsymbol{f},\boldsymbol{u}|\boldsymbol{y})) \qquad (3.26)\\ \log p(\boldsymbol{y}) \geq &\int q(\boldsymbol{u}|\boldsymbol{\theta}_V)\left[\int p(\boldsymbol{f}|\boldsymbol{u})\log p(\boldsymbol{y}|\boldsymbol{f})d\boldsymbol{f}\right]d\boldsymbol{u} \\ &- \mathcal{KL}(q(\boldsymbol{u}|\boldsymbol{\theta}_V)\|p(\boldsymbol{u})) \qquad (3.27)\end{aligned}$$

Here, maximising the ELBO equates to minimizing the KL-divergence between $q(\boldsymbol{f}|\boldsymbol{\theta}_V)$ and $p(\boldsymbol{f}|\boldsymbol{u},\boldsymbol{y})$.

The computation gains of $\mathcal{O}(NM^2)$ in SVI is achieved by handling the variational distribution $q(\boldsymbol{u}|\boldsymbol{\theta}_V)$ in KL-divergence of the RHS of equation (3.27) to compute the variational lower bound for the likelihood $p(\boldsymbol{y}|\boldsymbol{f})$. In the original paper, Titsias (2009) differentiates this bound with respect to $q(\boldsymbol{u}|\boldsymbol{\theta}_V)$ to find an optimal solution. The derivation for the optimal solution is

out of scope of this thesis, but it can be shown that it follows a Gaussian distribution given by:

$$q(\boldsymbol{u}|\boldsymbol{\theta}_V) = \mathcal{N}(\boldsymbol{u}|\boldsymbol{K_{uu}}(\boldsymbol{K_{uu}} + \sigma^{-2}\boldsymbol{K_{uf}}\boldsymbol{K_{fu}})^{-1}\boldsymbol{K_{fu}}\sigma^{-2}\boldsymbol{y}, \boldsymbol{K_{uu}}(\boldsymbol{K_{uu}} + \sigma^{-2}\boldsymbol{K_{uf}}\boldsymbol{K_{fu}})^{-1}\boldsymbol{K_{uu}}) \tag{3.28}$$

for some noise σ^2. Plugging in this result into the variational lower bound in equation (3.27), it is possible to rewrite $p(\boldsymbol{y}|\boldsymbol{f})$ into a more explicit form as:

$$\log p(\boldsymbol{y}|\boldsymbol{X}) \geq \log \mathcal{N}(\sigma^2 I + \boldsymbol{K_{fu}}\boldsymbol{K_{uu}}^{-1}\boldsymbol{K_{uf}}) - \frac{1}{2\sigma^2}\mathrm{Tr}(\boldsymbol{K_{ff}} - \boldsymbol{K_{fu}}\boldsymbol{K_{uf}}) \tag{3.29}$$

In equation (3.29), the bound requires all data to be considered at once and each evaluation of the lower bound or its derivatives is $\mathcal{O}(NM^2)$, hence this is why SVI is faster by a partial magnitude of M^2.

Chapter 4

Experimental results

Objectives:

- ✓ Applying the BOCPDMS algorithm using the LGCP model to the UK property transactions dataset and a synthetic dataset.
- ✓ Optimising the hyper-parameters of the LGCP model and interpreting them.
- ✓ Comparing the computational speed of VI and SVI.
- ✓ Drawing a comparison between non-parametric and parametric models.

To reiterate the contents of this thesis so far, Chapter 2 provided details on how the BOCPDMS algorithm works; whilst Chapter 3 described the Log Gaussian Cox Process (LGCP) model and amongst other things, properties of Gaussian Processes and different approximation methods to overcome intractability: the Laplace approximation, Variational Inference (VI), Sparse Variational Inference (SVI). We now proceed on performing experiments in unification of BOCPDMS and the LGCP model. This is important to provide a comprehensive analysis and to showcase interesting capabilities of the model. Firstly, we consider a case-study of the UK property transactions dataset[1] given by the Office for National Statistics, Since real-world data typically do not have labels, it may be treated as an *unsupervised learning* problem, hence we will attempt to benchmark its performance based on mapping CPs declared to real-world events. Secondly, we consider a sensitivity analysis using synthetic data with artificially constructed CPs. Note that this falls under the context of a *supervised learning* problem. For both cases, only univarate data will be used.

[1] UK property transaction available here.

4.1 UK property transaction dataset

The HM Revenue & Customs presents monthly estimates of historic residential transactions of properties within the UK (England only) between April 2005 to February 2018 for transactions above £40000. We chose this dataset in particular because it contains two interesting narratives:

1. During the financial crisis in 2007-08, this coincided with the housing market slump shown in Figure (4.1) implying that the property market was experiencing high volatility.

2. In 2016, former Prime Minister David Cameron announced the Brexit referendum - this triggered a temporary spike in UK housing transactions shown in Figure (4.1). Unlike the financial crisis, the latter event can be considered an anomaly and should be interpreted differently.

Ideally, a good CPD model is able to discriminate between actual structural changes interpreted as CPs and *anomalies* as discussed as one of the main challenges in Chapter 1. In this case study, we want to study how the LGCP model interprets these events.

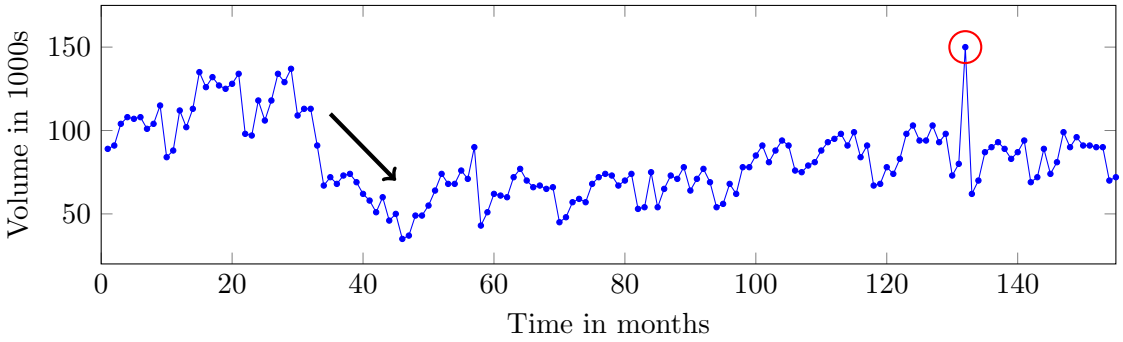


Figure 4.1: The UK property transaction from April 2005 - February 2018. The black arrow indicates the 2007-08 financial crisis; red circle indicates the outlier point from the Brexit referendum announcement.

4.1.1 Hyper-parameter tuning

Unlike parametric models, the LGCP model does not contain explicit hyper-parameters, but is specified by a kernel function by virtue of an exponential GP. Ideally, we want to model its general trend and avoid over-fitting the data - an RBF kernel fits this criteria because of its attractive ability to draw smooth functions and is stationary. However, an RBF kernel alone is insufficient. For predictions beyond the most recent time at $t+1, t+2, \ldots$, the posterior mean of a GP will always eventually converge to zero (Rasmussen and Williams, 2006).

This can be avoided simply by combining other kernel functions. Let $k_1(\boldsymbol{x}, \boldsymbol{x}')$ and $k_2(\boldsymbol{x}, \boldsymbol{x}')$ be PSD kernel functions evaluated at points $\boldsymbol{x}, \boldsymbol{x}'$, then the sum of two kernels is a kernel; the product of two kernels is a kernel. (Rasmussen and Williams, 2006).

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}') + k_2(\boldsymbol{x}, \boldsymbol{x}') \tag{4.1}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}')k_2(\boldsymbol{x}, \boldsymbol{x}') \tag{4.2}$$

Hence we consider the sum of the RBF and a *Bias* or *Constant* kernel $k(\boldsymbol{x}, \boldsymbol{x}') = k_{RBF}(\boldsymbol{x}, \boldsymbol{x}') + k_{BIAS}(\boldsymbol{x}, \boldsymbol{x}')$ where $k_{BIAS}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2_{signal}$, which essentially puts a prior over where the latent mean function (usually zero) to offset it to some suitable constant. During initialisation, the prior hyper-parameters of $k_{RBF}(\boldsymbol{x}, \boldsymbol{x}')$ are set to lengthscale $\ell = 1$ and signal-variance $\sigma^2_{signal} = 1$, and the Hazard of the CP prior is set to be 30. At each timestep, the hyperparameters of kernels are optimized in the GPFlow library, using VI and SVI as methods to approximate the posterior distribution.

4.1.2 Results

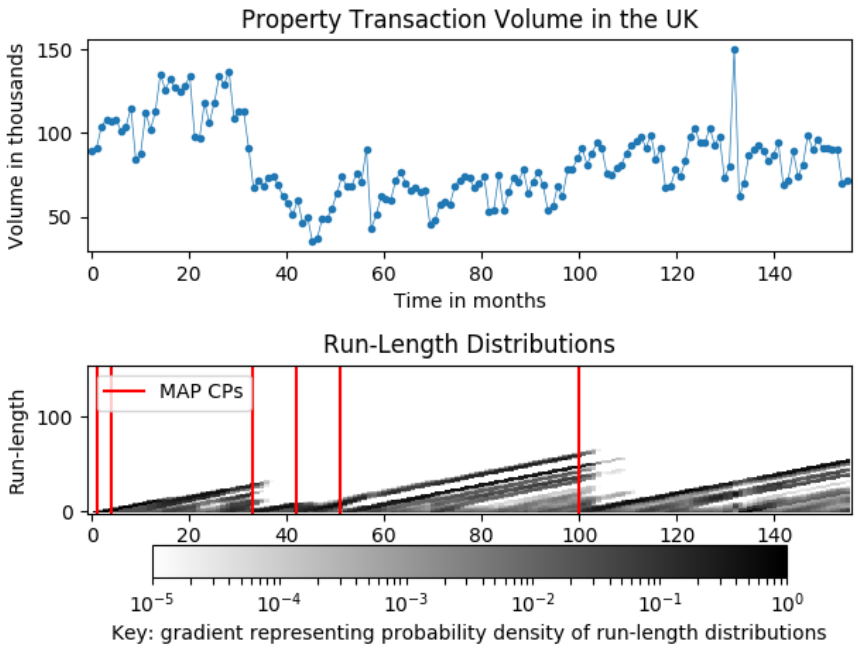


Figure 4.2: BOCPDMS using the LGCP model (with Variational Inference) on the UK property transaction dataset using the sum of an RBF and Bias kernel.

The time taken to compute the CPD results in Figure (4.2) totaled a 115 seconds using VI. Summarized, we first observe that there are 5 CPs declared excluding $t = 0$, and that the beginning of the financial crisis in 2007-08 is declared a CP at $t = 33$. Next, we also observe that the outlier point during the Brexit referendum announcement at $t = 131$ is not

declared as a CP meaning that the LGCP model was able to successfully treat this as an outlier point. Judging by the run-lengths alone, the darker gradients of the run-lengths represent a higher probability density. Focusing on the time of the financial crisis within the period $t = \{33, \ldots, 51\}$, we observe some interesting trends: there are 3 CPs declared within a short period of time, meaning the LGCP model was uncertain on the structural changes happening. This implies that the UK real-estate was experiencing market volatility during the crisis. Table (4.1) contains a summary of all the CPs declared mapped onto major real-world events. With the evidence above, we can argue that the LGCP model is fairly indicative of truth in regards to the structural changes that coincide with real-world events. Mainly, it was able to detect key CPs during the financial crisis and managed to avoid the anomaly at $t = 131$.

Time	Date	Event
33	October 2007	Start of the financial crisis in 2007-08; housing slump in the UK; subprime mortgage crisis in the US.
42	November 2008	Second largest record decrease in housing prices in the UK.
97	April 2013	Mortgage costs in the England starts to fall due to the Bank of England and Government introducing a new funding scheme.
131	February 2016	Former PM David Cameron announces the Brexit referendum, triggering a spike in the UK real-estate market (CP not declared).

Table 4.1: Mapping CPs by the LGCP model to events between April 2005 to February 2018 in the UK property transaction dataset.

Suppose we want to investigate the effects of SVI; that we want to measure the computational gains of sparse approximations. Here, we sample inducing points uniformly directly from the observations of the dataset. Specifically, pick M inducing points uniformly from timesteps between the start and current time. Table (4.1) contains a summary of the computation time in relation to the number of inducing points chosen. Note that if the inducing points are $M = X$ i.e. all points are chosen as inducing points, then this defaults as regular VI instead of SVI.

As the number of inducing points reduces, we observe that the time it takes to compute BOCPDMS reduces too. At the lowest possible inducing points $M = 5$, the maximum possible speed gain is a 24% reduction in time. However, some precaution should be taken because it is known that the representation accuracy of a GP reduces as the number of inducing points reduces. To showcase this, consider the CPD result of SVI with inducing points $M = 5$ shown

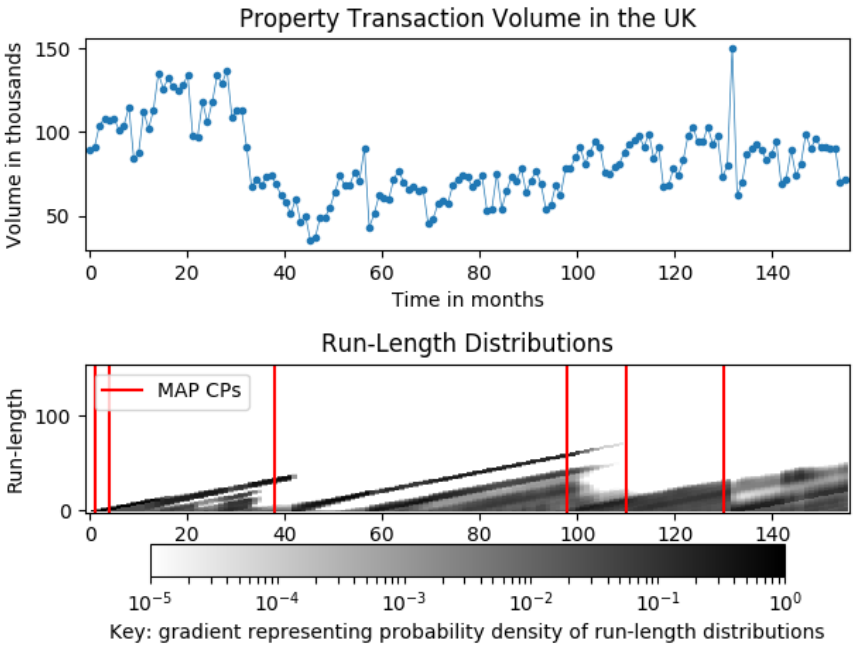


Figure 4.3: BOCPDMS using the LGCP model on the UK property transaction using Sparse Variational Inference. M=5 inducing points are chosen uniformly.

in Figure (4.3). Contrastingly we observe that the previous anomaly point at $t = 131$ is now declared a CP; and the CP detected during the financial crisis period between $t = \{33, \ldots, 51\}$ is now declared much slower at time $t = 38$ compared to $t = 33$ previously using VI in Figure (4.2). For this reason we argue that there is a trade-off between the performance of CPD and the number of inducing points. Ideally, practical users of SVI should choose inducing points such it can represent the data without compromising accuracy.

Method	Inducing Points	Time in seconds
VI	M = X	115
SVI	M = 15	92
SVI	M = 10	90
SVI	M = 5	88
		$\approx 24\%$ speed increase at M=5.

Table 4.3: Effects of Sparse Variational Inference on the computation time of the UK property transaction data.

4.2 Synthetic dataset

Despite showing in the previous section that SVI allows for substantial gains in computation speed, non-parametric models still face the challenge of scalability (Ghahramani, 2013). To illustrate the severity of this, using a parametric model such as the conjugate Poisson Gamma (PG) model available in Appendix (B), we can compute the previous dataset within precisely 0.21 seconds. On a scale this is a 440-fold reduction in computation time against the LGCP model.

This leads an important question of *why* do we need the LGCP model - if parametric models can compute the same dataset exponentially faster with a marginally lower accuracy? For this reason, we consider a synthetic dataset given by Figure (4.4). In particular, this dataset has 7 artificially constructed CPs and contains a cyclical trend between $t = \{40, \ldots, 90\}$. The goal is to compare how the LGCP model fares against the PG model, to demonstrate that both non-parametric and parametric model classes have different use-cases.

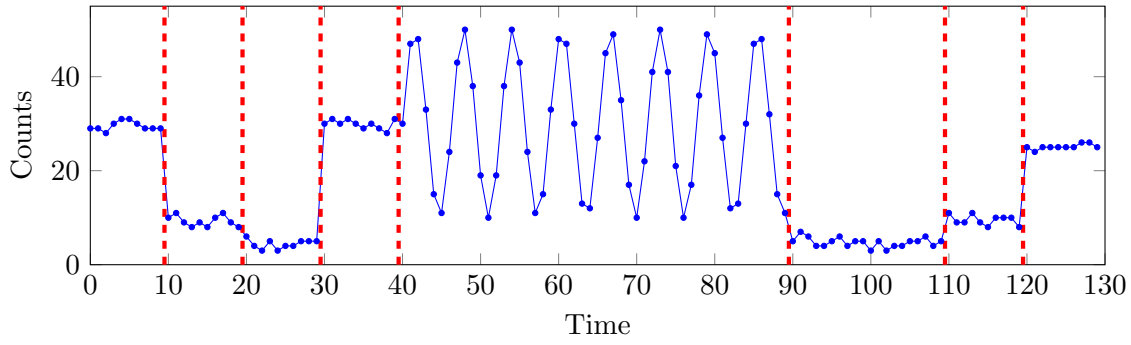


Figure 4.4: Synthetic dataset with a periodic trend from $t = 40$ to $t = 90$. Red dashed lines indicate the true CPs.

4.2.1 Hyper-parameter tuning

After hyper-parameter optimization, we set $\alpha = \beta = 1$ for the PG model. For the LGCP model, we choose an RBF kernel multiplied by a Periodic kernel given by $k(\boldsymbol{x}, \boldsymbol{x}') = k_{RBF}(\boldsymbol{x}, \boldsymbol{x}') \times k_{PER}(\boldsymbol{x}, \boldsymbol{x}')$ in attempt to capture the cyclical trend from $t = 40$ to $t = 90$, and we use the Laplace approximation implemented in the GPy library for the posterior distribution. The Hazard function hyper-parameter is set to 30.

4.2.2 Results

The result of applying the PG model is shown in Figure (4.5). Using the PG model we observe that it identifies CPs within the stationary segments of the dataset; but erroneously detects

multiple CPs within the cyclical part from $t = 40$ to $t = 90$. Regardless of how the hyper-parameters (α, β) are tuned, the PG model is still not able to capture the periodic trend within the dataset.

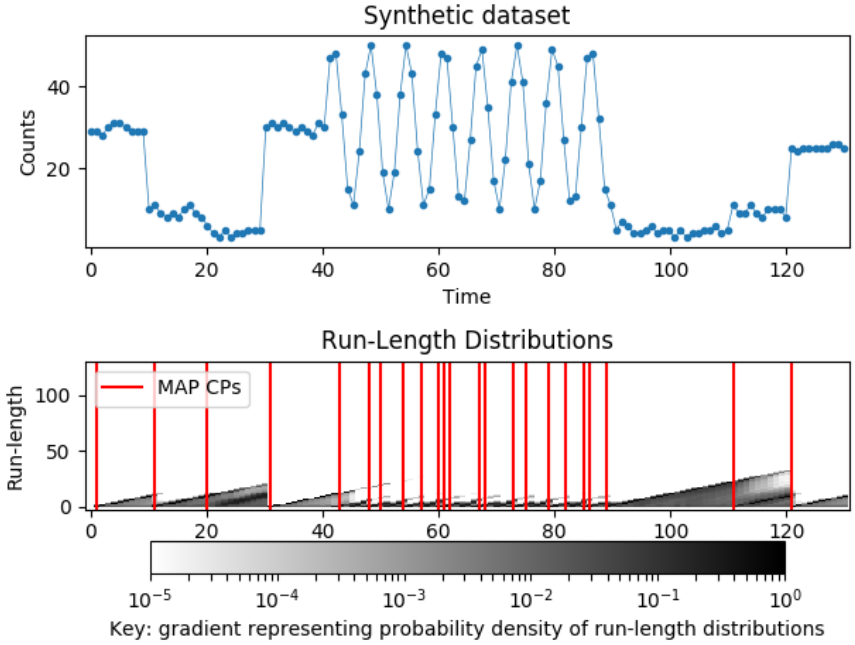


Figure 4.5: BOCPDMS using the PG model on the synthetic dataset with hyper-parameters $\alpha = \beta = 1$.

However, by applying model-selection with the PG model and LGCP model ($|\mathcal{M}| = 2$), we obtain the results in Figure (4.6). In contrast we observe that the CPs at the start of the cyclical trend at $t = 40$ and $t = 90$ are now detected; furthermore the model posterior $p(\boldsymbol{m}|\boldsymbol{y})$ in panel 2 of Figure (4.6) indicates that the LGCP model is selected between $t = 40$ and $t = 90$. This implies that the periodic kernel is able to describe the cyclical dependencies and hence the BOCPDMS algorithm chooses it over the PG model.

Overall despite the PG model being able to compute the dataset much faster than the LGCP model - it is unable to capture the cyclical trend. Given the results above, we can argue that both the PG and LGCP model serve different use-cases; the former being more suited for speed whilst the latter more for flexibility. Alternatively, we can think of the PG model as having only two possible parameters (α, β); whereas the LGCP is specified by dozens of possible kernel functions, all of which can be combined via sums or products to model infinitely unique trends.

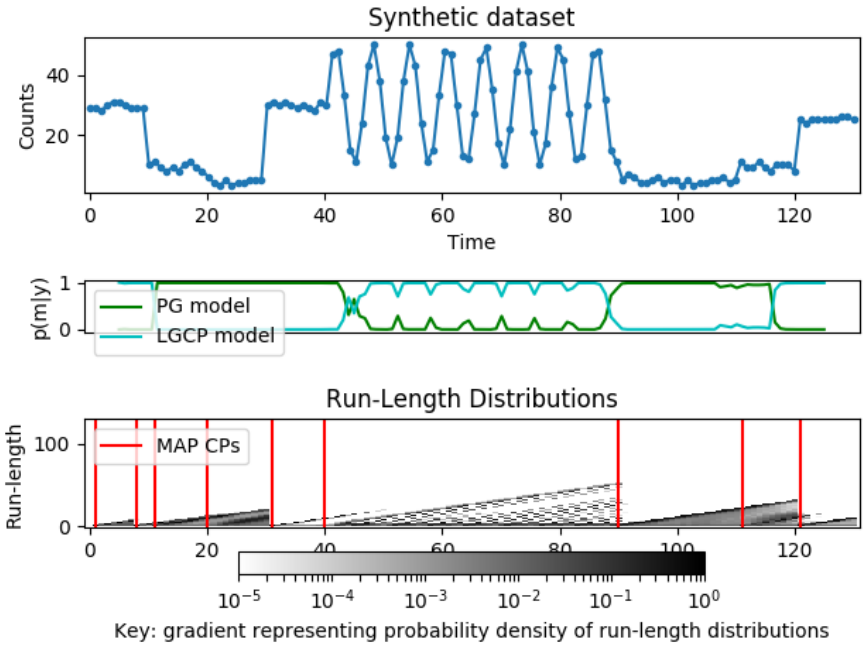


Figure 4.6: BOCPDMS using model selection with the PG model ($\alpha = \beta = 1$) and the LGCP model with a periodic kernel.

Chapter 5

Multivariate extensions

Objectives:

- ✓ Extending the LGCP model into multivariate streams.
- ✓ Outlining the difference between the Linear Model of Coregionalization and Intrinsic Coregionalization Model.
- ✓ Applying BOCPDMS using the LGCP model to the ECMWF climatic dataset.

In general, there are two possible directions for extending an LGCP model to a multivariate setting; either consider constructing individual models for each stream, or a covariance matrix that captures dependencies between multiple streams. In this chapter, we focus on exploring the latter. This is motivated by the fact that many real-world phenomenons are often interconnected amongst each other.

Within the machine learning community, the approach through a joint prediction exploiting the interaction between different components to improve on individual predictions is known as *multi-task learning* (Alvarez et al., 2011). This is also related to another concept known as *transfer learning* which refers to systems that learn by transferring knowledge between different domains (Bonilla et al., 2008). The basic idea behind multi-task learning for GPs, is that given a group of latent outputs $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_D(\boldsymbol{x})$, we can group them into a vector known as a

vector-valued function

$$f(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ f_2(\boldsymbol{x}) \\ \vdots \\ f_d(\boldsymbol{x}) \\ \vdots \\ f_D(\boldsymbol{x}) \end{bmatrix}$$

If we can compute the covariance matrix for this vector-valued function, then we are done, i.e. we can model multiple dependencies. In this chapter, we attempt to cover two different approaches for computing this covariance matrix.

5.1 Multi-task learning

According to (Alvarez et al., 2011), the use of probabilistic models and GPs for multi-output learning was pioneered and developed within the context of geostatistics (Goovaerts, 1997), where predictions over vector-valued output data is known as *cokriging*. The original approach to multivariate modelling is formulated around the *Linear Model of Coregionalization* or LMC for short. The basic idea behind the LMC is that it is formulated under a *sum of separable* (SoS) kernels, which means we can define a more general class of kernels by taking sums and products of valid PSD matrices.

In the LMC framework, outputs are essentially expressed as linear combinations of independent latent functions. Consider the set $\{f_d(\boldsymbol{x})\}_{d=1}^{D}$ of outputs and $\boldsymbol{x} \in \mathbb{R}^p$. Then we can express each component of f_d as:

$$f_d(\boldsymbol{x}) = \sum_{q=1}^{Q} \sum_{i=1}^{R_q} a_{d,q} u_q^i(\boldsymbol{x}) \tag{5.1}$$

where $\{u_q^i(\boldsymbol{x})\}_{q=1}^{Q}$ are latent functions and are independent when $q \neq q'$, and $a_{d,q} \in \mathbb{R}$ are scalar coefficients. In words this means that there are Q groups of latent functions $u_q^i(\boldsymbol{x})$ and the functions within each group share the same covariance $k_q(\boldsymbol{x}, \boldsymbol{x}')$ but are independent (Alvarez et al., 2011), and each group contains R_q samples from the same covariance matrix. This describes the sums over Q and R_q in the RHS of expression (5.1).

The next step shows how to compute dependencies between multiple streams. Given any two components of a vector-valued function $f_d(\boldsymbol{x})$ and $f_{d'}(\boldsymbol{x})$, we can compute the cross-covariance in terms of $u_q^i(\boldsymbol{x})$ as

$$\text{Cov}[f_d(\boldsymbol{x}), f_{d'}(\boldsymbol{x}')] = \sum_{q=1}^{Q} \sum_{q'=1}^{Q} \sum_{i=1}^{R_q} \sum_{i'=1}^{R_q} a_{d,q}^i a_{d',q'}^{i'} \text{Cov}[u_q^i(\boldsymbol{x}), u_{q'}^{i'}(\boldsymbol{x}')] \tag{5.2}$$

Due to the independence of functions $u_q^i(\boldsymbol{x})$, the expression above can be reduced to the following (Alvarez et al., 2011):

$$\text{Cov}[f_d(\boldsymbol{x}), f_{d'}(\boldsymbol{x}')] = \sum_{q=1}^{Q}\sum_{i=1}^{R_q} a_{d,q}^i a_{d',q}^i k_q(\boldsymbol{x}, \boldsymbol{x}') \tag{5.3}$$

$$= \sum_{q=1}^{Q} b_{d,d'}^q k_q(\boldsymbol{x}, \boldsymbol{x}') \tag{5.4}$$

where $b_{d,d'}^q = \sum_{i=1}^{R_q} a_{d,q}^i a_{d',q}^i$.

Finally, the kernel $\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}')$ for the vector-valued function can be expressed as:

$$\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}') = \sum_{q=1}^{Q} \boldsymbol{B}_q k_q(\boldsymbol{x}, \boldsymbol{x}') \tag{5.5}$$

and $\boldsymbol{B}_q \in \mathbb{R}^{D\times D}$ is referred formally as the *coregionalization matrix* such that the coefficients $b_{d,d'}^q$ are elements of $\boldsymbol{B}_q$ in equation (5.4). $\boldsymbol{B}_q$ must be a symmetric and PSD matrix, and the rank of each matrix $\boldsymbol{B}_q$ is R_q which is the number of latent functions that share the same covariance function $k_q(\boldsymbol{x}, \boldsymbol{x}')$. In short, the LMC leads to an SoS kernels that represents the covariance function as a sum of the products of two covariance functions. The coregionalization matrix $\boldsymbol{B}_q$ models dependencies between outputs independently of input vectors $\boldsymbol{x}$; and the kernel $k_q(\boldsymbol{x}, \boldsymbol{x}')$ models input dependencies independently of the set of functions $\{f_d(\boldsymbol{x})\}_{d=1}^D$ (Alvarez et al., 2011).

Notationally, the covariance matrix in the LMC is obtained by taking the *Kronecker product* $\otimes$ of the coregionalization matrix and kernel at each group Q. This has a general form for any $\boldsymbol{X}$ stated by:

$$\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) = \sum_{q=1}^{Q} \boldsymbol{B}_q \otimes k_q(\boldsymbol{X}, \boldsymbol{X}) \tag{5.6}$$

As opposed to the LMC, there is an alternative method for computing the covariance matrix of a vector-valued function. This is known as the *Intrinsic Coregionalization Model* or ICM for short. The ICM is simplified version of the LMC (Goovaerts, 1997). In fact, the set-up is almost identical and notationally easier than the LMC - simply replace the elements $b_{d,d'}^q$ of the coregionalization matrix $\boldsymbol{B}_q$ as $b_{d,d'}^q = v_{d,d'} b_q$ for a constant $v_{d,d'}$. Then the covariance of two outputs $f_d(\boldsymbol{x}), f_{d'}(\boldsymbol{x}')$ becomes (Alvarez et al., 2011):

$$\text{Cov}[f_d(\boldsymbol{x}), f_{d'}(\boldsymbol{x}')] = \sum_{q=1}^{Q} v_{d,d'} b_q k_q(\boldsymbol{x}, \boldsymbol{x}') \tag{5.7}$$

$$= v_{d,d'} \sum_{q=1}^{Q} b_q k_q(\boldsymbol{x}, \boldsymbol{x}') \tag{5.8}$$

$$= v_{d,d'} k(\boldsymbol{x}, \boldsymbol{x}') \tag{5.9}$$

where $k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{q=1}^{Q} b_q k_q(\boldsymbol{x}, \boldsymbol{x}')$. The ICM is a special case of the LMC when $Q = 1$. More generally, the covariance matrix in the ICM takes the form of:

$$\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) = \boldsymbol{B} \otimes k(\boldsymbol{X}, \boldsymbol{X}) \tag{5.10}$$

According to Goovaerts (1997), the ICM is said to be more restrictive than the LMC; mainly due to the fact that it assumes each $k_q(\boldsymbol{x}, \boldsymbol{x}')$ contributes equally to cross covariances for outputs which may not always be true. However, this also means that the computation costs for the ICM are greatly simplified due to properties of the Kronecker product (Alvarez et al., 2011). In particular, it also known that for the ICM, if outputs are considered to be noise-free, predictions under isotopic data is equivalent to independent predictions over each output (Helterbrand and Cressie, 1994). This effect is known as *autokrigeability* (Alvarez et al., 2011; D. Helterbrand and Cressie, 1994). Finally, the computational overheads for multi-output GPs are expected to be even slower. A naive implementation has a time and space complexity of $\mathcal{O}(D^3N^3)$ and $\mathcal{O}(D^2N^2)$ respectively because the it requires the inversion of a $ND \times ND$ matrix.

Apart from the LMC and ICM, another popular approach for computing the covariance matrix for multi-output GPs is the Semiparametric Latent Factor Model (SLFM). The SLFM can be thought of as the LMC where $R_q = 1$ and $Q \neq 1$ (Teh et al., 2005).

5.2 ECMWF climatic dataset

We now proceed to a case-study on climatic spatio-temporal data compiled from the European Center for Medium-Range Weather Forecasts (ECMWF) by courtesy of Cervest. The ECMWF dataset consists of climatic variables: which we focus mainly on *total precipitation*, *soil moisture* and *soil temperature* only[1]. More information about the dataset can be found, here. The goal of applying CPD is to gauge severe weather conditions as a basis for an alert system for agricultural workers. Specifically, being able to notify farmers for the time location of extreme weather conditions and give them estimates about the abundance of extreme events throughout the growing season. For example, a high number of CPs within a short period would imply that there is high volatility and further implications for harsh weather conditions.

The ECMWF dataset spans across a temporal scale of 12 years from 2007 to 2018, containing months April-August in each year; and it has a spatial scale of 133 spatial grids (of size 0.5 by 0.5) across the UK. There are two variants of the dataset: a weekly and monthly version containing 23 weeks and 5 months per year respectively. Each grid is associated with multiple climatic variables: total precipitation, soil moisture and soil temperature which we refer to as '*tp*', '*sm*' and '*st*' for abbreviation; and each climatic variable is measured via count data above

[1]Other climatic variables not included are total cloud cover 'tcc' and 2-meter temperature 't2m'.

or below a certain threshold, and we focus on the upper and lower tail values of this threshold only. These are counts greater than equal the 85-th percentile and less than equal the 15-th percentile which we refer to as '*geq*85' or '*leq*15' for abbreviation[2]. For example, we refer the total precipitation counts above the 85-th percentile as '*tp_geq*85'[3].

We introduce a parametric model for a side-by-side comparison, known as the *Multinomial Dirichlet* (MD) model available in Appendix (B). The MD model belongs to a conjugate class and unlike the PG model in Chapter 4, it is able to take account covariance when assessing the likelihood of a CP - hence being more suited for multivariate analysis. Ideally we want to assess the models on the basis of: the distribution of CPs, the validated performance of each model by tracking the model posterior/one-step ahead prediction errors, and deciding which of the weekly or monthly dataset is more suitable for the proposed alert system.

5.2.1 Hyper-parameter tuning

After hyperparameter optimization, we set the MD model as $\boldsymbol{\eta} = (2, \ldots, 2)^\intercal$ for the weekly dataset and $\boldsymbol{\eta} = (5, \ldots, 5)^\intercal$ for the monthly dataset on the basis that an uninformative prior causes the model to overfit or underfit the data. For the LGCP model we choose the sum of the RBF and Bias kernel $k(\boldsymbol{x}, \boldsymbol{x}') = k_{RBF}(\boldsymbol{x}, \boldsymbol{x}') + k_{BIAS}(\boldsymbol{x}, \boldsymbol{x}')$ for both the weekly and monthly dataset. During initialisation the prior lengthscale ℓ and signal-variance σ^2_{signal} of the kernel is set to 1, and is optimized at each time step. Both models are configured with a prior Hazard of 30.

We chose the Laplace approximation as the inference method, and the Linear Coregionalization Model (LMC) for multi-task learning. The latter simply means that instead of using a single coregionalization matrix as in the ICM:

$$\boldsymbol{K}_{ICM} = \boldsymbol{B} \otimes (\boldsymbol{K}_{RBF} + \boldsymbol{K}_{Bias})$$

we define our kernel to be more general and less restrictive as:

$$\boldsymbol{K}_{LMC} = \boldsymbol{B}_1 \otimes \boldsymbol{K}_{RBF} + \boldsymbol{B}_2 \otimes \boldsymbol{K}_{Bias}$$

5.2.2 Results

For a comprehensive analysis of the weekly dataset, we decide to split the data into individual years $\{2018, 2017, 2016, \ldots\}$ rather than a single contiguous TS - since there are only 23 weeks available per year out of a possible 52 weeks. This allows for a more equal, granular comparison.

[2]Other measurements not included are '*geq*65', '*geq*75', '*leq*35' and '*leq*25'.

[3]Not that for total precipitation, the lower tail '*tp_leq*15' is replaced by '*tp_zero*' instead.

One of the challenges posed in the ECMWF dataset is that there are *multiple* dimensions to consider simultaneously, i.e. there are a 133 spatial-grids, 12 years, 23 weeks per year, 3 features per spatial-grid and 2 measurements per feature, excluding the fact that there are 2 other features and 4 other measurements per feature, totalling to a *maximum possible number of 3990 features across 12 years.*

Instead, we simplify this problem by looking only at a single grid at (grid=1) corresponding to coordinates (50.5, -4.5), and running BOCPDMS across a temporal scale from years 2010 to 2018 on selected features:

$$\{tp\_geq85,\ tp\_zero,\ st\_geq85,\ st\_leq15,\ sm\_geq85,\ sm\_leq15\}$$

Using the MD model and LGCP model, we obtain results in Figure (5.1) and (5.2) respectively. We observe that the MD model in general tends to declare CPs significantly more often than the LGCP model. In the MD model, the CPs declared seemed to represent the structural changes in the climatic variables better, i.e. if we observe all years from 2010 to 2018 in Figure (5.1), the CPs declared constitute to changes in a mean and covariance. Contrastingly for the LGCP model, the CPs declared seemed less reliable in representing structural changes. A good example to illustrate this, if we observe years 2016 and 2017 in Figure (5.2), no CPs are detected from $t \geq 7$ onwards although there are clear shifts in its mean and covariance. We also observe some similarities and differences between the MD and LGCP model in the distribution of CPs. In some years, CPs declared in both models are identical in placements (see years 2011, 2013, 2015 and 2018); whereas in other years, CPs declared are non-identical (see years 2010, 2012, 2014, 2016, 2017).

Alternatively we can apply BOCPDMS across a spatial scale too. We fix the temporal scale at year 2018 and using the same features before, applying the MD and LGCP model across different spatial grids from 1 to 9 yields Figures (5.3) and (5.4) respectively. Here we notice that the MD model in general tends to declare CPs more frequently than the LGCP model, again. The CPs declared using the MD model are more consistent; there is a trend in Figure (5.3) such that 4 CPs are always declared between $t = 0$ to $t = 9$, followed by a gap until the next declared CP around $t = 21$. Contrastingly for the LGCP model, the CPs declared are less consistent. To illustrate this, grids {1, 2, 3, 4, 5, 9} in Figure (5.4) have identical CPs declared at $t = 8$ and $t = 21$; but CPs declared in grids 6 to 8 differ slightly.

For the reasons above, we argue that the MD model is more suitable for CPD when looking at a single grid. As for *why*? Parametric models tend to be responsive in capturing structural changes that happen quickly; whereas non-parametric models are more robust which in turn requires more observations to induce CPs.

After attempting to integrate model selection with the MD and LGCP model ($|\mathcal{M}| = 2$)

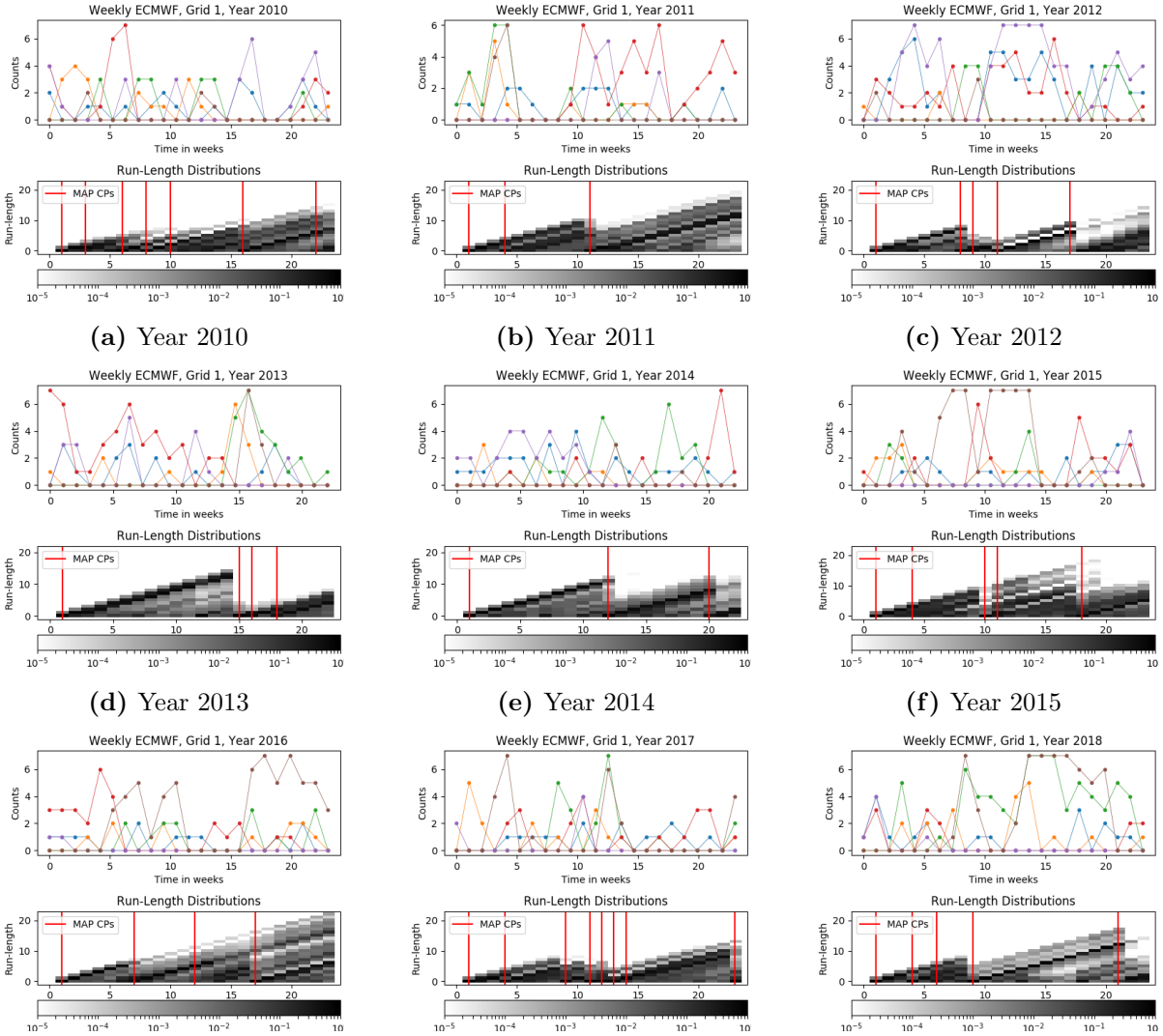
across the same temporal and spatial scale, we found that the model posterior selects the MD model almost always. In addition, the one-step-ahead prediction error in both the MD and LGCP model showed that the MD model was significantly lower in mean error and variance. For this reason, we decide to omit the results for model selection/predictions and focus more on segmentations. In the next study, we investigate the effects of combining multiple grids (spatial) across different years (temporal) into a spatio-temporal process.

We apply BOCPDMS across spatial grids 1-9 and choose upper tail features $\{tp\_85, st\_85, sm\_geq85\}$ per grid to represent a spatio-temporal TS. Using the three most recent years 2018, 2017, 2016 as examples, we obtain results in Figure (5.5), (5.6) and (5.7). Here we observe some interesting findings: the MD model seems to perform poorly in capturing structural changes of the spatio-temporal TS; but the opposite is true for the LGCP model.

To illustrate this, the MD model declares CPs at $t = \{3, 4, 5, 6, 9\}$ in Figure (5.5) but does not declare CPs from $t \geq 10$ onwards although there are clear shifts in the covariance. In constrast, the LGCP model declares CPs at $t = \{7, 13, 18, 23\}$ which corresponds to shifts in the mean and covariance. Similar in Figure (5.6), the MD model declares a CP at $t = 8$ corresponding to a shift in mean, but declares numerous CPs at time $t = \{8, 9, 11, 12, 13, 14\}$ although it appears that mean and covariance is stationary between $t = 8$ to $t = 14$. In contrast, the LGCP model declares CPs at $t = 7$ and $t = 14$ which coincide with shifts in the mean and covariance and does not declare multiple CPs in between.

For completeness, it is worth highlighting that when the MD and LGCP model is applied on the monthly dataset in Figure (5.8) as a contiguous TS from years 2007 to 2018, we find that the latter is more robust and the former is more responsive. Note that the monthly dataset contains only 5 months per year - which is insufficient for comparing individual years, nor as a full contiguous TS.

To condense the findings of the ECMWF dataset, we argue that the LGCP model in general tends to be more robust than the MD model. Both models appear to have different use-cases: in the weekly dataset, we argue that the MD model is more suitable for analysing individual grids across different years or spatial grids in Figures (5.1) and (5.3); whilst the LGCP model is more suitable for analysing spatio-temporal TS with multiple grids in Figures (5.5), (5.6) and (5.7). As for the monthly dataset, there is insufficient evidence to argue which model is more suitable because the dataset is inherently non-contiguous. We end this chapter by proposing that if all 12 months are available in the monthly dataset - it would interesting to conduct a similar analysis across both individual years and a full-contiguous TS.

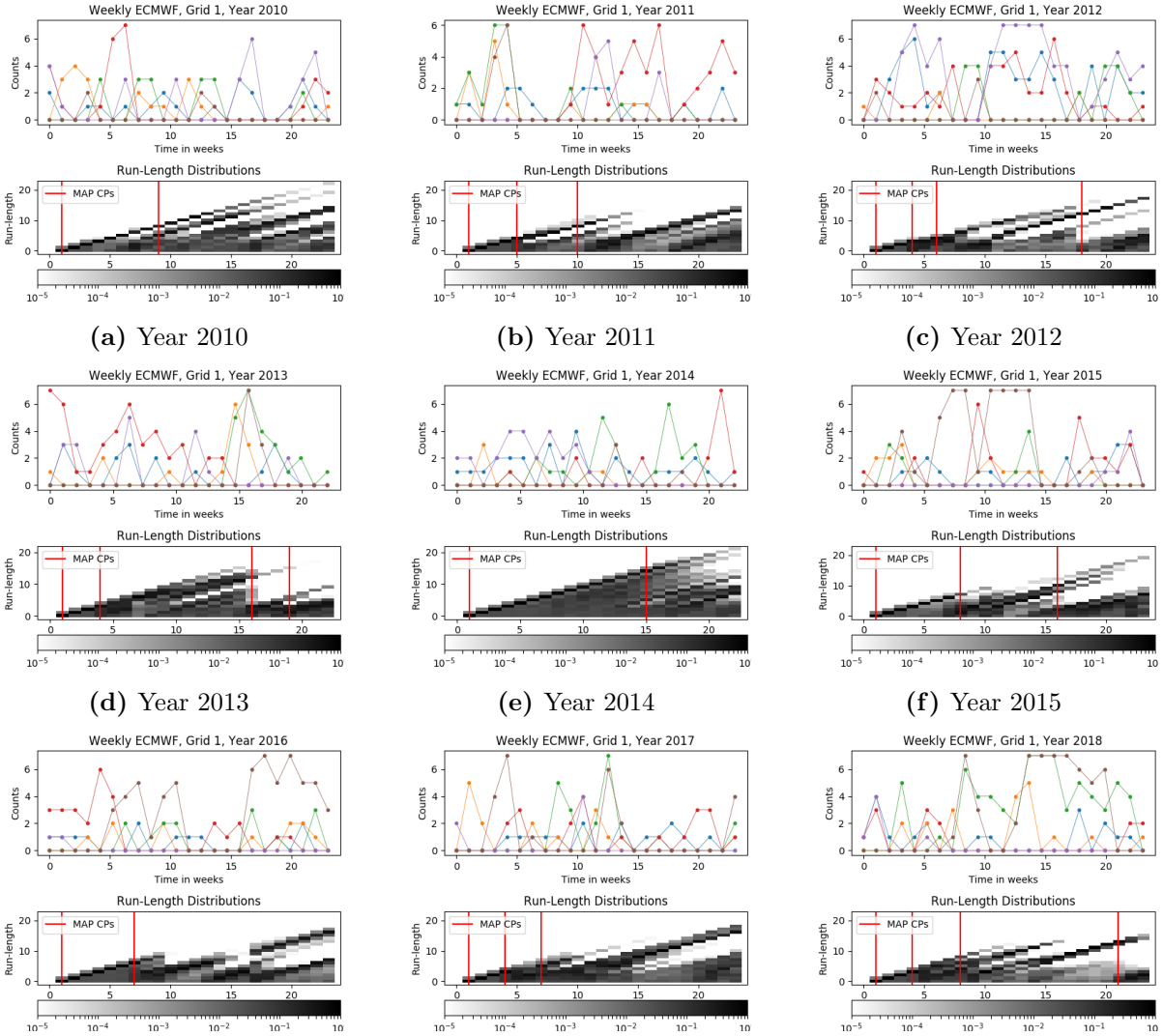


(a) Year 2010 **(b)** Year 2011 **(c)** Year 2012

(d) Year 2013 **(e)** Year 2014 **(f)** Year 2015

(g) Year 2016 **(h)** Year 2017 **(i)** Year 2018

Figure 5.1: BOCPDMS using the MD model across different years, in a single grid, using features $\{tp\_geq85, tp\_zero, st\_geq85, st\_leq15, sm\_geq85, sm\_leq15\}$

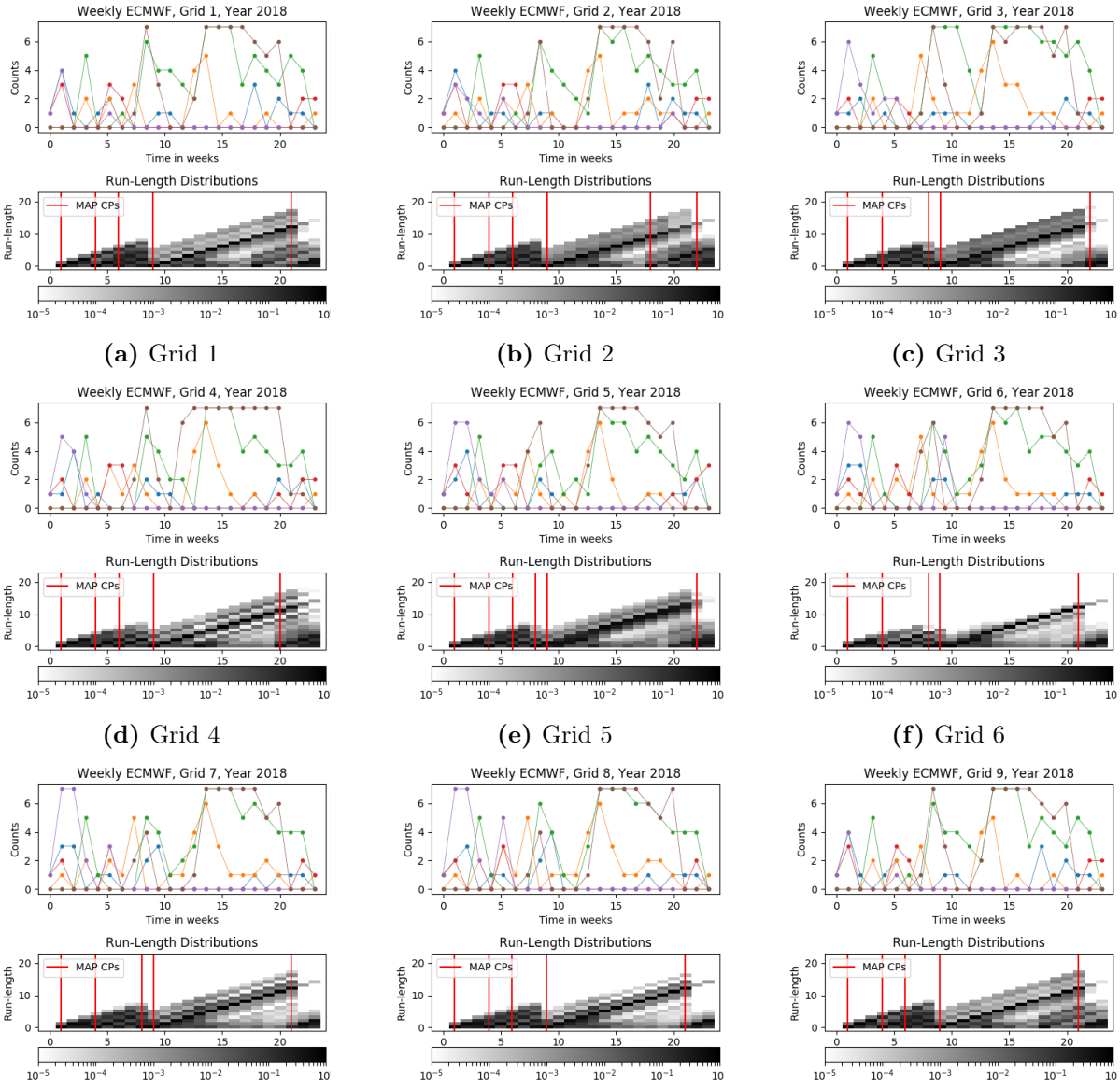


(a) Year 2010 **(b)** Year 2011 **(c)** Year 2012

(d) Year 2013 **(e)** Year 2014 **(f)** Year 2015

(g) Year 2016 **(h)** Year 2017 **(i)** Year 2018

Figure 5.2: BOCPDMS using the LGCP model across different years, in a single grid, using features $\{tp\_geq85, tp\_zero, st\_geq85, st\_leq15, sm\_geq85, sm\_leq15\}$.

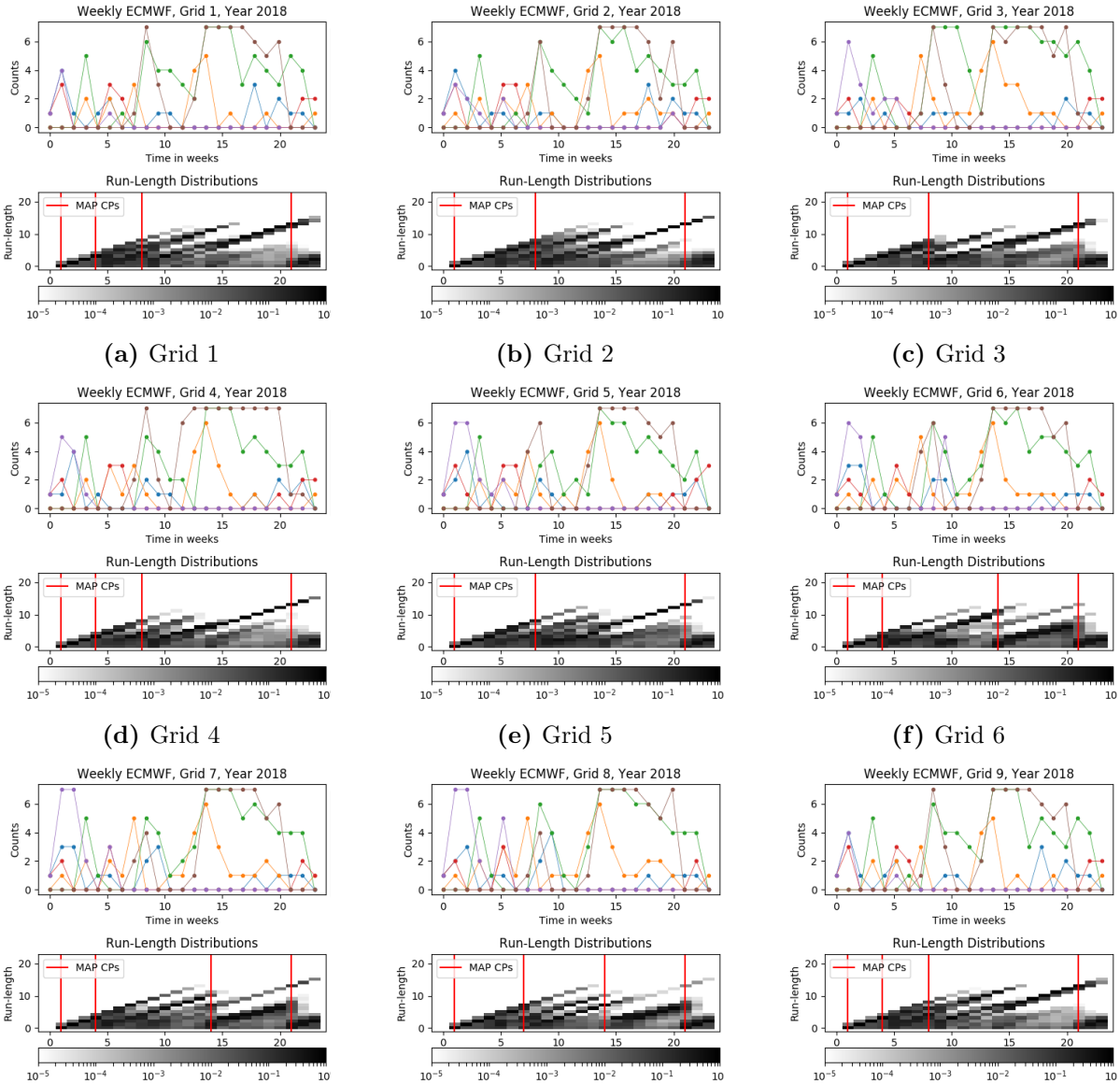


(a) Grid 1 **(b)** Grid 2 **(c)** Grid 3

(d) Grid 4 **(e)** Grid 5 **(f)** Grid 6

(g) Grid 7 **(h)** Grid 8 **(i)** Grid 9

Figure 5.3: BOCPDMS using the MD model across different spatial grids, in year 2018, using features {*tp_geq*85, *tp_zero*, *st_geq*85, *st_leq*15, *sm_geq*85, *sm_leq*15}.

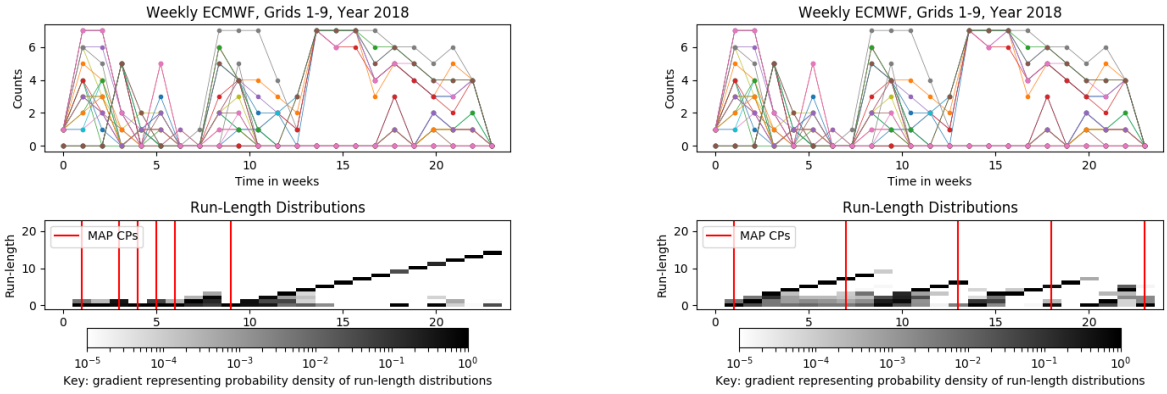


(a) Grid 1 **(b)** Grid 2 **(c)** Grid 3

(d) Grid 4 **(e)** Grid 5 **(f)** Grid 6

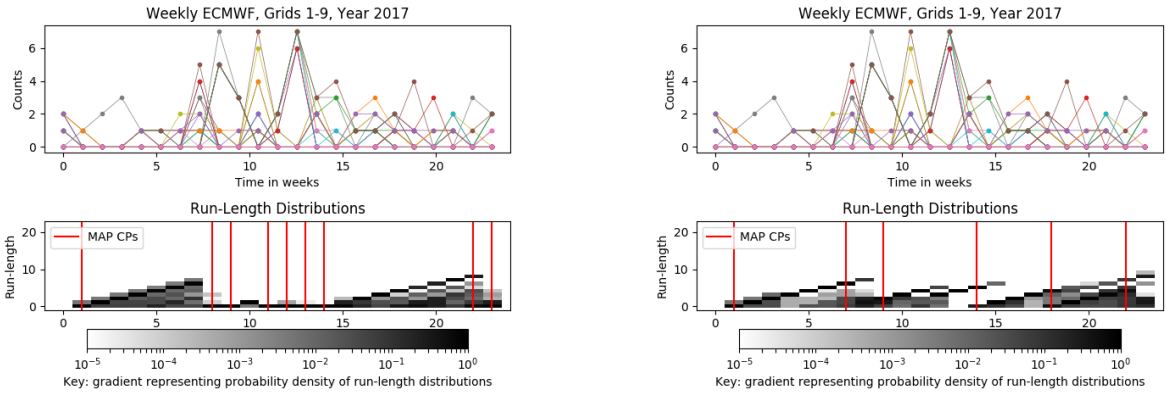
(g) Grid 7 **(h)** Grid 8 **(i)** Grid 9

Figure 5.4: BOCPDMS using the LGCP model across different spatial grids, in year 2018, using features $\{tp\_geq85, tp\_zero, st\_geq85, st\_leq15, sm\_geq85, sm\_leq15\}$.



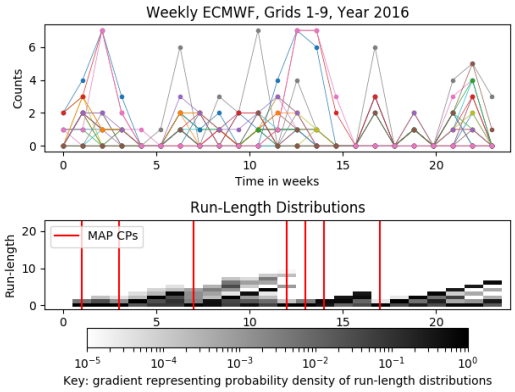
(a) MD model **(b)** LGCP model

Figure 5.5: Comparing the MD and LGCP model in year 2018 using spatial grids 1-9 on features $\{tp\_geq85, st\_geq85, sm\_geq85\}$, totalling 27 features.

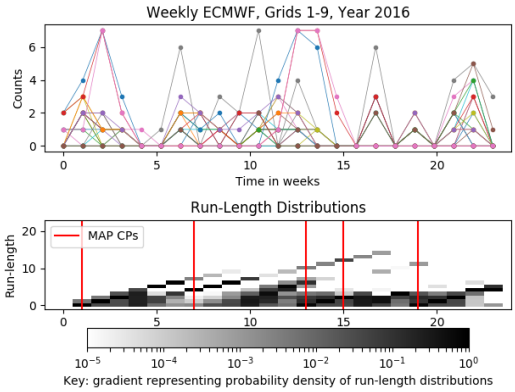


(a) MD model **(b)** LGCP model

Figure 5.6: Comparing the MD and LGCP model in year 2017 using spatial grids 1-9 on features $\{tp\_geq85, st\_geq85, sm\_geq85\}$, totalling 27 features.

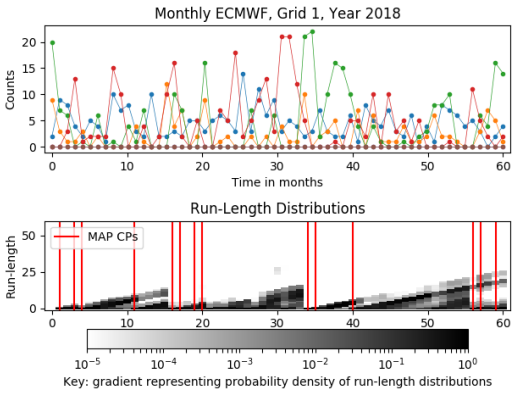


(a) MD model

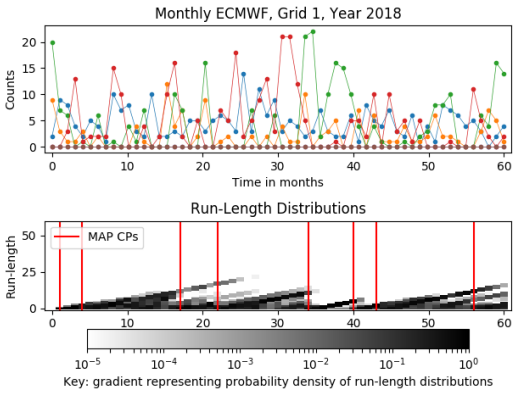


(b) LGCP model

Figure 5.7: Comparing the MD and LGCP model in year 2016 using spatial grids 1-9 on features {*tp_geq*85, *st_geq*85, *sm_geq*85}, totalling 27 features.



(a) MD model



(b) LGCP model

Figure 5.8: BOCPDMS using both models on the monthly dataset, on grid 1, across years 2007 to 2018 using features {*tp_geq*85, *tp_zero*, *st_geq*85, *st_leq*15, *sm_geq*85, *sm_leq*15}.

Chapter 6

Conclusions

Objectives:

- ✓ Summarising the work and personal contributions of this thesis.
- ✓ Discussing any issues faced in reference to the project objectives.
- ✓ Addressing any ethical, social and professional issues.
- ✓ Proposing future directions for this thesis.

Throughout this thesis we have considered how the LGCP model is a powerful, flexible, non-parametric class that has many interesting capabilities within the context of CPD. Mainly, their practicalities are that they are extremely fluid by nature of a stochastic GP intensity, and they are well suited for applications that necessitates multivariate streams and robustness. Its limitations however is that they are inherently slow, hence for certain applications that put speed as a top priority, this becomes less suitable.

6.1 Summary

In this section, we now bring forth a conclusion by recapitulating key points of this thesis. To remind ourselves, Chapter 2 started by introducing basic concepts in Bayesian Inference and TSA, followed by a survey of the original Bayesian On-line CPD framework by Adams and MacKay (2007). In particular, the computational efficiency of the algorithm is achieved by placing efficient CP priors for probabilistic recursions to avoid retrospection. However, one of the main limitations of this framework is that it assumes a single model; which may be insufficient in representing the encompassing data generating process. The works of Knoblauch

and Damoulas (2018) solves this by combining Fearnhead and Liu (2007) and Adams and MacKay (2007). We call the resulting algorithm Bayesian On-line Changepoint Detection with Model Selection (BOCPDMS) and it can be used in three functional ways: TS segmentation, h-step predictions and model inference.

In Chapter 3, Point Processes, or more specifically Poisson Processes were introduced as a building block for count data. Gaussian Processes were reviewed, too. Mainly, they can visualized in terms of latent functions specified by a mean and kernel function and are slow due to the inversion of the covariance matrix (Rasmussen and Williams, 2006). The Log Gaussian Cox Process (Møller et al., 1998) is simply a unification of both. When dealing with non-Gaussian likelihoods in GPs, it turns out that there is no analytically tractable way to compute its integration hence approximation techniques must be resorted. For this reason, we introduced two directions: the Laplace approximation as a fast, scalable method of approximating the posterior distribution as a Gaussian distribution; and Variational Inference (VI) as a principled procedure of optimizing an approximated posterior by minimizing some measure of divergence. As a means to tackle the large computational overheads of GPs, we introduced sparse approximations and variational sparse GP (SVI) (Titsias, 2009) to reduce the time complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(NM^2)$, $M \ll N$.

In Chapter 5, we extended the LGCP model into multivariate streams by introducing the notions of vector-valued functions and multi-task learning. By taking the Kronecker product of the coregionalization matrix and kernels, this allows us to model cross-dependencies in multiple dimensions. The two methods covered in multi-task learning are the Linear Model of Coregionalization (LMC) and Intrinsic Coregionalization Model (ICM); and the ICM is a simplified version of the LMC.

Two real-world and one synthetic datasets were used to test the LGCP model. In the first UK property transaction dataset, we demonstrated the reliability of the LGCP model by mapping real-world events to the CPs declared and attempted to reap the computational gains of SVI. Interestingly enough, we could potentially leverage up to a 24% increase in speed using SVI, but the accuracy of the algorithm is compromised substantially if the number of inducing points are set too low. In the second synthetic dataset, we argued that that despite the LGCP being slower than parametric models such as the Poisson Gamma (PG) model (by 440-folds to be exact), it also has capabilities beyond standard parametric bottlenecks. Finally, we studied the ECMWF dataset in detail using the LGCP model and compared it with the Multinomial Dirichlet (MD) model. We found that the LGCP model is effective in modelling multivariate, spatio-temporal streams and is more robust than the MD model.

6.2 Project management

We now move on to the project evaluation stage - for the reason that it is important to reflect on the obstacles faced throughout the project for future improvements. An exhaustive list of the project objectives is available in Appendix (A).

The first phase of the project was primarily concerning Literature Review (**OBJ1.0**) and Code Integration (**OBJ2.0**). Specifically, trying to understanding core-concepts of the BOCPDMS algorithm such as the recursive MAP segmentation (**OBJ1.2**) and learning to work with the existing codebase by Jeremias Knoblauch (**OBJ2.1**). One of the main challenges was to fix the outdated codebase for the PG and the MDGP model (**OBJ2.2**) which was originally incompatible with the new codebase. This obstacle was by far the most time-consuming and demanding - but after many consultations with the authors of the code, Jeremias Knoblauch and Yannis Zachos, it was eventually debugged by the end of Term 1.

The second phase of the project involved studying the Log Gaussian Cox Process (**OBJ3.0**) and generating results (**OBJ4.0**). Originally, the plan was to study and implement the LGCP in a univariate settings followed by possibly exploring multivariate extensions (**OBJ3.4**) or improving its computation speed (**OBJ3.3**) if time permits. Overcoming the first step was challenging, and tutorials online from the Gaussian Process Summer Schools were helpful, here. Nearing the end of the project, both the multivariate extension and sparse approximations were implemented to tick objectives (**OBJ3.3**) and (**OBJ3.4**).

The external libraries GPy and GPFlow were chosen to implement the GP models rather than writing software from scratch mainly due to time constraints. As for why we chose these two specific libraries, GPFlow is developed using TensorFlow for its core-computation, and is faster because it is engineered with a particular emphasis on being able to exploit GPU hardware. GPy on the other hand is the most popular framework for Gaussian Processes, with an identical syntax with GPFlow and has many tutorials and online forums for support.

Other obstacles faced throughout the project include the implementation of approximation methods VI and SVI using GPFlow. In particular, the library was not built originally for time-series data and it does not support sequential optimizations. The current implementation of the LGCP model using GPFlow requires it to rebuild and re-optimize observations from scratch at a specified refresh rate (i.e. every 10 counts). Secondly, it is worth noting that the numerical stability of GPy and GPFlow is not always guaranteed when handling observations with large values. Both are potential areas of improvements for the project implementation-wise.

6.3 Ethical, social and professional issues

There were no major ethical, social or professional concerns raised throughout the project. Apart from the dataset given by Cervest, which is under their full authorization and permission, any other datasets used did not involve any sensitive information and is from open-licence free sources. Furthermore, no professional issues were raised during supervision hours with Jeremias Knoblauch and Theo Damoulas.

6.4 Future directions

As it stands, the LGCP model's main concern is still its speed with a current complexity of $\mathcal{O}(NM^2)$ using sparse approximations - with the N order of magnitude being the main computational bottleneck. We propose a number of ways to further reduce this complexity. Firstly, Hensman et al. (2013) developed a more scalable version of VI called *Stochastic Variational Inference*. Instead of evaluating the entire ELBO and its derivatives, Hensman et al. (2013) computed a series of lower bound contributions, each on a 'mini-batch' of size N_b, and then summing over the entire contribution. With this, it is possible to reduce the time complexity of VI to $\mathcal{O}(N_bM^2)$. Secondly, we propose the use of a kernel approximation techniques such as the *Kernel Interpolation for Scalable Structured Gaussian Processes* (KISS-GP) by Wilson and Nickisch (2015). The KISS-GP combines properties of the Kronecker and Toeplitz algebra with a new inducing point method to create a sparse approximation to the covariance between training points. Unlike any of the previous approximation methods, this has a $\mathcal{O}(N)$ complexity in both time and space.

Appendix A

Project objectives

Objective	Priority	Description
1.0	*High*	*Literature Review*
1.1	High	Study the core papers for Bayesian On-line Changepoint Detections by Adams and MacKay (2007), Fearnhead and Liu (2007) and Knoblauch and Damoulas (2018).
1.2	High	Understand the recursive MAP segmentation.
1.3	High	Understand the role of Bayesian inference and key assumptions such as the Product Partition Model within CPD.
1.4	High	Study the Poisson Gamma and Multinomial Dirichlet model.
1.5	High	Study Point Processes and its subclasses, i.e. Poisson Proceesses and Cox Processes.
2.0	*High*	*Code Integration*
2.1	High	Experiment with the detector and probability model scripts.
2.2	High	Fix the current Poisson Gamma and Multinomial Dirichlet scripts to integrate the new detector and probability model class objects.
3.0	*High*	*The Log Gaussian Cox Process*
3.1	High	Study Gaussian Processes and its properties, i.e. how the mean and kernel function is used to make predictions.
3.2	High	Implement the Log Gaussian Cox Process in a naive off-line setting using external libraries.
3.3	Medium	Improve the computational efficiency of the Log Gaussian Cox Process model using methods such as sparse approximations.

3.4	Medium	Explore multivariate extensions of the Log Gaussian Cox Process model.
3.5	High	Understand the relation between the stochastic intensity rate in the Log Gaussian Cox Process and a Gaussian Process.
4.0	*High*	*Sensitivity Analysis*
4.1	Medium	Draw a comparison against parametric models (PG and MD model) versus non-parametric models (LGCP model).
4.2	High	Conduct a detailed analysis on the ECMWF climatic dataset by Cervest using the LGCP model.
4.3	High	Analyse the strengths/weaknesses of the LGCP model. Motivate the need for the LGCP model.

Appendix B

Probability distributions

B.1 Gaussian

The multivariate Gaussian or Normal distribution $\boldsymbol{x}$ has joint probability density Rasmussen and Williams (2006):

$$p(\boldsymbol{x}|\boldsymbol{m}, \Sigma) = (2\pi)^{-D/2}|\Sigma|^{-1/2} \exp\big(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m})^\intercal \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{m})\big) \tag{B.1}$$

where $\boldsymbol{m}$ is the mean vector of length D and Σ is a symmetric, PSD covariance matrix of size $D \times D$. For a shorthand expression we write $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{m}, \Sigma)$.

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be joint Gaussian random vectors

$$\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_x\\ \boldsymbol{\mu}_y\end{bmatrix}, \begin{bmatrix}\boldsymbol{A} & \boldsymbol{C}\\ \boldsymbol{C}^\intercal & \boldsymbol{B}\end{bmatrix}\right) \tag{B.2}$$

then the marginal distribution of $\boldsymbol{x}$ and the conditional distribution of $\boldsymbol{x}$ given $\boldsymbol{y}$ is:

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}_x, \boldsymbol{A}) \tag{B.3}$$

$$\boldsymbol{x}|\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}_x + \boldsymbol{C}\boldsymbol{B}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_y), \boldsymbol{A} - \boldsymbol{C}\boldsymbol{B}^{-1}\boldsymbol{C}^\intercal) \tag{B.4}$$

B.2 Poisson

The Poisson distribution X is a univariate discrete distribution, with support $0 \leq k < \infty$ and intensity $\lambda > 0$, given by:

$$\mathbb{P}(X = k) = \frac{\lambda^k \exp^{-\lambda}}{k!} \tag{B.5}$$

$$\mathbb{E}[x] = \lambda \tag{B.6}$$

$$\mathrm{Var}[x] = \lambda \tag{B.7}$$

B.3 Gamma

The Gamma distribution X is a univariate continuous distribution, with support $0 \leq x < \infty$ with parameters shape $\alpha > 0$ and rate $\beta > 0$, given by:

$$\mathbb{P}(X = x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp^{-\beta x} \tag{B.8}$$

$$\mathbb{E}[x] = \frac{\alpha}{\beta} \tag{B.9}$$

$$\text{Var}[x] = \frac{\alpha}{\beta^2} \tag{B.10}$$

where $\Gamma(\alpha)$ is the Gamma function.

B.4 Multinomial

The Multinomial distribution $\boldsymbol{X}$ is the multivariate generalisation of the Binomial distribution. A k-dimensional Multinomial distribution with support $\boldsymbol{x} = (x_1, \ldots, x_k)^\intercal > 0$, number of trials $n = \sum_{i=1}^k x_i$, probability $\boldsymbol{p} = (p_1, \ldots, p_k)$ with $0 < p_i < 1 \ \forall i = 1, \ldots, k$ and $\sum_{i=1}^k p_i = 1$, is given by:

$$\mathbb{P}(\boldsymbol{X} = \boldsymbol{x}) = \frac{n!}{x_1! \ldots x_k!} p_1^{x_1} \ldots p_k^{x_k} \tag{B.11}$$

$$\mathbb{E}[x_i] = np_i \tag{B.12}$$

$$\text{Var}[x_i] = np_i(1 - p_i) \tag{B.13}$$

$$\text{Cov}[x_i, x_j] = -np_ip_j, \quad i \neq j \tag{B.14}$$

B.5 Dirichlet

The Dirichlet distribution $\boldsymbol{X}$ is the multivariate generalisation of the Beta distribution. A k-dimensional Dirichlet distribution with support $\boldsymbol{x} = (x_1, \ldots, x_k)^\intercal > 0$, $\sum_{i=1}^k x_i \leq 1$, with

concentration parameters $\boldsymbol{\alpha} = \alpha_1, \ldots, \alpha_k$, is given by

$$\mathbb{P}(\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{k} x_i^{\alpha_i - 1} \tag{B.15}$$

$$\text{s.t. } B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{k} \alpha_i)}$$

$$\mathbb{E}[x_i] = \frac{\alpha_i}{\sum_{i=1}^{k} \alpha_i} \tag{B.16}$$

$$\text{Var}[x_i] = \frac{\tilde{\alpha}_i(1 - \tilde{\alpha}_i)}{\bar{\alpha} + 1} \tag{B.17}$$

$$\text{Cov}[x_i, x_j] = \frac{-\tilde{\alpha}_i \tilde{\alpha}_j}{\bar{\alpha} + 1}, \quad i \neq j \tag{B.18}$$

$$\text{s.t. } \tilde{\alpha}_i = \frac{\alpha_i}{\sum_{i=1}^{k} \alpha_i}, \bar{\alpha} = \sum_{i=1}^{k} \alpha_i$$

Appendix C

Code

Code was written as part of this project in Python3.6 to implement the LGCP model. There are two variants of the script found under the names `lgcp_model.py` for univariate and `mlgcp_model.py` for multivariate streams respectively, which implements the functions below. To run the scripts, external libraries GPy (version 1.9.6) and GPFlow (version 1.3.0) need to be installed along with the latest version of TensorFlow and NumPy.

- `initialization:` Initialises data structures for storing parameters, probability distribution and objects.
- `evaluate_predictive_log_distribution:` Returns the log densities of $\boldsymbol{y}_t$ using the predictive posteriors for all possible run-lengths $r_t = 0, 1, \ldots, t-1$.
- `evaluate_log_prior_predictive:` Returns the prior log density of the predictive distribution for all possible run-lengths $r_t = 0, 1, \ldots, t-1$.
- `update_predictive_distributions:` Takes the next observation $\boldsymbol{y}_t$ and re-builds/re-optimizes the predictive distributions of the GP model for all possible run-lengths $r_t = 0, 1, \ldots, t-1$.
- `get_posterior_expectation:` Returns the predicted value/expectation from the current posteriors at time t for all possible run-lengths.
- `get_posterior_variance:` Returns the predictive variance from the current posteriors at time t for all possible run-lengths.
- `prior_log_density:` Computes the prior log density of $\boldsymbol{y}_t$.
- `trimmer:` Prunes run-lengths and other key quantities based on the k-most probable run-lengths.

In addition to this, the PG and MD model used in Chapters 4 and 5 are found under the scripts `poisson_gamma_model.py` and `multinomial_dirichlet_model.py` developed by Yannis Zachos. All model classes extend `probability_model.py`, which is an abstract class developed by Jeremias Knoblauch. This class is called upon the `detector.py` object upon receiving every new $\boldsymbol{y}_t$.

Bibliography

Ryan Adams and David J. C. MacKay. Bayesian Online Changepoint Detection. *arXiv e-prints*, art. arXiv:0710.3742, Oct 2007.

Mauricio A. Alvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for Vector-Valued Functions: a Review. *arXiv e-prints*, art. arXiv:1106.6251, Jun 2011.

Samaneh Aminikhanghahi and Diane Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51, 09 2016. doi: 10.1007/s10115-016-0987-z.

Adrian Baddeley. Spatial point processes and their applications. *Stochastic Geometry: Lectures given at the C.I.M.E. 2004, Lecture Notes in Mathematics 1892*, 01 2006.

Daniel Barry and J. A. Hartigan. Product partition models for change point problems. *The Annals of Statistics*, 20(1):260–279, 1992. ISSN 00905364. URL http://www.jstor.org/stable/2242159.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *arXiv e-prints*, art. arXiv:1601.00670, Jan 2016.

G. Bodenstein and H. M. Praetorius. Feature extraction from the electroencephalogram by adaptive segmentation. *Proceedings of the IEEE*, 65(5):642–652, May 1977. ISSN 0018-9219. doi: 10.1109/PROC.1977.10543.

Edwin V Bonilla, Kian M. Chai, and Christopher Williams. Multi-task gaussian process prediction. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 153–160. Curran Associates, Inc., 2008. URL http://papers.nips.cc/paper/3189-multi-task-gaussian-process-prediction.pdf.

Jie Chen and A. K. Gupta. Testing and locating variance changepoints with application to stock prices. *Journal of the American Statistical Association*, 92(438):739–747, 1997. ISSN 01621459. URL http://www.jstor.org/stable/2965722.

H. Chernoff and S. Zacks. Estimating the current mean of a normal distribution which is subjected to changes in time. *Ann. Math. Statist.*, 35(3):999–1018, 09 1964. doi: 10.1214/aoms/1177700517. URL https://doi.org/10.1214/aoms/1177700517.

Siddhartha Chib. Estimation and comparison of multiple change-point models. *Journal of Econometrics*, 86(2):221 – 241, 1998. ISSN 0304-4076. doi: https://doi.org/10.1016/S0304-4076(97)00115-2. URL http://www.sciencedirect.com/science/article/pii/S0304407697001152.

Jeffrey D. Helterbrand and Noel Cressie. Universal cokriging under intrinsic coregionalization. *Mathematical Geology*, 26:205–226, 02 1994. doi: 10.1007/BF02082764.

F. Desobry, M. Davy, and C. Doncarli. An online kernel change detection algorithm. *IEEE Transactions on Signal Processing*, 53(8):2961–2974, Aug 2005. ISSN 1053-587X. doi: 10.1109/TSP.2005.851098.

Paul Fearnhead and Zhen Liu. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605, 2007. doi: 10.1111/j.1467-9868.2007.00601.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2007.00601.x.

Stephen E. Fienberg. When did Bayesian inference become "Bayesian"? *Bayesian Anal.*, 1(1):1–40, 03 2006. doi: 10.1214/06-BA101. URL https://doi.org/10.1214/06-BA101.

Zoubin Ghahramani. Bayesian non-parametrics and the probabilistic approach to modelling. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 371:20110553, 02 2013. doi: 10.1098/rsta.2011.0553.

Mark Gibbs and David J. C. MacKay. Variational gaussian process classifiers. *IEEE Transactions on Neural Networks*, 11:1458–1464, 1997.

Pierre Goovaerts. *Geostatistics for Natural Resource Evaluation*, volume 42. 01 1997.

GPy. GPy: A gaussian process framework in python. http://github.com/SheffieldML/GPy, 2012.

Jeffrey D. Helterbrand and Noel Cressie. Universal cokriging under intrinsic coregionalization. *Mathematical Geology*, 26(2):205–226, Feb 1994. ISSN 1573-8868. doi: 10.1007/BF02082764. URL https://doi.org/10.1007/BF02082764.

James Hensman, Nicoló Fusi, and Neil D. Lawrence. Gaussian processes for big data. *CoRR*, abs/1309.6835, 2013. URL http://arxiv.org/abs/1309.6835.

James Hensman, Alex Matthews, and Zoubin Ghahramani. Scalable Variational Gaussian Process Classification. *arXiv e-prints*, art. arXiv:1411.2005, Nov 2014.

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, Nov 1999. ISSN 1573-0565. doi: 10.1023/A:1007665907178. URL https://doi.org/10.1023/A:1007665907178.

J.F.C. Kingman. *Poisson Processes*. Oxford Studies in Probability. Clarendon Press, 1992. ISBN 9780191591242. URL https://books.google.co.uk/books?id=VEiM-OtwDHkC.

Jeremias Knoblauch and Theodoros Damoulas. Spatio-temporal Bayesian On-line Changepoint Detection with Model Selection. *arXiv e-prints*, art. arXiv:1805.05383, May 2018.

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. Doubly Robust Bayesian Inference for Non-Stationary Streaming Data with β-Divergences. *arXiv e-prints*, art. arXiv:1806.02261, Jun 2018.

S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1): 79–86, 03 1951. doi: 10.1214/aoms/1177729694. URL https://doi.org/10.1214/aoms/1177729694.

G. Lorden. Procedures for reacting to a change in distribution. *The Annals of Mathematical Statistics*, 42(6):1897–1908, 1971. ISSN 00034851. URL http://www.jstor.org/stable/2240115.

Gary M. Koop and Simon M. Potter. Forecasting and estimating multiple change-point models with an unknown number of change points. *Rev. Econ. Stud.*, 74, 12 2004. doi: 10.2139/ssrn.628561.

David J.C. MacKay. Introduction to gaussian processes. *NATO Adv Stud Inst Ser F Comput Syst Sci*, 168, 01 1998.

Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *arXiv e-prints*, art. arXiv:1610.08733, Oct 2016.

Alexander Graeme de Garis Matthews. *Scalable Gaussian process inference using variational methods*. Doctoral thesis, University of Cambridge, 2017. URL https://doi.org/10.17863/CAM.25348.

Thomas P. Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, pages 362–369, San

Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-800-1. URL http://dl.acm.org/citation.cfm?id=647235.720257.

George D. Montanez, Saeed Amizadeh, and Nikolay Laptev. Inertial hidden markov models: Modeling change in multivariate time series. In *AAAI*, 2015.

Jesper Møller. Shot noise cox processes. *Advances in Applied Probability*, 35(3):614–640, 2003. doi: 10.1239/aap/1059486821.

Jesper Møller and Rasmus P. Waagepetersen. Modern statistics for spatial point processes*. *Scandinavian Journal of Statistics*, 34(4):643–684, 2007. doi: 10.1111/j.1467-9469.2007.00569.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9469.2007.00569.x.

Jesper Møller, Anne Randi Syversveen, and Rasmus Plenge Waagepetersen. Log gaussian cox processes. *Scandinavian Journal of Statistics*, 25(3):451–482, 1998. doi: 10.1111/1467-9469.00115. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-9469.00115.

Radford M. Neal. Monte carlo implementation of gaussian process models for bayesian regression and classification. 1997.

Hannes Nickisch and Carl Edward Rasmussen. Approximations for Binary Gaussian Process Classification. *Journal of Machine Learning Research*, 9:2035–2078, October 2008.

Manfred Opper and Cedric Archambeau. The variational gaussian approximation revisited. *Neural computation*, 21:786–92, 10 2008. doi: 10.1162/neco.2008.08-07-592.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954. ISSN 00063444. URL http://www.jstor.org/stable/2333009.

M. B. Priestley. *Spectral analysis and time series / M.B. Priestley.* Academic Press London ; New York, 1981. ISBN 0125649010 0125649029.

Joaquin Quiñonero-Candela, Carl Edward Ramussen, and Christopher K. I. Williams. Approximation methods for gaussian process regression. In *In Large-Scale Kernel Machines, Neural Information Processing*, pages 203–223. MIT Press, 2007.

Rajesh Ranganath, Sean Gerrish, and David M. Blei. Black Box Variational Inference. *arXiv e-prints*, art. arXiv:1401.0118, Dec 2013.

CE. Rasmussen and CKI. Williams. *Gaussian Processes for Machine Learning.* Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA, January 2006.

Yunus Saatchi, Ryan Turner, and Carl Edward Rasmussen. Gaussian process change point models. pages 927–934, 08 2010.

Alan Saul. *Gaussian Process Based Approaches for Survival Analysis*. PhD thesis, University of Sheffield, 2016.

Robert Schlaifer and Howard Raiffa. Applied statistical decision theory. *Journal of the American Statistical Association*, 57, 08 1961. doi: 10.2307/2332787.

Matthias Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse gaussian process regression. 06 2014.

A. F. M. Smith. A bayesian approach to inference about a change-point in a sequence of random variables. *Biometrika*, 62(2):407–416, 1975. ISSN 00063444. URL http://www.jstor.org/stable/2335381.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1257–1264. MIT Press, 2006. URL http://papers.nips.cc/paper/2857-sparse-gaussian-processes-using-pseudo-inputs.pdf.

D. A. Stephens. Bayesian retrospective multiple-changepoint identification. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 43(1):159–178, 1994. ISSN 00359254, 14679876. URL http://www.jstor.org/stable/2986119.

Yee Whye Teh, Matthias W. Seeger, and Michael I. Jordan. Semiparametric latent factor models. In *AISTATS*, 2005.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, 2009. URL http://proceedings.mlr.press/v5/titsias09a.html.

C. K. I. Williams and D. Barber. Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, Dec 1998. ISSN 0162-8828. doi: 10.1109/34.735807.

Andrew Gordon Wilson and Hannes Nickisch. Kernel Interpolation for Scalable Structured Gaussian Processes (KISS-GP). *arXiv e-prints*, art. arXiv:1503.01057, Mar 2015.

Xiang Xuan and Kevin Murphy. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 1055–1062, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: 10.1145/1273496.1273629. URL http://doi.acm.org/10.1145/1273496.1273629.

Yannis Zachos. Bayesian On-line Change-point Detection: Spatio-temporal point processes. unpublished, Apr 2018.