Towards Efficient Bayesian Inference: Cox Processes and Probabilistic Integration

Tom Gunter

Department of Engineering Science

University of Oxford

This dissertation is submitted for the degree of

Doctor of Philosophy

Balliol College

May 2017

I would like to dedicate this thesis to Lydia.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Tom Gunter

May 2017

Acknowledgements

I am most grateful to the organizations and people who have supported me while I researched the material which came to form this thesis. In particular I would like to thank: My supervisors Prof. Steve Roberts and Dr. Mike Osborne; for allowing me intellectual freedom and providing moral, technical and logistical aid over the past few years. My colleagues for providing a sounding board for my ramblings, in particular those who I worked most closely with: Chris Lloyd (with whom I pursued much of this work), Tom Nickson, Rory Beard, and Sam Albanie. My near family, especially my aunt Philippa, my late grandfather Louis, and my parents-in-law Philippa and Andrew. Last but most of all, I owe a deep gratitude to Lydia, for all her support over the years.

I should also like to acknowledge the UK research councils for financing this project, especially the National Environmental Research Council and the Engineering and Physical Sciences Research Council.

Much of the work in this thesis is based on a number of double blind peer-reviewed joint-authored papers. These are listed below, along with a description of my contribution to them:

T. Gunter*, C. Lloyd*, M. A. Osborne, and S. J. Roberts. Efficient Bayesian Nonparametric Modelling of Structured Point Processes. In *Uncertainty in Artificial Intelligence (UAI)*, 2014

I contributed most of the theory and algorithmic development and did the majority of the experimental work, while Lloyd helped fine tune the technical details of the new inference scheme, and took responsibility for some of the experiments. In doing so he also rewrote the majority of the original code base. Osborne and Roberts provided background information, and helpfully suggested appropriate benchmarks.

C. Lloyd*, T. Gunter*, M. A. Osborne, and S. J. Roberts. Variational Inference for Gaussian Process Modulated Point Processes. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML, 2015

I contributed the original theory and algorithmic development alongside some of the experimental work. Lloyd discovered the link to the series representation of the confluent hyper geometric function which made inference computationally stable as well as analytic, solved other issues which cropped up during the development and he also performed the rest of the experiments. Osborne and Roberts provided background information, and were helpful in suggesting applications and appropriate contacts in epidemiology.

C. Lloyd, T. Gunter, T. Nickson, M.A. Osborne, and S.J. Roberts. Latent Point Process Allocation (LPPA). In *The 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016

Lloyd drove the majority of the algorithmic development and experimental work, while I provided the theoretical links to both permanental point processes and marked point processes, and aided the 'bird data' experimental work. The collapsed variational inference scheme built on unpublished work we had both previously pursued independently. Nickson provided a highly optimised Kronecker matrix operations library as well as information on Kronecker structure for Gaussian processes. Osborne and Roberts were insightful as ever in their comments and suggestions during the submission and review process.

T. Gunter, M. A. Osborne, R. Garnett, P. Hennig, and S.J. Roberts. Sampling for inference in probabilistic models with fast Bayesian quadrature. In C. Cortes and N. Lawrence, editors, *Advances in Neural Information Processing Systems (NIPS)*, 2014

I supplied the theory, algorithmic development and experimental work. Osborne was key in suggesting the research direction, and educating me about probabilistic numerics. Garnett provided the citation dataset and one of the figures, and Hennig worked on trying to ameliorate some of the intractabilities associated with squaring a Gaussian process. Roberts provided background information, and was instrumental in the structured rechecking of repeated calculations for corrections made late in the research process.

Abstract

In this thesis we present a variety of new, continuous, Bayesian Gaussian-process-driven Cox process models. These are used to model sparse event data distributed on a continuous domain, where the events may have a tendency to cluster. These find direct use in application areas ranging from disease incidence modelling through to statistical cosmology, where the distribution of galaxies in the universe is weakly clustered due to the effects of dark matter. They may also be deployed in a more abstract sense, for example as a structured prior for network communications.

In previous work, the difficulty of performing inference in Gaussian-process-driven Cox processes has hindered their application to large, high-dimensional datasets. We develop novel and computationally efficient inference schemes for these models as well as our own extensions to them, demonstrating an improvement on the existing state of the art using real data. In particular, we present the first known variational inference scheme for such models, which scales linearly with the size of the dataset.

Spurred on to consider the problem of computationally efficient Bayesian inference in general, we tackle model evidence estimation. Arriving at an accurate measure of model evidence quickly allows for the objective measure of model fit, and ensures we select a set of assumptions which most closely embody the data-generating process.

We deviate from the traditional core Monte Carlo estimator, and instead present a computationally efficient general Bayesian quadrature scheme for model evidence computation. This is the first such scheme which can be shown to be demonstrably wall-clock competitive with state of the art Monte Carlo approaches.

Table of contents

Chapter 1

Introduction

1.1 Motivation

Methods and results developed by the computational statistics and machine learning community have helped yield new findings in a diverse set of application areas: Fields such as astro-statistics (Borucki et al., 2011), zoology (Psorakis et al., 2012), quantitative finance (J.P. Bouchaud, 2009), geology (Chilès and Delfiner, 2012), and epidemiology (Bhatt et al., 2015), have all benefited from improved probabilistic modelling and inference techniques. In a world of finite data, these disciplines typically all aim to build a full probability model of observable and unobservable quantities for a given problem, where such a model is informed by existing theory available to the researcher in question. Data is then collected, and the distributions over the quantities in the model are updated to incorporate the information contained in the observations. Model fit may then be evaluated, or (Bayesian) decision theory can be applied: typically to instruct the researcher how to collect further, maximally informative observations. (Berger, 1985; Diaconis, 1988).

Increasingly, we are also interested in developing semi-autonomous 'agents' (Gmytrasiewicz and Durfee, 2000; Shoham et al., 2007; Stone and Veloso, 2000). Typically these are pieces of software implementing as full a probabilistic model of the environment as is feasible,

alongside a decision making framework. In a similar manner to the researcher previously mentioned, it is usually intended that they acquire data, update the probability model by integrating over all possible (in-model) scenarios through performing inference, and then maximise some notional utility function in order to make a decision on how to proceed. If we are relying on inferences in order to make decisions, then the freedom to ascribe an appropriate model is vital, in order to prevent actions conditioned on poorly calibrated uncertainty estimates.

At a high level, it is clear that in order to achieve these goals at scale and in the most general of cases, we need a flexible computational framework for probabilistic modelling, which places minimal constraint on the size and form of the model, while retaining the ability to cope with large numbers and types of observation. This is, no doubt, a lofty goal, and is arguably the research focus of the majority of the Bayesian statistics community. In this body of work, we focus on two sub-problems within this larger challenge.

- A variety of simple but useful models may be constructed using only base members of the exponential family of probability distributions (Barndorff-Nielsen, 1978). In this manner, one can deal with many data modes, and inference is typically closed form or approximate but efficient[1]. Within the machine learning and statistics communities at large, relatively little[2] work has focused on practical extensions to the simple homogeneous Poisson (Kingman, 1993) likelihood, despite the abundance of sparse event data observed on continuous domains. We focus on building efficient inference schemes for Gaussian process driven Cox processes (Adams et al., 2009; Møller et al., 1998), a form of inhomogeneous Poisson process (although our work could be extended to include renewal processes more generally). We show that our innovations outperform existing approaches on real data from a variety of sources, including

[1]For this discussion, we will assume even Markov Chain Monte Carlo (MCMC) is approximate, as asymptotically exact is, in practical terms, approximate.

[2]Cf. other members of the exponential family.

epidemiology, and demonstrate a particular construction which enables the efficient inference of community structure in a group of birds, given bird-feeder arrival events. Importantly, we maintain the ability to combine many types of observation into one larger model in a flexible and efficient fashion, through retaining a variational Gaussian process framework.

- Recognising the occasional limitations of the variational approaches we converged on for inference in our Cox process work, and furthermore acknowledging the fact that the alternative—randomised sampling—is, from a theoretical point of view, fundamentally inefficient (Briol et al., 2015b; O'Hagan, 1987); we attempt to construct a general purpose, computationally efficient Bayesian quadrature algorithm (O'Hagan, 1991; Osborne, 2010; Rasmussen and Ghahramani, 2003). We demonstrate that even a naïve implementation is highly competitive with state of the art semi-randomised sampling approaches.

In summary, assuming the appropriateness of tools from probabilistic modelling for both: informing theory given data, and for providing well calibrated uncertainty estimates to aid decision making; the motivation behind this work is clear: enable researchers to better incorporate commonly seen inhomogeneous Poisson observations into their models, and explore Bayesian schemes for the actual computations involved in inference.

1.2 Background

Throughout this thesis, we will assume that the reader is reasonably familiar with probability theory (Bishop, 2007; Grimmett and Stirzaker, 2001; MacKay, 2002). We provide a light introduction in the following paragraphs, where the notation is kept deliberately semi-formal, however should the reader find this lacking in some sense the references provided list a few texts which provide an excellent grounding.

In general we will provide specific background material at the start of each chapter. As a result each chapter is roughly self contained, up to and including the notation used. In this section, we focus on delivering a suitable primer on point processes, in order that the reader may be better able to see where our work fits in with existing art. We also include a compact introduction to the Gaussian process (GP), but do not expect it to cover the many interesting interpretations of the humble GP. These may, for the most part, be found in Rasmussen and Williams (2006).

1.2.1 Probability, Inverse Probability and Bayesian Inference

The classical definition of probability as synonymous with the long-run frequency of outcome in a random experiment is not one which we adopt for the majority of this thesis. We rather pursue the more general definition of probability—which pertains to the degree of belief in a preposition given evidence and our assumptions. In order for this view to align with probability theory the degrees of belief must satisfy the Cox axioms (Cox, 1946). These simply ensure that: 1. degrees of belief may be ordered, and so may be mapped onto $\mathbb{R}^1$, 2. a function exists which maps between the degree of belief in a proposition and its negation, 3. degree of belief in a conjunction of propositions factorises into a conditional proposition and the degree of belief in the conditioning proposition.

For simplicity the definitions and descriptions below cover discrete probability. All the same properties exist for non-atomic probability, however the operators are adapted to cope with sets of uncountably infinite cardinality.

Basic definitions

Let us consider performing an experiment which may yield a countable set of possible outcomes Ω, known as the 'sample space' (where a single possible outcome is denoted ω). $F_\Omega \in 2^\Omega$ is the σ-algebra on Ω—in other words a particular collection of subsets of Ω, where each subset is termed an 'event'. We also define a probability measure $P_\Omega : F_\Omega \to [0, 1]$

which assigns probabilities to events, where P_Ω is countably additive. Together these three concepts form what is known as a probability 'space' or 'triple': $(\Omega, F_\Omega, P_\Omega)$.

We may extend these notions to arbitrary joint probability spaces, for example $\Omega\Phi$, where each outcome is an ordered pair $\{\omega, \phi\}$, after which all else follows as before to yield $(\Omega\Phi, F_{\Omega\Phi}, P_{\Omega\Phi})$.

Marginal probability If our experiment produces a joint probability space, we may want to calculate probabilities over just one variable, in effect taking into account all possible values of our joint variable. This is known as a 'marginalising' operation: If we define $x \in F_\Omega$, $y \in F_\Phi$, then we may obtain the marginal probability $P_\Omega(x)$ from $P_{\Omega,\Phi}(x, y)$ by summation:

$$P_\Omega(x) = \sum_{y \in F_\Phi} P_{\Omega,\Phi}(x, y). \tag{1.1}$$

Conditional probability We often wish to calculate the probability of seeing an event, having taken an observation of some other, related and non-deterministic (given our current information set) property of the world. For example, let us assume that our experiment involves measuring the speed of a steel ball as it exits the bottom of a ramp it has rolled down. We do not know the slope of the ramp, however we may want to be able to compute the probability of the ramp having a certain slope, after observing a given exit velocity. Allowing for the vagaries of the real world (friction, non-perfectly round ball, non homogeneous ramp etc.) this will still be non-deterministic (even at this scale). In order to compute such a probability, we must perform a conditioning operation:

$$P_{\Omega,\Phi}(x|y) = \frac{P_{\Omega,\Phi}(x, y)}{P_\Phi(y)}. \tag{1.2}$$

Independence Should our joint probability function factor,

$$P_{\Omega,\Phi}(x, y) = P_\Omega(x) P_\Phi(y), \tag{1.3}$$

then we can say that the two variables are 'independent' of each other—this implies that we gain no information about one variable having made observations of the other. Notions of conditional independence follow as one might expect given the introduction of a third related set of outcomes, X, the associated probability space and $z \in F_{\mathrm{X}}$:

$$P_{\Omega,\Phi,\mathrm{X}}(x, y|z) = P_{\Omega,\mathrm{X}}(x|z)P_{\Phi,\mathrm{X}}(y|z). \tag{1.4}$$

We now define a set of assumptions on which the probabilities are based, namely I, and also specify the three most important rules in applied probability: the sum rule, the product rule and Bayes' theorem. These are effectively instances of the marginalising, factoring and conditioning operations we specified above.

Sum rule

$$P_{\Omega}(x|I) = \sum_{y \in F_{\Phi}} P_{\Omega,\Phi}(x, y|I). \tag{1.5}$$

Product rule

$$P_{\Omega,\Phi,\mathrm{X}}(x, y|z, I) = P_{\Omega,\mathrm{X}}(x|z, I)P_{\Phi,\mathrm{X}}(y|z, I). \tag{1.6}$$

Bayes' rule and inverse probability Bayes' rule involves simply computing one conditional given another, and relies on symmetry of factorisation. For this definition we will further relax notation, and assume that in addition to the above we have some data D, a probability function 'likelihood/generative model' for how that data may have been generated (parameterised by Θ), and a 'prior' probability function on Θ. Armed with this, we may write down the 'posterior' distribution of Θ conditioned on D and I as:

$$P(\Theta|D, I) = \frac{P(D|\Theta, I)P(\Theta|I)}{P(D|I)}, \tag{1.7}$$

which is interpreted as:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}. \tag{1.8}$$

Bayes' rule is synonymous with inverse probability and 'inference'. Forward probability involves computing functions of quantities which depend on the data produced by the generative model, while inverse probability requires us to conditionally compute the properties of the generative model itself. Bayes' rule as stated is deceptive in its simplicity, and as we shall later see in the vast majority of cases computing the evidence, (and thereby the posterior), poses an intractable problem.

Inference

The process of inference can at its core be reduced to integration—for instance, the integral associated with computing the model evidence,

$$M = P(D|I) = \int P(D|\Theta, I)P(\Theta|I)\,\mathrm{d}\Theta, \tag{1.9}$$

as well as a conditioning operation.

In the many cases where the above integral and conditioning operations are not analytic, a variety of techniques exist to make inference approximate but tractable. We cover the broad classes of approach below, drilling down into specifics on those chosen for this thesis.

Sampling based inference

Two broad classes of sampling based techniques exist: those which provide a convergent estimator for the integral, and those which directly generate samples from the posterior (i.e. approximate the result of the integration and conditioning as a sum of delta functions).

Integral approximation techniques naturally also include traditional quadrature/cubature (particularly for lower dimensional domains of integration—e.g. the empirical Riemannian integral) and Chapter 5 describes a novel and efficient approach along these lines. The most commonly used set of methods are based around a core Monte Carlo estimator (Neal, 2010), however.

Monte Carlo integration The key idea is that we may approximate M as:

$$M \approx \frac{1}{n} \sum_{i=1}^{n} P(D|\Theta = \theta_i, I), \tag{1.10}$$

where θ_i are drawn independently from $P(\Theta|I)$. The fact that this is a strongly consistent estimator of M follows directly from the strong law of large numbers assuming that $|P(D|\Theta, I)|$ has finite expected value (Neal, 2010), and we can use the standard version of the central limit theorem to show that the standard deviation of the estimator is $\mathcal{O}(n^{-\frac{1}{2}})$.

For many, the appeal of Monte Carlo is that the error in the estimate is not a function of the dimensionality of the domain of integration. The limiting arguments of consistency, lack of bias, and independence from the dimension of the space give many an applied statistician a (potentially false) feeling of reassurance. It is clear that should we be able to make more specific claims about the behaviour of $P(D|\Theta, I)$, then a faster converging algorithm may be designed.

Should we find ourselves unable to directly sample from $P(\Theta|I)$, we may still evaluate M in Monte Carlo fashion via the use of an importance re-weighting trick Neal (2010): $P(\Theta|I) = \frac{P(\Theta|I)}{\pi(\Theta)}\pi(\Theta)$, enabling one to integrate against any density.

As the precision of the Monte Carlo estimator scales as $\frac{1}{n}\text{Var}[P(D|\Theta = \theta_i, I)]$, importance sampling can enable us to reduce the variance of the estimator *if* we can select a sampling distribution which concentrates mass on areas of the domain which highly influence the value of M. In Chapter 5 we present an approach to effectively infer such a distribution, allowing us to improve the effective sampling efficiency.

There are many elaborations on this core Monte Carlo estimator which lead us to: Annealed Importance Sampling (AIS) (Neal, 2001), nested sampling (Skilling, 2004), and bridge sampling (Meng and Wong, 1996) amongst others. They incur parameters which must be tuned, but as a result improve the convergence rate of the estimator under certain conditions.

Markov chain Monte Carlo MCMC is used to generate samples from the exact posterior. The premise is simple: define a Markov chain on Θ, which has as its stationary distribution $\pi(\Theta) = P(\Theta|D, I)$—the target posterior of interest. In order to do this, we must design a transition kernel which guarantees the correct stationary distribution independent of the starting location. As in the case of Monte Carlo, there are numerous complications which one can adopt to (hopefully) improve the performance of the algorithm, but the original concept by Metropolis et al. (1953) (known as the Metropolis-Hastings algorithm) is a good starting point.

A Markov process is determined by a transition kernel/probability. This is the probability of reaching a new state given the current state, $P(\theta_{i+1}|\theta_i)$. In order to guarantee that we will eventually generate samples from the stationary distribution, two requirements on $P(\theta_{i+1}|\theta_i)$ must be met: 1. To ensure existence of $\pi(\Theta)$ it is sufficient but not necessary that $P(\theta_{i+1}|\theta_i) = P(\theta_i|\theta_{i+1})$ (this implies 'detailed balance' is observed). 2. $\pi(\Theta)$ must be unique which is implied if every state is aperiodic and positive recurrent—we do not return to a state periodically, and we will return to any given state with probability 1 in finite expected time.

In the Metropolis-Hastings algorithm we achieve the above conditions by writing

$$P(\theta_{i+1}|\theta_i) = g(\theta_{i+1}|\theta_i)A(\theta_{i+1}|\theta_i) + (1 - A(\theta_{i+1}|\theta_i))\delta(\theta_i), \tag{1.11}$$

where $g(\theta_{i+1}|\theta_i)$ is a proposal distribution, and $A(\theta_{i+1}|\theta_i)$ determines the probability that any given proposal is accepted. $A(\theta_{i+1}|\theta_i)$ is naturally chosen to ensure that we converge to the desired $\pi(\Theta)$, and a common choice is:

$$A(\theta_{i+1}|\theta_i) = \min\left(1, \frac{P(\theta_{i+1}|I)g(\theta_i|\theta_{i+1})}{P(\theta_i|I)g(\theta_{i+1}|\theta_i)}\right). \tag{1.12}$$

We therefore accept each proposed move if $A \geq 1$, otherwise we accept with probability given by A.

Many variants, adaptations and extensions of Metropolis-Hastings exist, the majority are detailed in Neal (2010).

As in the case of our Monte Carlo discussion, a correctly specified instance of MCMC is guaranteed to eventually both reach the stationary distribution, and achieve an appropriately fine grained representation of the posterior through a composition of delta functions. Bounds on the number of iterations required before we can be satisfied to within some specified degree of error are highly problem- and dimension-specific however, and in practice most users rely on a composition of heuristic indicators to decide when to stop.

In addition, several post processing steps are typically applied to reduce effects due to sample autocorrelation, and decrease the finite sample bias induced by the starting position.

Despite the above, in situations where the domain is high dimensional and we are unable to describe (and therefore exploit) high level properties of the likelihood surface, MCMC remains the algorithm of choice for most practitioners. It is for these reasons that we choose a MCMC based approach in Chapter 2.

Approximate inference

Although we title this section 'approximate' inference, in practice these methods are potentially no more approximate than MCMC based approaches. Perhaps a better description in light of our foray into Monte Carlo techniques would be 'deterministic' methods, as in each case we will effectively convert an integral problem into a deterministic energy minimisation problem, which we will then solve either directly or via iterative gradient descent.

Our favoured approach in Chapters 3-4 is known as variational Bayes (Bishop, 2007; Jaakkola, 2000; Jordan et al., 1999; Kapur, 1989) which has its origins in the 18^{th} century work of Euler and Lagrange, but we pursue a more general introduction here.

Let us assume that we wish to functionally approximate the complex and possibly multivariate posterior probability distribution $P(\Theta|D, I)$ using $q(\Theta)$. In order to do so we select a $q(\theta)$ which minimises a divergence measure $D(P\|q)$ between the two distributions. The most common divergence measure chosen is the Kullback-Leibler (KL) divergence in the form specified below, which yields a variational Bayes approximation:

$$D_{KL}(q\|P) = \int_{\Theta} q(\Theta) \log \frac{q(\Theta)}{P(\Theta|D, I)} \, \mathrm{d}\Theta. \tag{1.13}$$

More generally, we can write down the so called α-divergence (Amari and Nagaoka, 2007). This is in fact a family of divergences, where choosing specific values for α allows us to recover traditional methods:

$$D_{\alpha}(P\|q) = \frac{\int_{\Theta} \alpha P(\Theta|D, I) + (1-\alpha) q(\Theta) - P(\Theta|D, I)^{\alpha} q(\Theta)^{(1-\alpha)} \, \mathrm{d}\Theta}{\alpha(1-\alpha)}. \tag{1.14}$$

Allowing α to tend to 0 yields a divergence which when minimised will result in a variational Bayes approximation to $P(\Theta|D, I)$. Assuming a lack of complete freedom in the functional form of $q(\Theta)$ such an approximation will typically underestimate the true support of $P(\Theta|D, I)$—it is known as a 'zero-forcing' approximation. If we instead let α go to 1, we arrive at the reverse KL divergence. Minimising this divergence exactly would correspond to performing perfect inference, which is typically intractable. We can however do so for an iteratively moment-matched approximation to the posterior, which yields a belief or expectation propagation approximation (Minka, 2001) (assuming we are willing to assume a posterior which factorises over the datapoints). Expectation propagation in comparison to variational Bayes will more accurately model the bulk of $P(\Theta|D, I)$, however assuming restrictions on the form of $q(\Theta)$ and given a truly multimodal $P(\Theta|D, I)$ this may not be ap-

propriate. Naturally a whole continuum of other divergences exist, depending on the exact choice of α, but we do not delve further into their properties here.

The exact process for performing approximate inference therefore consists of the following steps:

1. Restrict our approximating distribution to a family which enables us to perform the necessary algebraic manipulations analytically. Typically this means allowing it to vary according to some parameters ϕ, e.g. $q_\phi(\Theta)$.

2. Alter ϕ in order to find the $q(\Theta)$ within the chosen family which minimises the divergence measure of choice with respect to our true posterior, typically through maximising some equivalent objective function. This step may require gradient descent, or iterative fixed point updates, however is always deterministic cf. MCMC.

Assuming that we do not need to restrict our family of possible posterior distributions too aggressively, approximate inference will yield a continuous posterior, and will typically require only a few tens to hundreds of iterations to converge. This is in contrast to MCMC where it may take several thousand iterations to converge to the stationary distribution, at which point each additional sample contributes only a point estimate to the overall posterior. This means that even if both approaches have identical computational scaling, an approximate inference approach will converge significantly faster in wall-clock time.

For these reasons, in Chapters 3-4 when searching for a maximally computationally efficient inference algorithm we choose to use a variational Bayes scheme, as we determined that the restriction on $q(\Theta)$ was not overly aggressive and we were able to make the required analytic operations tractable.

1.2.2 Stochastic Processes

We now give a brief introduction to stochastic processes, in order to familiarise the reader with some of the terms and concepts relied upon later on. We aim for a formal, but light re-

view, and encourage the interested to pursue further study in Grimmett and Stirzaker (1985); Øksendal (2014).

A stochastic process is a formulation of a probability distribution on an infinite dimensional space. Demonstrating that such an object exists presents several technical challenges, which can be solved by following one of many paths, but here we concentrate on the route which relies on the Kolmogorov extension theorem.

Kolmogorov's Extension Theorem

The Kolmogorov extension theorem Øksendal (2014) relies on a consistency argument with the distribution found when considering a finite dimensional marginal of the full infinite dimensional object. A good explanation of this theorem and how it relates to Gaussian processes (see Section 1.2.4) may be found in G de G Matthews (2016).

Consider an index set Y and a function G which maps this onto the set of real numbers $G : Y \to \mathbb{R}$. Y may be finite dimensional or indeed anything through to uncountably infinite dimensional. Relying on sequence notation, we denote $G_S \coloneqq (G(y)_{y\in S})$ to mean the function evaluated on the set S, where $S \subseteq Y$.

Before stating the extension theorem, (without proof), we also define the a mapping $\pi_{U\to V} : \mathbb{R}^U \to \mathbb{R}^V : (G(y))_{y\in U} \to (G(y))_{y\in V}$.

Theorem: If a family of probability measures μ_V on a Borel set E, labelled by the finite index set V obeys:

$$\mu_U(\pi^{-1}_{U\to V}(E)) = \mu_V(E), \tag{1.15}$$

for all $V \subset Y$ and all finite $U \supset V$, then there is a unique probability measure on the product σ-algebra with the property:

$$\mu_Y(\pi^{-1}_{Y\to V}](E)) = \mu_V(E), \quad \forall E \in \mathscr{B}(\mathbb{R}^V). \tag{1.16}$$

In other words, for a stochastic process to exist, (which necessitates a probability triple on the (possibly infinite dimensional) set Y), it is enough to find a collection of measures that obey the consistency property above on the finite dimensional marginals. If this is the case, than in very general cases a probability triple will exist on the product σ-algebra—i.e. for the full stochastic process.

Later, when we come to look at Gaussian processes, this theorem allows us to fully characterise an infinite dimensional stochastic process via a set of functions which govern the finite dimensional (Gaussian) marginal distributions—namely a mean and covariance function.

1.2.3 Random Point Processes

Random configurations of points in a complete separable metric space—point processes—arise frequently in physical systems, and as a result have been an object of study for many decades in Physics, Mathematics, Statistics and Computer Science. The most fundamental of point processes is the homogeneous Poisson process, which arises as the limiting case of picking points independently and uniformly over a large region.

It is characterised by complete independence of the process when restricted to disjoint subsets of the space. If one constructs a point set in such a manner, the number of points in any subset of the space will be Poisson distributed. For a full derivation of this see Kingman (1993).

Formally, we define S to be a point process state space (for example the real line $\mathbb{R}$), defined on the probability triple $(\Omega, F_\Omega, P_\Omega)$. The point process is then a function $\Pi : \Omega \to S^\infty$, where S^∞ is the set of all countable subsets of S. A given realisation of a point process consists of a 'test set' $A \in S^\infty$, whereby we denote the number of points of Π in A as:

$$N(A) = \#\{\Pi(\omega) \cap A\}, \tag{1.17}$$

which makes $N(A)$ a function $N(A) : \Omega \to \mathbb{N}$, where we require that this function be measurable for all $A \in S^{\infty}$. As this is the definition of a random variable (a measurable function from Ω into some other space), $N(A)$ are random variables, and we may impose conditions on their distributions. The conditions on A such that it is a valid 'test set' are reasonably light, and covered in detail by Kingman (1993).

A *Poisson* point process is formally defined as a point process whereby the counting random variables $N(A)$ obey two properties: the first is that for non overlapping test sets $A_1, ...A_N$, all $N(A_i)$ are statistically independent. The second is that $N(A) \sim \text{Poisson}(\mu)$ where the parameter μ is a non-atomic measure on S for A.

Henceforth, we define 'Poisson process' to be equivalent to 'homogeneous Poisson process'. Furthermore, we will use the words 'point' and 'event' exchangeably, and when the point process is defined on the real line will also include 'arrival' in the set of descriptors.

While this is perhaps one of the most fundamental random processes (alongside the Wiener process), for the purposes of general statistical modelling two different classes of adaptation may be made:

Renewal processes

Let $X(t)$ be a Poisson process defined on $t \in \mathbb{R}^1$ with a rate of λ. For this point process on the real line, the number of points in the interval $(t_0, t_1]$ (denoted $\#(t_0, t_1]$) is Poisson distributed, according to:

$$P(\#(t_0, t_1] = n) = \frac{[\lambda(t_1 - t_0)]^n}{n!} \exp\big(-\lambda(t_1 - t_0)\big). \tag{1.18}$$

Generalisations of this to other complete separable metric spaces involve simply replacing the interval length with a measure on subsets of the space. E.g. if A is a subset on the space of interest, we would replace $(t_1 - t_0)$ with $\mu(A)$ where μ is a reference measure on the space of interest. For simple cases e.g. R^d this may just be the generalised Euclidean volume spanned by A (otherwise known as the Lebesgue measure).

By considering the definition in 1.18 in the special case for which there are 0 points between t_0 and t_1, it becomes clear that the interval between random arrivals is exponentially distributed with parameter λ. As a direct result of the completely random nature of the Poisson process, the interval between arrivals futher exhibits what is known as a 'memoryless' property: given 0 events between t_0 and t_1, and assuming that event locations are strictly ordered, $(t_0 < ... < t_n)$, the distribution over an arrival between t_1 and t_2 conditioned on no arrivals between t_0 and t_1 is still exponential with parameter λ.

There are many applications for which the exponential distribution on arrival times is too restrictive, a classic example being when modelling the event process describing the failure times of a piece of machinery in a factory. In these cases we may define a point process via a specific (non-exponential) renewal or inter-arrival gap density, while retaining the independence and identicality of distribution. Constructing a point process in one dimension in this fashion results in what is termed a homogeneous renewal process. If we in addition break the independence assumption this results in a modulated renewal process (Kingman, 1993).

Correlated point processes

Alternatively, in a rough and application focused taxonomy of point processes by correlation structure, we can introduce either positive or negative correlation between points.

Positive correlation encourages clustering (yielding the inhomogeneous point process as a super-set of examples), while negative correlation yields a variety of repulsive point processes—of which the determinantal point process is one important class of example (Hough et al., 2006). For a visual realisation of this, see Figure 1.1.

In this thesis we are primarily concerned with positive correlation, and as a result look to adapt and extend the inhomogeneous Poisson process:

Without loss of generality, we hereon out consider only point processes defined on $x \in \mathbb{R}^D$. The inhomogeneous Poisson process is a simple probabilistic construction which gives rise to clustered sets of points. We first pick a function $g(x) : \mathbb{R}^D \to \mathbb{R}^1_{\geq 0}$, before sam-

pling a Poisson process whose intensity measure has a density $X(x)$ with respect to $\mu(x)$, where $\mu(x)$ is typically a Lebesgue measure. It is clear that in regions where $g(x)$ is large, more points will be sampled, and so the resulting Poisson process will exhibit clustering behaviour, where the dynamics of the clustering may be adapted through appropriate choice of $g(x)$. This function is typically known as the 'intensity' function of the inhomogeneous Poisson process. Given a set of event locations $E = \{e_0, ..., e_k\}$, the likelihood function for such a model is of the form:

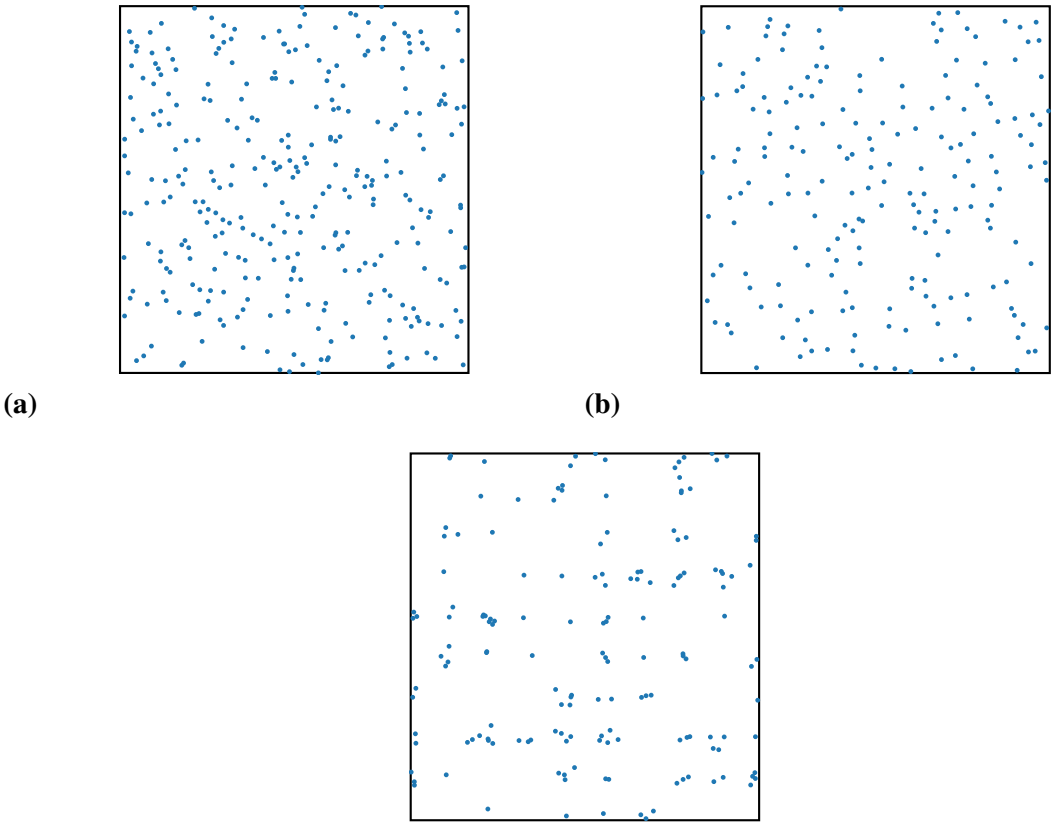
$$P(E|g) = \frac{1}{k!} \exp\left(\int g(x)\,\mathrm{dx}\right) \prod_{n=0}^{k} g(e_n). \tag{1.19}$$

If we further assume that $g(x)$ is a random function, then the resulting construction is known as a Cox process, or a doubly stochastic inhomogeneous Poisson process. If there is reason to believe that our event data is indeed positively correlated, then in practice a Cox process is a sensible (base) model to adopt, for the following reasons:

- The inhomogeneous point process likelihood is in some sense 'weak'. By this we mean that we are required to infer a continuous function ($g(x)$) given a likelihood evaluation which is dependent on an integral observation alongside a finite collection of point values (which must also be inferred). Unless we have very strong prior information about $g(x)$ it is therefore prudent to model it as a random function, in order to capture some of the inherent model uncertainty, and ensure that predictive likelihood estimates are well calibrated.

- The random function is usually specified as a Stochastic Differential Equation (SDE) over the domain of interest, possibly directly (Wei et al., 2002). This allows us to include a flexible range of expected behaviour, which may be adapted through inference from the data, whilst still acknowledging a degree of uncertainty.

Fig. 1.1 Sets of random points on a subset of the 2D plane sampled from: A homogeneous Poisson, (**a**), (no correlation between points); a determinantal point process, (**b**), (negative correlation between points); and an inhomogeneous Poisson process, (**c**), (positive correlation between points).



(a) **(b)** **(c)**

There are a variety of other interesting model adaptations (e.g. Hawkes processes (Hawkes, 1971), birth death processes (Latouche and Ramaswami, 1999)), as well as deep theoretical links (Stochastic geometry (Stoyan et al., 1987), completely random measures (Kingman, 1993)), to name but a few, but we will introduce these if and when required, so as to avoid cluttering this general introduction.

Bayesian inference for Cox processes

Whilst the authors are in any case a strong proponent of information efficiency, and therefore doing full Bayesian inference wherever it is possible, we feel that in the case of a Cox process model this is doubly important: Firstly because the vast majority of Cox process

data is by very nature 'sparse', in other words even if one ostensibly has 'many' observations in one dimension (or higher), the size of the space over which we must infer a function is exponentially larger, and theoretically continuous (bearing in mind that the number of function values which must also be inferred grows exactly linearly with the size of the data). We might equally argue that many Cox process problems are equivalent to traditional 'small data' problems due to the nature of the likelihood, and therefore it is important to both incorporate as much prior knowledge as possible, and derive well calibrated posterior uncertainty estimates. Secondly, throughout this thesis we construct joint models of related event processes. By viewing such a setting through the prism of conditional probability, we may efficiently share relevant information between related datasets, and the problem becomes one for which Bayesian inference is highly appropriate.

1.2.4 Gaussian processes

The GP is a fundamental Bayesian non-parametric statistical object. It is an infinite collection of random variables, where the joint distribution over any finite subset is Gaussian. Following the notation from the previous section, it is fully characterised by a mean function ($m(x) : \mathbb{R}^D \to \mathbb{R}^1$) and a covariance function—sometimes referred to as a kernel function ($k(x, x') : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^1$), and in fact can usually be described using only a covariance function (while setting $m(x) = 0$—as marginalising over an unknown mean function can be equivalently expressed as a new zero mean GP with a different covariance function). We define $g \sim \mathcal{GP}(m(x), k(x, x'))$ to mean that $g(x)$ is Gaussian process distributed, which then implies that for a collection $E = \{x_0, ..., x_n\}$, $g(E)$ will be an n dimensional Gaussian, derived through evaluating the mean and covariance functions.

Covariance functions

In our work, we use the GP as a prior over functions. We demonstrate this use case for the one dimensional regression problem in Figure 1.2. In this case the covariance function encodes

our beliefs about the structure of the function of interest. A good exploration of how to embed different types of structure can be found in Duvenaud (2014). The main restriction on covariance function design (in the case of GP models) is that it must be positive semidefinite, this is the case if and only if:

$$\int k(x, x')f(x)f(x')\,\mathrm{dx}\,\mathrm{dx}' \geq 0, \tag{1.20}$$

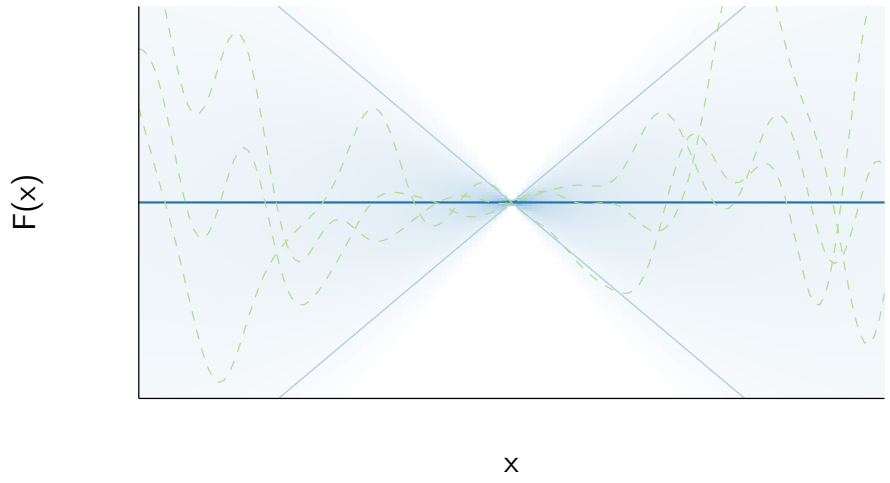
for all $f \in L_2(x, \mu)$—i.e. for all square integrable functions under μ. This is equivalent to guaranteeing that the covariance matrix arrived at by evaluating $k(x, x')$ at any collection E of points is positive semidefinite. Translation invariant covariances result in strongly stationary Gaussian process *priors*, whilst a range of interesting approaches exist to generate non-stationary priors (Paciorek and Schervish, 2004; Rasmussen and Williams, 2006).
Positive semidefinite covariances will have a spectral (Fourier domain) representation via Bochner's theorem (Rudin, 1987). Such a representation may be leveraged to translate between a GP and a Stochastic Partial Differential Equation (SPDE) representation of the underlying dynamics. Depending on a variety of factors this may prove to be a more efficient domain in which to perform inference via e.g. (in the simplest one dimensional stationary case) a Rauch-Tung-Striebel smoother (Sarkka, 2013). It also allows us to make the link between the traditional approach of using a SDE to represent $g(x)$ in the point process literature, and the machine learning equivalent of using a GP.

Useful properties

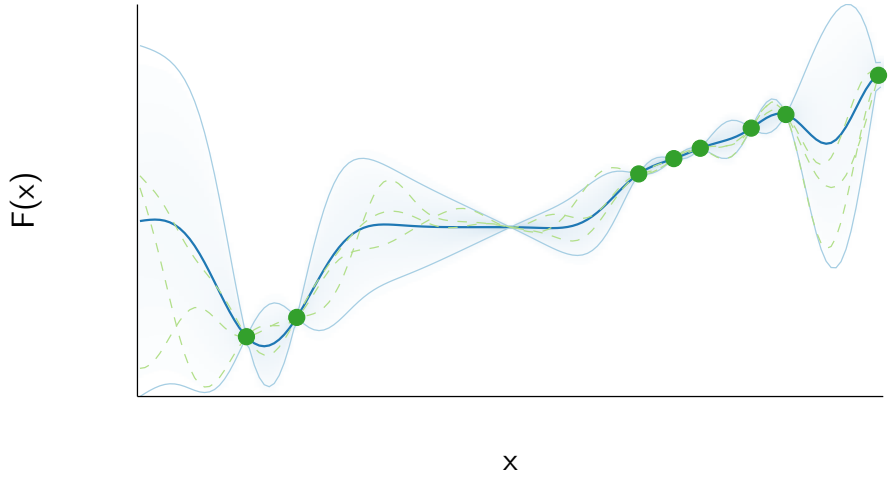
The Gaussian process shares many traits with the Gaussian. These (and others) make the GP highly useful as a prior for function modelling. A few key examples are:

- *Expressivity through kernel selection:* If we wish to perform accurate modelling of $g(x)$, it is important to minimise model mis-specification, by placing as much prior mass as is reasonable on the right area of function space. A variety of prior knowl-

Fig. 1.2 Draws from a one dimensional GP prior assuming $m(x) = 0$, and that $k(x, x') = xx'^T \times \cos(x - x') \times \exp(-\frac{|x-x'|^2}{2})$, (1.2a), and the resulting process after conditioning on eight observations (the green dots) (1.2b). The light blue shading indicates the $\pm 2\sigma$ (two standard deviation) credible interval. This covariance is the product of a linear kernel with a single translation invariant Gabor wavelet. It encodes locally periodic smooth functions, with a linearly increasing covariance.



(a)



(b)

edge may be embedded through judicious choice of covariance function, as fortunately the set of positive semi definite covariance functions is closed under multiplication and addition, allowing for unlimited potential variability. Work exists towards covariance discovery, spanning the gap through to what might be termed 'empirical Bayes' for covariance function selection (Lloyd et al., 2014), however we would suggest that at the time of writing this, the GP is best employed when strong, encodable, prior knowledge exists; or alternatively where well calibrated posterior uncertainty estimates are important.

- *Analytic likelihood and associated manipulations:* As in the case of a finite dimensional Gaussian distribution, conditioning, marginalisation and inference operations are often both closed form and analytic. This allows for integration over a range of hypotheses, as well as posterior prediction for missing values in x.

- *Closure under linear maps:* As in the case of the Gaussian, the GP retains distributional properties under linear operations. This extends from multiplication or addition by a scalar, through to integration and differentiation. We make extensive use of this fact in Chapter 5, but are by no means the first to benefit (Osborne et al., 2009; Rasmussen and Ghahramani, 2003).

Naturally the GP does not come without limitations. The two which we will encounter most frequently are:

- If we have n observations, naïve GP inference scales computationally as $\mathcal{O}(n^3)$. Fortunately many techniques exist to improve upon this worst-case upper bound, e.g. Snelson and Ghahramani (2005).

- Many useful likelihoods (where by likelihood we mean the link function between the GP and the observation set) make inference non analytic. We shall see this happen

time and time again throughout the thesis. Fortunately there are a variety of approximate inference techniques which allow us to make the intractable inexact, but tractable (Lloyd* et al., 2015).

1.3 Outline of Thesis Structure

In the remainder of this Thesis, we present and demonstrate contributions as outlined in Section 1.1:

- In Chapter 2, we provide a framework for modelling structured Cox processes, alongside an efficient MCMC inference scheme.
- In Chapter 3, we develop the first variational inference scheme for Gaussian-process-driven Cox processes, without resorting to thinning or discretisation. The scheme provides vast improvements in computational scaling as cf. existing approaches.
- In Chapter 4, the variational inference scheme developed in Chapter 3 is applied to both a variant of the structured Cox process problem presented in Chapter 2, as well as more abstract point process modelling problems.
- In Chapter 5, we deviate from the theme of inference for point processes, and instead develop a computationally efficient Bayesian algorithm for computing model evidence.
- Finally, in Chapter 6 we conclude by summarising related work, pursued in parallel to our own, while also suggesting related and potentially fruitful avenues for future exploration.

Chapter 2

Structured Point Processes

The material in this chapter is based on the following paper:
T. Gunter*, C. Lloyd*, M. A. Osborne, and S. J. Roberts. Efficient Bayesian Nonparametric Modelling of Structured Point Processes. In *Uncertainty in Artificial Intelligence (UAI)*, 2014,
where our contributions were as outlined in the acknowledgements section of this thesis.

In this chapter we build upon work done in Adams et al. (2009) in order to derive a flexible Bayesian generative model for dependent Cox processes, alongside an associated efficient MCMC scheme for inference. This scheme is an independent contribution, and one which we would use if implementing Adams et al. (2009) today, particularly if the data of interest exists in a space of dimension 2 or higher.

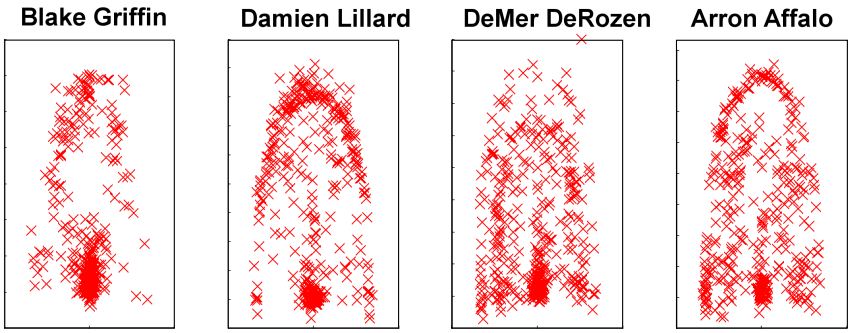
2.1 Introduction

Point processes are effectively used to model a variety of event data, and have also shown a recent popularity within the Machine Learning community as priors over sets. As stated in the introduction, the most fundamental example of such a stochastic model for random sets is the homogeneous Poisson process. This is defined via an intensity which describes the expected number of points found in any bounded region of some arbitrary domain. An

inhomogeneous Poisson process allows the intensity to vary throughout the domain over which the process is defined. As we do not know the functional form of this intensity given only event data, another stochastic process is typically used to model it nonparametrically. This is then termed a doubly-stochastic Poisson process, more commonly known as a Cox process. In our particular construction, we use transformed Gaussian processes to model the intensity functions of the individual dependent point processes, in such a manner as to enable fully nonparametric Bayesian inference (Adams et al., 2009). While we only explicitly consider the doubly stochastic Poisson process, any general renewal process (Rao and Teh, 2011) could be incorporated into the framework we define.

There are many occasions when we have multiple point processes which we expect to be dependent: If the domain is temporal, then an example would be individual clients making trades with a specific financial services provider, or individual customers purchasing items from a specific vendor. If the domain is spatial, we might consider different categories of crime defined over some geographic region. Or, in a more trivial (but perhaps easy to visually interpret) example, we might consider modelling the rate processes which describe the locations of point-shots taken by a group of NBA basketball plyers; Blake Griffin, Damien Lillard, DeMer DeRozen and Arron Affalo:

Fig. 2.1 Basketball shot data: the points on the court where each named player took shots in an attempt to score during the $2013 - 2014$ NBA season.



The four player shot profiles depicted in Figure 2.1 share many characteristics, for example a number of shots are clearly concentrated around the hoop and the three-pointer line. The players we selected all in fact perform subtly different roles, however, and this too is evident in the data: for example Damien Lillard primarily shoots from the hoop and the three-pointer line. It is clear that these four shot profiles contain some shared structure, not just visually, but also through knowledge of the 'generative' process by which this data set was created!

In this Chapter, we seek to build a generative model for just such a collection of dependent streams of point process data. Defining this flexible model for inter-process dependency structure, alongside an efficient inference scheme allows us to learn the underlying intensity functions which drive the typical behaviour. These can then be used to make more accurate predictions, especially during periods of unobservability for an individual process.

In order to maximise the flexibility of our approach, we specifically assume that the individual intensity functions arise via a weighted summation of convolutions of latent functions with a kernel. Intuitively this means that we take a small number of latent functions, individually smooth and scale them, and then add them together to yield an intensity function. This approach allows a wide range of intensities to arise from only a few latent functions, and

We will present and validate the following novel contributions:

- The first generative model for dependent Cox process data without discretisation of the domain (Section 2.2).
- An efficient, parallelised inference scheme, which scales benignly with the number of observed point processes (Section 2.3).
- A new adaptation of thinning (Lewis, 1979), which we term 'adaptive thinning'. This introduces multiple uniformisation levels over the space, making the model viable for higher dimensional spaces and larger datasets (Section 2.5.5).

2.2 The Model

We first briefly review the Cox process, before describing the innovative nonparametric Bayesian model outlined in (Adams et al., 2009) as the Sigmoidal Gaussian Cox Process (SGCP), which allows a full Gaussian process to be used as a prior over an individual intensity function. We then move on to review the convolution process (Álvarez and Lawrence, 2011), a method of modelling dependent functions and the underlying latent processes which govern them. Our novel combination of these constituent elements represents the first model for dependent Cox point processes without resorting to discretisation of the domain.

2.2.1 The Inhomogeneous Poisson Process

For a domain $\mathcal{X} = \mathbb{R}^D$ of arbitrary dimension D, we may define an inhomogeneous Poisson process via an intensity function $\lambda(x) : \mathcal{X} \to \mathbb{R}^+$, and a Lebesgue measure over the domain, $\mathrm{d}x$. The number of events $N(\mathcal{T})$ found over a subregion $\mathcal{T} \subset \mathcal{X}$ will be Poisson distributed with parameter $\lambda_{\mathcal{T}} = \int_{\mathcal{T}} \lambda(x)\,\mathrm{d}x$. Furthermore, we define $N(\mathcal{T}_i)$ to be independent random variables, where $\mathcal{T}_i$ are disjoint subsets of $\mathcal{X}$ (Kingman, 1993).

If we bound the region to be considered, and assume there are K observed events, labelled as $\{x_k\}_{k=1}^K$, then the inhomogeneous Poisson process likelihood function may be written as

$$p(\{x_k\}_{k=1}^K \mid \lambda(x)) = \exp\left\{-\int_{\mathcal{T}} \mathrm{d}x\, \lambda(x)\right\} \prod_{k=1}^{K} \lambda(x_k). \tag{2.1}$$

2.2.2 The Sigmoidal Gaussian Cox Process

In order to model the intensity nonparametrically, we place a Gaussian process (Rasmussen and Williams, 2006) prior over a random scalar function $g(x) : \mathcal{X} \to \mathbb{R}$. This is defined by a positive definite covariance function $k(.,.) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and a mean function $m(.) :$

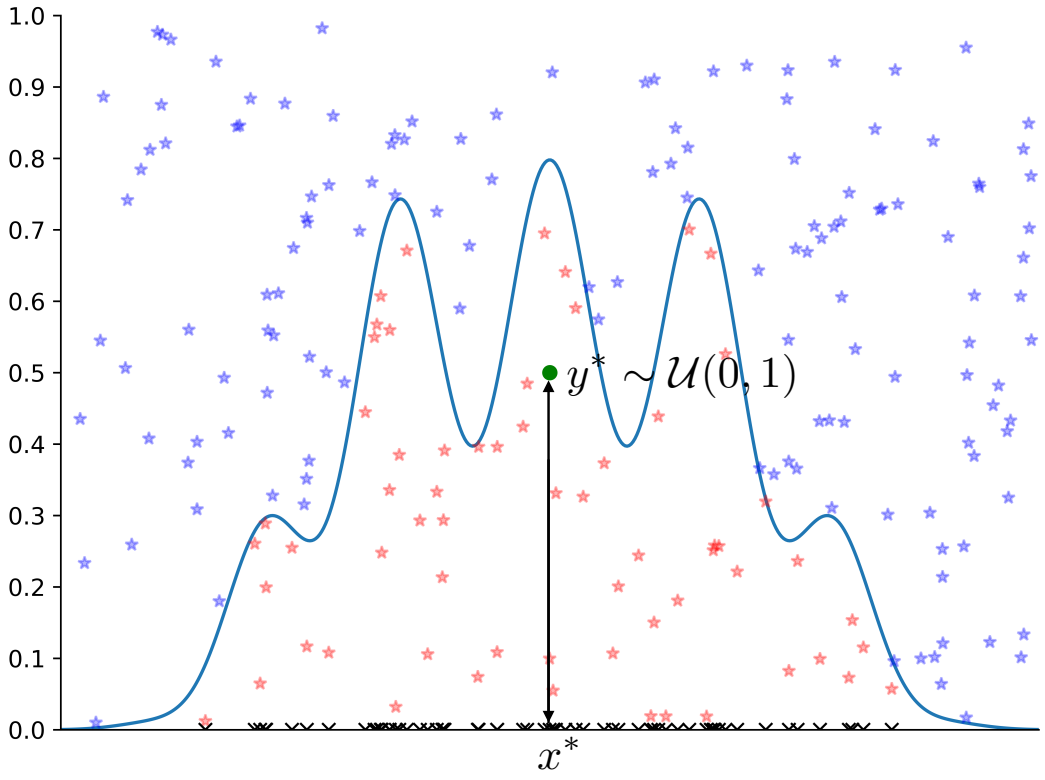
$\mathcal{X} \to \mathbb{R}$. The mean and covariance function are parameterised by a set of hyperparameters, which we denote γ.

In the SGCP, a Gaussian process is transformed into a prior over the intensity function by passing it through a sigmoid function and scaling it against a maximum intensity λ^*: $\lambda(x) = \lambda^* \sigma(g(x))$, where $\sigma(.)$ is the logistic function. This forms the basis of a generative prior, whereby exact Poisson data can be generated from $\lambda(x)$ via thinning (Lewis, 1979), which involves adding M events, such that the joint point process over the $M + K$ events is homogeneous with fixed rate λ^*.

The generative process is illustrated in Figure 2.2. We begin by sampling a value for λ^* from a Gamma prior. We then draw a sample function from a GP, before passing it through the sigmoid function and then scaling it by λ^* to give us $\lambda(x)$, shown as the blue line. We next select a uniform distribution, scaled so as to cover the (bounded) domain and upper-bound $\lambda(x)$ at λ^*. We then sample a point-count N from a Poisson distribution parameterised as $\mathcal{P}(\lambda^* \int_{\mathcal{T}})$, before simulating N points from the upper-bounding uniform. We then reject (or 'thin') samples if they fall *above* $\lambda(x^*)$ for a given x^*, and keep them as data otherwise. In this manner we have generated inhomogeneous Poisson process data by thinning (rejecting) a subset of the points generated by the homogenous Poisson generative process. The thinning process is analagous to the method of 'rejection sampling' Neal (2010) for simulation, whereby a hard-to-simulate density is strictly upper bounded by an easy to sample from density, $t(x)$. We then sample from $t(x)$, and for each proposed point, (e.g. x^* in Figure 2.2), we simulate from a one dimensional uniform $\mathcal{U}(0, t(x^*))$, accepting the sample if it lies below the density of interest, $\lambda(x)$, otherwise rejecting it.

As we are using an infinite dimensional proxy for $\lambda(x)$, the integral in Equation 2.1 is intractable. Furthermore, using Bayes' theorem with this likelihood yields a posterior with intractable integrals in both the numerator and denominator. These challenges are overcome by making use of the generative prior, and augmenting the variable set to include the number

Fig. 2.2 Graphical representation of thinning (and hence the generative process for the SGCP): Blue stars indicate thinned points, red stars represent accepted data points (also shown in the rug-plot along the x-axis). The blue line shows the intensity function (equivalent to the density we wish to simulate from in the rejection sampling analogy mentioned in 2.2.1).



of thinned points, M, and their locations, $\{\tilde{x}_m\}_{m=1}^M$. This then means that the value of the intensity function need only be inferred at the $M+K$ point locations, $\mathbf{g_{M+K}} = \{g(x_k)\}_{k=1}^K \cup \{g(\tilde{x}_m)\}_{m=1}^M$. Noting that $\sigma(-z) = 1-\sigma(z)$, the joint likelihood over the data, function values and latent variables is

$$\begin{aligned} p(\{x_k\}_{k=1}^K, M, \{\tilde{x}_m\}_{m=1}^M, \mathbf{g_{M+K}} \mid \lambda^*, \mathscr{T}, \theta) =& (\lambda^*)^{M+K} \exp\left\{-\lambda^* \mu(\mathscr{T})\right\} \\ & \times \prod_{k=1}^K \sigma(g(x_k)) \prod_{m=1}^M \sigma(-g(\tilde{x}_m)) \\ & \times \mathscr{GP}(\mathbf{g_{M+K}} \mid \{x_k\}_{k=1}^K, \{\tilde{x}_m\}_{m=1}^M, \gamma), \quad (2.2) \end{aligned}$$

where we have defined $\mu(\mathscr{T}) = \int_{\mathscr{T}} \mathrm{d}x$.

Notably, this likelihood equation does not involve any intractable integrals. This means that inference is now possible in this model, albeit subject to the cost of an augmented variable set.

2.2.3 The Convolution Process

The convolution process framework is an elegant way of constructing dependent output processes. Instead of assuming the typical instantaneous (Teh et al., 2005) mixing of a set of independent processes to construct correlated output processes, we generalise to allow a blurring of the latent functions achieved via convolution with a kernel, $G(x, z)$, prior to mixing. This gives us more flexibility in the overall link function between the latent structure and observed rate functions, and enables us to model data whereby the generative process implies a scaling and output specific diffusion of the latent rate function. For example, in the basketball example presented earlier, it is possible that a single latent function could be smoothed and scaled in such a way as to create each of the different player shot profiles—if we relied on a simple instantaneous mixing, we would require several more latent functions before our model had the same expressivity.

z is typically defined on the same domain as x. If we place a Gaussian process prior over the latent function, the output function turns out to also be a Gaussian process (Álvarez and Lawrence, 2011). Specifically, given D dependent intensity functions $g_d(x)$ and Q latent processes $u_q(x)$, (where typically $Q < D$), the stochastic component of the dth intensity is

$$g_d(x) = \sum_{q=1}^{Q} \int_{\mathscr{T}} G_d(x, z) u_q(z) \, \mathrm{d}z. \tag{2.3}$$

Given full knowledge of the latent functions, the $g_d(x)$ are independent and deterministic. The $G_d(x, z)$ encode the observed process specific characteristics, and the $u_q(z)$ can be thought of as encoding the latent driving forces.

The convolution process has strong links with the Bayesian kernel method, as described in (Pillai et al., 2007). This allows a function $f(x)$ on $\mathscr{X}$ to arise as

$$f(x) = \int_{\mathscr{X}} K(x, z) U(\mathrm{d}z), \tag{2.4}$$

where $U(\mathrm{d}z) \in \mathscr{M}(\mathscr{X})$ is a signed measure on $\mathscr{X}$. The integral operator $\mathscr{L}_K : U(\mathrm{d}z) \to f(x)$ maps the space of signed measures $\mathscr{M}(\mathscr{X})$ into $\mathscr{H}_K$, a Reproducing Kernel Hilbert Space (RKHS) defined by the kernel, $K(x, z)$. This mapping is dense in $\mathscr{H}_K$.

The convolution process is also known as a latent force model (Álvarez et al., 2009). In this guise, it is used to infer the solution of a differential equation when there is uncertainty in the forcing function. The convolution kernel is the Green's function of a particular differential equation, and the Gaussian process prior is placed on the driving function. This representation lets us consider the latent functions as driving forces, which are viewed through the intensity function specific convolution kernel. The convolution kernel can, for example, be used to model differing speeds of information propagation from the latent factors to each of the observed processes.

2.2.4 Sparse Latent Functions

To ensure tractability of inference, we make use of the property that the intensities are independent conditioned on the latent functions. This is made clear from the perspective of a generative model with only one latent function: we first draw a sample of the object $u(z)$, before solving the integral in equation 2.3, where uncertainty about $u(z)$ is propagated through the convolution. Now instead of maintaining the full, infinite dimensional object $u(z)$, let us condition on a finite dimensional draw of $u(z)$, $u(Z) = [u(z_1), \dots, u(z_J)]^T$ where $Z = \{z_j\}_{j=1}^J$. We can then sample from $p(u(z) \mid u(Z))$, as this is a conditional Gaussian distribution, and use this function to solve the convolution integral. With multiple latent functions we can approximate each $u_q(z)$ by $\mathbb{E}[u_q(z) \mid u_q(Z)]$, replacing Equation 2.3 with

$$g_d(x) \approx \sum_{q=1}^{Q} \int_{\mathcal{T}} G_d(x, z)\mathbb{E}[u_q(z) \mid u_q(Z)]\,\mathrm{d}z. \tag{2.5}$$

This is reasonable as long as each $u_q(z)$ is smooth, in the sense that it is well approximated given the covariance function and the finite dimensional sample $u_q(Z)$. In Section 2.3, we use the approximation in Equation 2.5 along with the conditional independence assumption to build a tractable inference scheme.

2.2.5 Constructing the Model

Let the Q latent functions $u_q(z)$ be modelled as Gaussian processes with Gaussian covariance functions such that

$$u_q \mid \phi_q \sim \mathcal{GP}(0, K_q(z, z')), \tag{2.6}$$

where $K_q(z, z')$ is simply the Gaussian kernel

$$K_q(z, z') = \mathcal{N}(z; z', \phi_q). \tag{2.7}$$

We use a scaled Gaussian convolution kernel

$$G_{d,q}(x,z) = \kappa_d \mathcal{N}(x; z, \theta_{d,q}). \tag{2.8}$$

This restricts $g_d(x)$ to be at least as smooth as the random draws from $u_q(z)$. The covariance linking $u_q(z)$ to $g_d(x)$ is

$$\begin{aligned} K_{g_d,u_q}(x,z) &= \int_{\mathcal{X}} G_{d,q}(x,z) K_q(z,z') \mathrm{d}z \\ &= \kappa_{d,q} \mathcal{N}(x; z, \theta_d + \phi_q), \end{aligned} \tag{2.9}$$

and the overall covariance between output functions is

$$\begin{aligned} K_{g_d,g_{d'}}(x,x') &= \sum_{q=1}^{Q} \int_{\mathcal{X}} G_d(x,z) \int_{\mathcal{X}} G_{d'}(x',z') K_q(z,z') \, \mathrm{d}z' \, \mathrm{d}z \\ K_{g_d,g_{d'}}(x,x') &= \sum_{q=1}^{Q} \kappa_{d,q} \kappa_{d',q} \mathcal{N}(x; z, \theta_d + \theta'_d + \phi_q). \end{aligned} \tag{2.10}$$

We could use this structured covariance function to construct one large joint Gaussian process over all the intensity functions. In doing so, however, the $u_q(z)$ have been implicitly integrated out, and the resulting inference problem will scale computationally as $\mathcal{O}(D^3N^3)$ with storage requirements of $\mathcal{O}(D^2N^2)$, where $N = M + K$ is the joint number of events. This is intractable for any real problem, where we would hope to leverage many dependent point processes to learn a few latent factors with minimal uncertainty.

Let us now define some additional notation: $\mathbf{K}$ denotes a covariance matrix obtained by evaluating the appropriate covariance function at all eligible pairs of data points. Subscripts determine which covariance is used and hence which inputs are valid, e.g. $\mathbf{K}_{g_d,u_q}$ denotes the cross covariance between the dth output and qth input function. $\mathbf{K}_{g_d,u}$ means stack the

Q $\mathbf{K}_{g_d,d_q}$ matrices vertically, $\mathbf{K}_{u,u}$ is a block diagonal matrix where each block corresponds to $\mathbf{K}_{u_q,u_q}$, and $\mathbf{u}$ is the result of stacking the draws from the finite dimensional Gaussians $p(u_q(Z))$ vertically.

We also define: $\phi = \{\phi_q\}_{q=1}^{Q}$, $\kappa = \{\kappa_d\}_{d=1}^{D}$, $\theta = \{\theta_d\}_{d=1}^{D}$, $X_d = \{x_{d,k}\}_{k=1}^{K_d} \cup \{\tilde{x}_{d,m}\}_{m=1}^{M_d}$. If we wish to allow the latent functions to be sampled at different points, then we define a separate Z_q for each $u_q(z)$: $Z_q = \{z_{q,j}\}_{j=1}^{J}$. The set of inputs over all latent functions is then $Z = \{Z_q\}_{q=1}^{Q}$, and similarly for the intensities: $X = \{X_d\}_{d=1}^{D}$.

Notation in place, we determine that given the approximation in Equation 2.5, the conditional likelihood for $g_d(x)$ is

$$p(g_d \mid u, Z, X_d, \kappa_d, \theta_d, \phi) = \mathcal{N}(\mathbf{K}_{g_d,u}\mathbf{K}_{u,u}^{-1}\mathbf{u}, \mathbf{K}_{g_d,g_d} - \mathbf{K}_{g_d,u}\mathbf{K}_{u,u}^{-1}\mathbf{K}_{g_d,u}^{T}). \tag{2.11}$$

Still conditioning on the latent functions, the joint likelihood over all D intensity functions is then simply

$$p(g_1, \ldots, g_D \mid u, Z, X, \kappa, \theta, \phi) = \prod_{d=1}^{D} p(g_d \mid u, Z, X_d, \kappa_d, \theta_d, \phi). \tag{2.12}$$

Bayes' rule for Gaussians gives us the posterior over the $u_q(Z)$ as

$$p(u_1, \ldots, u_Q \mid g_1, \ldots, g_D, Z, X, \kappa, \phi, \theta) = \mathcal{N}(u_1, \ldots, u_Q; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p). \tag{2.13}$$

Where the mean and covariance are

$$\begin{aligned} \boldsymbol{\Sigma_p} &= \left[\mathbf{K}_{u,u}^{-1} + (\mathbf{K}_{g,u}\mathbf{K}_{u,u}^{-1})^T\mathbf{D}^{-1}(\mathbf{K}_{g,u}\mathbf{K}_{u,u}^{-1})\right]^{-1} \\ \boldsymbol{\mu_p} &= \boldsymbol{\Sigma}_p(\mathbf{K}_{g,u}\mathbf{K}_{u,u}^{-1})^T\mathbf{D}^{-1}\mathbf{g} \end{aligned} \tag{2.14}$$

and where $\mathbf{D} = \mathbf{K}_{g,g} - \mathbf{K}_{g,u}\mathbf{K}_{u,u}^{-1}\mathbf{K}_{g,u}^{T}$.

The exact form of $\mathbf{D}$ depends on the degree to which we are willing make independence assumptions in order to approximate the Gaussian processes used to model the functions. Naturally the higher the degree of approximation, the more scalable the resulting inference scheme.

Under full dependence, a single event is linked to both inter and intra-process data. We will be assuming that the dependency structure across the $g_d(x)$ is entirely contained by the latent processes, $u_q(z)$. Intuitively, this means that we maintain the full Gaussian process structure for each individual intensity function, while summarising the latent functions via a set of inducing inputs $Z = \{z_j\}_{j=1}^{J}$. This approximation scheme results in a functional form which is similar to what Quiñonero Candela and Rasmussen (2005) call the Partially Independent Training Conditional (PITC) scheme. Importantly it allows inference to scale computationally as $\mathcal{O}(DN^3)$, with storage requirements of $\mathcal{O}(DN^2)$ even in the worst case scenario of $J = N$. Further approximations may be made, and these are especially useful if the number of events per process is large, however for our purposes they are not necessary. For more information on approximation methods for Gaussian processes see Quiñonero Candela and Rasmussen (2005) and Snelson and Ghahramani (2005).

Under the PITC low rank covariance, the resulting form for $\mathbf{D}$ is: $[\mathbf{K}_{g,g} - \mathbf{K}_{g,u}\mathbf{K}_{u,u}^{-1}\mathbf{K}_{g,u}^{T}] \circ \mathbf{M}$, where $\mathbf{M} = \mathbf{I}_N \otimes \mathbf{1}_N$ and $\mathbf{1}_N$ is a $N \times N$ matrix of ones. This may be more familiar as `blkdiag`[$\mathbf{D}$].

2.3 Inference

For each of the D point processes we need to learn $|X_d|$, X_d, κ_d, θ_d, λ_d^* and $g_d(x)$. For each of the Q latent functions $u_q(Z)$ and ϕ_q must be inferred. Z_q are fixed to an evenly spaced grid which is identical across the latent processes. To find posteriors over all these variables, we choose a MCMC algorithm, as detailed below.

Using the PITC approximation scheme, the likelihood over the point processes factorises conditioned on the latent functions. This means that given D compute units the updates associated with each point process may be made in parallel. This is important as the inference algorithm is computationally bottlenecked by operations associated with learning the locations of the thinning points, X.

We now give a recap of the inference scheme from the SGCP for a single point process, while listing our minor modifications. Updates for the latent functions are then given, conditioning on the D intensity functions.

2.3.1 Learning the Intensity Function

Recalling Equation 2.2, three kinds of Markov transitions are used to draw from this joint distribution: 1) Sampling the number of thinned points, M. 2) Sampling the locations of the thinned events, $\{\tilde{x}_m\}_{m=1}^{M}$. 3) Resampling the intensity function, $\mathbf{g_{M+K}}$.

Metropolis-Hastings is used to sample M. The probability of insertion/deletion is parameterised by a Bernoulli proposal function: $b(K, M) : \mathbb{N} \times \mathbb{N} \to (0, 1)$, where the parameter has been arbitrarily set to $\frac{1}{2}$. If an insertion is required, a new x_{M+1} is drawn uniformly and at random from $\mu(\mathscr{T})$, and $g(x_{M+1})$ is drawn from the Gaussian process conditioned on the current state. A deletion results in a thinned event $\tilde{x}_m$ being removed at random from $\{\tilde{x}_m\}_{m=1}^{M}$. The overall transition kernels q, and Metropolis-Hastings acceptance ratios, a, are:

$$q_{ins}(M+1 \leftarrow M) = \frac{b(K,M)}{\mu(\mathscr{T})}\mathscr{G}\mathscr{P}(g(\tilde{x}_{M+1}) \mid \{\tilde{x}_m\}_{m=1}^{M}, \mathbf{g_{M+K}}),$$

$$a_{ins} = \frac{(1 - b(K, M+1))\mu(\mathscr{T})\lambda^*}{(M+1)b(K, M)(1 + \exp(g(\tilde{x}_{M+1})))}, \tag{2.15}$$

$$q_{del}(M-1 \leftarrow M) = \frac{1 - b(K, M)}{M} \tag{2.16}$$

$$a_{del} = \frac{M b(K, M-1)(1 + \exp(g(\tilde{x}_m)))}{(1 - b(K, M))\mu(\mathscr{T})\lambda^*}. \tag{2.17}$$

Sampling the locations of the thinned events also makes use of the Metropolis criterion. For each event $\tilde{x}_m$ a move to $\hat{x}_m$ is proposed via a Gaussian proposal density. A function value $g(\hat{x}_m)$ is then drawn conditioned on the state with $g(\tilde{x}_m)$ removed, denoted $\mathbf{g}_{\mathbf{M}\_+\mathbf{K}}$. This gives the move acceptance ratio

$$a_{move} = \frac{q_{move}(\tilde{x}_m \leftarrow \hat{x}_m)(1 + \exp(g(\tilde{x}_m)))}{q_{move}(\hat{x}_m \leftarrow \tilde{x}_m)(1 + \exp(g(\hat{x}_m)))}. \tag{2.18}$$

where q_{move} is the proposal distribution. We use a symmetric Gaussian proposal

$$q_{move}(\hat{x}_m \leftarrow \tilde{x}_m) = \mathcal{N}\left(0, \frac{\mu(\mathcal{T})}{100}\right). \tag{2.19}$$

To sample the function we opt to use the elliptical slice sampling approach developed in Murray et al. (2010). This is an algorithm specifically designed for sampling from high dimensional, highly correlated, Gaussian process posteriors. The log conditional posterior over function values is

$$\begin{aligned}\log\, p(\mathbf{g}_{\mathbf{M}+\mathbf{K}} \mid M, \{x_k\}_{k=1}^K, \{\tilde{x}_m\}_{m=1}^M, \gamma) = &-\frac{1}{2}\mathbf{g}_{\mathbf{M}+\mathbf{K}}\square^{-\mathbf{1}}\mathbf{g}_{\mathbf{M}+\mathbf{K}} - \sum_{k=1}^{K}\log(1+\exp(-g(x_k))) \\ &- \sum_{m=1}^{M}\log(1+\exp(g(\tilde{x}_m))) + const.\end{aligned} \tag{2.20}$$

In our case, $\square$ is equal to the covariance in Equation 2.11, and naturally for each iteration we perform all the above updates in parallel for each observed point process, conditioned on the latent functions.

To infer the posteriors over the Gaussian process hyperparameters, we use Hamiltonian Monte Carlo (HMC) (Duane et al., 1987; Neal, 2010), with log-normal priors over each hyperparameter. By placing a Gamma prior with shape α and inverse scale β over λ^*, we infer

Algorithm 1 MCMC Scheme

Input: $\{X_k\}_{k=1}^K$, priors.
repeat
  ParFor $d = 1$ **to** D
    Sample thinned events: Equations 2.15 → 2.17
    Sample locations: Equations 2.18 → 2.19
    Sample function: Equation 2.20
    Sample hyperparameters: Equation 2.20
    Sample λ^*: Equation 2.21
  EndParFor
  Sample latent functions: Equation 2.13
  Sample latent hyperparameter: Equation 2.22
until *convergence* is *true*

the posterior conditioned on the thinned and true points using a Gibbs update as follows:

$$\alpha_{post} = \alpha + K + M, \qquad \beta_{post} = \beta + \mu\mathcal{T}. \tag{2.21}$$

2.3.2 Learning the Latent Functions

Conditioning on the point process intensity functions, $g_d(x)$, the latent functions are dependent, with conditional posterior distribution given by Equation 2.13.

Having drawn new values for each of the $u_q(Z_q)$, we can update the ϕ_q using a Metropolis-Hastings step under the following log conditional posterior which is

$$\log p(\phi_q \mid u_q(Z_q), Z_q) = -\tfrac{1}{2}u_q(Z_q)\mathbf{K}^{-1}_{u_q,u_q}\tfrac{1}{2}u_q(Z_q) - \tfrac{1}{2}\log\det(\mathbf{K}_{u_q,u_q}) + \text{const.} \tag{2.22}$$

The overall procedure is summarised in Algorithm 1.

2.4 Adaptive Thinning

In higher dimensional spaces, data is typically concentrated into small, high density subdomains. Under the current methodology, we must thin the entire empty space to a uniform concentration which matches that of the most dense subregion. If we wish to use Gaussian

process intensities this rapidly becomes infeasible, even under the most radical of sparse approximations (Snelson and Ghahramani, 2005).

Our novel solution to this problem is to model the upper bounded intensity over the space using a piece-wise constant function, where each section takes a fractional proportion of the global upper bound, λ^*. This preserves the tractability of the integrals in the likelihood and posterior, and does not violate any of the properties of the point process, while simultaneously allowing empty regions to be thinned to a far lower average density.

Consider Figure 2.3: this shows both the data and the thinned points, where for the left three quarters of the plot the maximum rate does not exceed 50% of λ^*. Let us assume we allow the maximum rate to take one of two values for each datapoint: $\frac{1}{2}\lambda^*$ and λ^*. For each new thinned point we sample an intensity function value, before also sampling an upper bound for the rate from the available levels. This upper bound is at least as great as the current function evaluation at that point.

In this manner we hope to infer that for a portion of the domain in Figure 2.3, the rate may be happily upper-bounded by half the global maximum rate, λ^*, and hence the bulk of the space may be thinned to a significantly lower density. As a result, the computational burden incurred will be significantly reduced, as far fewer expensive points need be incorporated into our GP.

In our particular implementation, we fix *a-priori* a set of B possible maximum rate 'levels':

$$L = \{l_i \in (0, 1] | l_i < l_{i+1}, l_B = 1\}_{i=1}^{B}. \tag{2.23}$$

We then augment the variable set to include for each thinned point $\tilde{x}_m$ which rate level $r_m \in \{1 \dots B\}$ it is currently assigned, where we set r_m such that $\sigma(g(\tilde{x}_m)) \leq l_{r_m}$. This causes the probability of seeing a thinned point $\tilde{x}_m$ under the sigmoid GP to become

$$p(\tilde{x}_m | r_m) = \frac{l_{r_m} - \sigma(g(\tilde{x}_m))}{l_{r_m}}, \tag{2.24}$$

while the probability of a non-thinned point remains unchanged. Using this relationship we modify the Metropolis acceptance criteria which now become

$$a_{ins} = \frac{(1 - b(K, M+1))\mu(\mathcal{T})\lambda^* l_{r_{M+1}} p(\tilde{x}_{M+1}|r_{M+1})}{(M+1)b(K, M)}, \tag{2.25}$$

$$a_{del} = \frac{M b(K, M-1))}{(1 - b(K, M))\mu(\mathcal{T})\lambda^* l_{r_m} p(\tilde{x}_m|r_m)}, \tag{2.26}$$

as well as the likelihood function for $p(g(\mathbf{X_{M+K}}))$, Equation 2.20.

In principle this scheme could slow mixing, since the function is constrained to lie below the maximum level at each point. By ensuring that there is always some slack, s, between the function and the rate level assigned we find that mixing is hardly affected. The slack is incorporated by assigning the rate as follows:

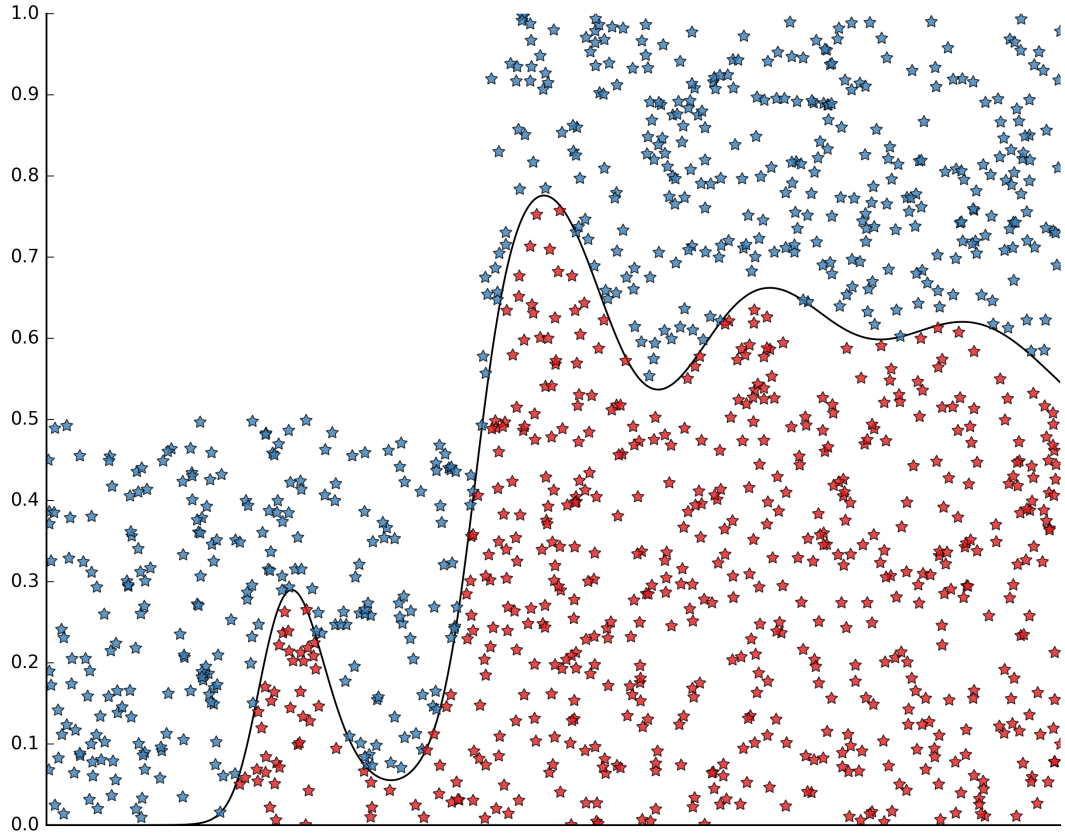
$$r_m \leftarrow \begin{cases} \underset{r}{\text{argmax}} \left\{ \sigma(g(\tilde{x}_m)) \leq l_r \times s \right\}, & \sigma(g(\tilde{x}_m)) \leq s \\ 1, & \text{otherwise.} \end{cases} \tag{2.27}$$

We used $s = 0.9$. The rate levels can also change at each iteration during the 'move' step when we compute the new rate level for jittered points and we compose the acceptance criteria as the product of the insertion and deletion criteria $a_{move} = a_{ins} \times a_{del}$.

Finally, when re-sampling λ^* we must compute an estimate of the total number of points under a single rate uniformisation of the space. This estimate is readily available because the number of points (including observed data points) with rate r is a Monte-Carlo integral of the proportion of space thinned to rate level l_r. Since the number of points in each region is scaled by l_r, the estimated total is

$$\hat{N}_{tot} = \sum_{k=1}^{K} \frac{1}{l_{\tilde{r}_k}} + \sum_{m=1}^{M} \frac{1}{l_{r_m}}, \tag{2.28}$$

Fig. 2.3 Graphical representation of adaptive thinning: Blue stars indicate thinned points, red stars represent data. The black line shows the intensity function. Each point is accepted as data with probability given by $\sigma(g(x_n))$. Fewer thinned points are required in areas of half maximum bound.



where $\bar{r}_k$ is the notional rate of observed data computed exactly as for the thinned points. The posterior value α_{post} is therefore $\alpha + \hat{N}_{tot}$.

To validate adaptive thinning, we return to the original SGCP and modify it in the manner described above. We perform two experiments: The first in 1D and the second in 2D. In both cases we use 10 known random intensity functions to generate event data: In the 1D case we sample 15 random datasets per function, while in the 2D case we generate 10. One dataset is used to learn the model, the rest are held out for testing purposes. Two metrics of performance are used: L2-norm error as measured between the posterior intensity and the

ground truth intensity function, and average predictive log-likelihood across all held out test datasets.

In 1D, we run each model for 6 minutes total compute time, in 2D we allow 20 minutes total. In both cases half the time is allocated to burn-in. In 1D the single rate method achieved roughly one sample per second, with the two rate case yielding just under four samples per second, and the four rate approach giving just under 8 samples per second. In 2D the number of points required was larger in all cases, with the multi-rate approach buying a factor of two speedup.

The results, (given in Tables 2.1 through 2.4), show that in almost all cases the multi-level approach performs best across both metrics. It is observed that typically the original, homogeneous rate approach performs worst of all.

Table 2.1 One dimensional adaptive thinning L2-norm function error.

Function (1D)	L2 Norm Error		
	Original	2 Rates	4 Rates
1	11.6	13.0	**7.1**
2	14.6	10.5	**7.2**
3	10.6	5.0	**4.9**
4	10.5	**4.7**	5.1
5	12.4	**10.1**	10.4
6	11.1	8.2	**7.6**
7	**12.0**	13.7	12.5
8	13.0	**12.0**	12.8
9	19.6	**16.4**	28.8
10	31.4	**27.2**	32.6

Table 2.2 One dimensional adaptive thinning average predictive log-likelihood on 14 held out datasets.

Function (1D)	Predictive Log-Likelihood		
	Original	2 Rates	4 Rates
1	373.8	381.8	**388.2**
2	626.1	644.2	**650.7**
3	274.2	285.1	**288.0**
4	435.5	**457.7**	456.0
5	877.0	885.8	**889.5**
6	995.4	1006.6	**1013.4**
7	753.0	**763.3**	760.6
8	522.3	**531.2**	528.3
9	1840.6	**1852.9**	1826.0
10	2328.1	**2365.8**	2349.8

Table 2.3 Two dimensional adaptive thinning L2-norm function error.

Function (2D)	L2 Norm Error		
	Original	2 Rates	4 Rates
1	13.3	13.4	**11.3**
2	14.3	**14.3**	14.7
3	13.5	14.7	**13.5**
4	12.5	12.9	**12.0**
5	17.9	**16.6**	17.8
6	**14.4**	16.2	15.3
7	15.1	**13.7**	17.5
8	14.6	**14.4**	14.8
9	18.3	17.1	**15.6**
10	15.4	**12.9**	13.6

Table 2.4 Two dimensional adaptive thinning average predictive log-likelihood on 9 held out datasets.

Function (2D)	Predictive Log-Likelihood		
	Original	2 Rates	4 Rates
1	2039.6	2080.6	**2203.0**
2	2757.9	2758.1	**2829.3**
3	2753.2	2689.5	**2827.2**
4	2803.2	2784.0	**2933.6**
5	2532.4	**2663.0**	2572.1
6	**3098.2**	3040.5	3054.4
7	3157.5	**3259.8**	3075.2
8	2086.9	**2101.0**	2087.4
9	5018.1	5146.6	**5185.3**
10	2008.1	**2205.0**	2174.0

2.5 Empirical Results

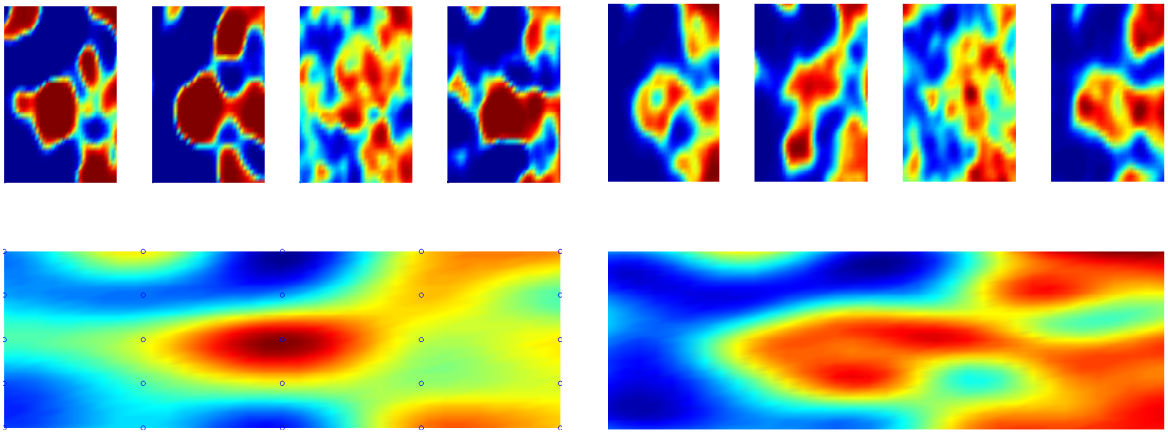
As this is the first model for structured point process data, the approach is initially validated on a synthetic dataset. It is then compared to both the independent SGCP, as well as a modern Kernel Density Estimator (Botev et al., 2010) on two real datasets.

2.5.1 Synthetic Data

Using the convolution process, we sample four intensity functions, using those to sample event data. The variety of intensities which may be observed given a single latent function is notable in Figure 2.4.

We then average over 2000 iterations after convergence had been achieved. The resulting learned intensity functions are shown in Figure 2.4. It is reassuring that the original latent function is reasonably well recovered given only four observed event processes.

Fig. 2.4 Learned functions using 3 rate levels: $\frac{1}{4}\lambda^*$, $\frac{1}{2}\lambda^*$, λ^*. Left hand half represents the synthetic generative data, with the single latent function at the bottom ($u_q(Z)$), and the four individual point process intensity functions above ($\{gd(X_d)\}_{d\in D}$). On the right hand side we depict their inferred counterparts. The blue circles in the latent function space denote the locations of the inducing points (Z).



2.5.2 Real Data

Two datasets were selected to test the model, both of which we considered were likely to exhibit a dependency structure which could be well captured by the convolution process.

- British politicians (MPs) tweet times during the week of Nelson Mandela's death (02/12/13-08/12/13). These were obtained using the Twitter API. Here we considered that there would naturally be a daily periodicity, however, it is not unreasonable to further postulate that some MPs may concentrate their Twitter activity into a smaller segment of the day. This behaviour should be well captured by the convolution process.

- NBA player shot profiles for the 2013-2014 season, scraped from the NBA website. Here we select a diverse subset of four players: Blake Griffin, Damien Lillard, DeMer DeRozen, and Arron Affalo. It was supposed that it might be possible for a single latent function to be blurred to represent a variety of player positions and styles.

2.5.3 Twitter Data Results

Four MPs active on twitter were selected at random. Here on we call them MPs A, B, C, and D. We select data from the period covering 02/12/13 through 13/12/13, and randomly partition each dataset into 75% training data, with the remainder being used to evaluate predictive test log-likelihood.

Figure 2.6 depicts the average learned intensity functions for each MP (red line), along with the one standard deviation bars (grey shading) derived from the function samples. The bottom plot depicts the learned latent driving function in the same manner.

The latent function clearly shows a strong daily period, particularly evident during the working week (02/12/13 was a Monday—corresponding to '1' in Figure 2.6). Furthermore, the largest two peaks in activity occur on the 3rd and the 5th of December. Potential contributing factors to these two spikes include a public sector strike, and the death of Nelson Mandela respectively.

Fig. 2.5 Learned basketball intensity functions using 3 rate levels: $\frac{1}{4}\lambda^*$, $\frac{1}{2}\lambda^*$, and λ^*.

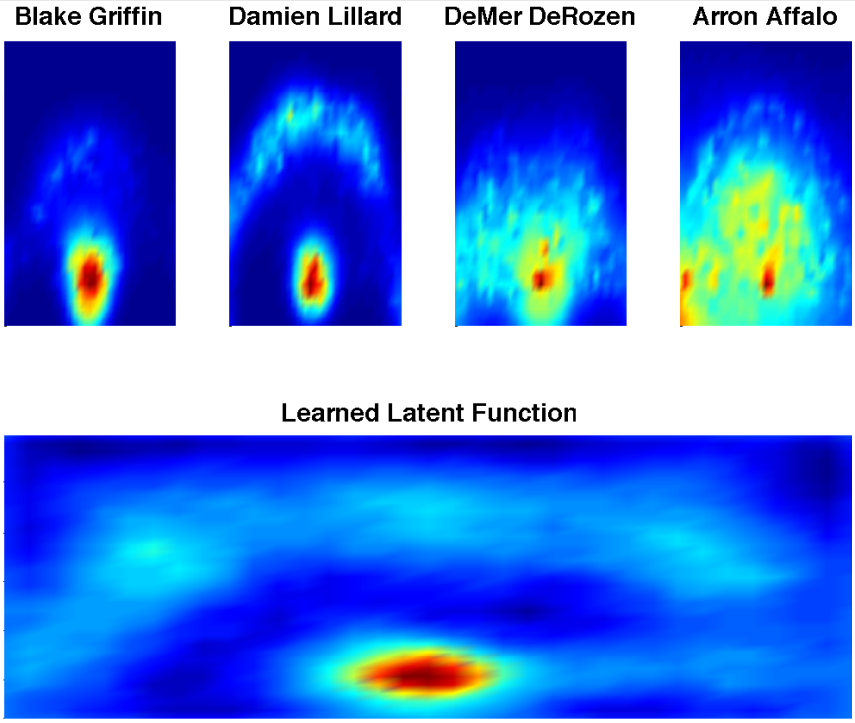


Fig. 2.6 Learned intensities over four MPs tweet data (A, B, C, D); posterior latent function at bottom. Actual data shown in blue.

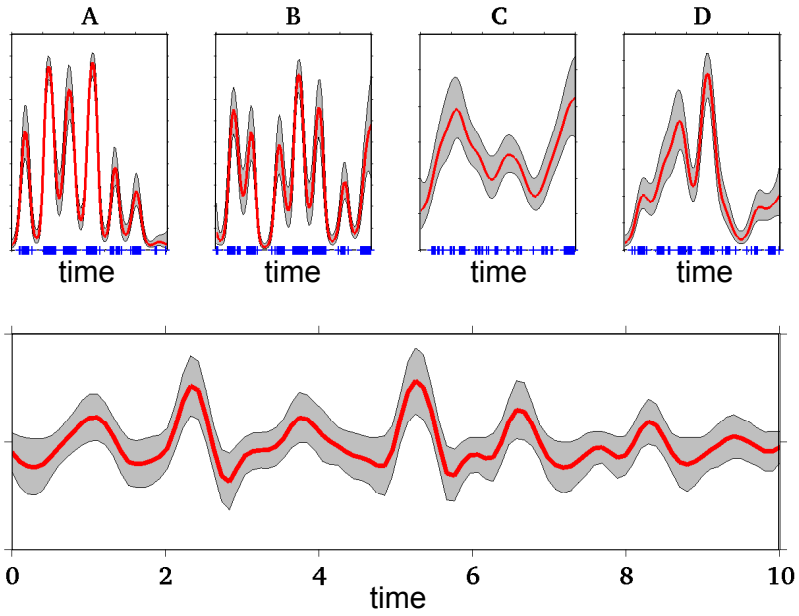


Table 2.5 Predictive log-likelihood for held out basketball data across models.

Player	Predictive Log-Likelihood		
	KDE	SGCP	Ours
Blake Griffin	-121.8	335.6	**374.7**
Damien Lillard	-22.6	231.2	**395.3**
DeMer DeRozen	-2.9	253.1	**410.7**
Arron Affalo	-260.7	-76.7	**84.2**

Table 2.6 gives predictive log-likelihood for the held out data, again evaluated across three approaches: An intensity function learned via Kernel Density Estimation (KDE) (Botev et al., 2010), the SGCP (Adams et al., 2009), and our own structured approach. Both the SGCP and our own approach use two maximum rate levels for adaptive thinning. Each intensity function is modelled using an independent SGCP/KDE. The structured approach performs vastly better, suggesting that it is highly appropriate for this type of data.

Table 2.6 Predictive log-likelihood for held out twitter data across models.

MP	Predictive Log-Likelihood		
	KDE	SGCP	Ours
A	177.1	176.6	**469.5**
B	-1.3	89.6	**412.9**
C	-5.4	49.6	**283.4**
D	-67.7	38.7	**293.7**

,

2.5.4 Basketball Data

For the basketball point shot data, the approach performed particularly well. Each player had around 600 attempted shots, of which we used 400, holding out the rest as test data. We used 3 rate boundaries: $\frac{1}{4}\lambda^*$, $\frac{1}{2}\lambda^*$, λ^*, and once again averaged over 2000 samples after convergence. We compare predictive log-likelihood to both the SGCP model (using the same set of rate boundaries) and using a rate function estimated via a state of the art KDE by (Botev et al., 2010).

Figure 2.5 depicts the resulting intensity functions. It is clear that the latent function represents a general view of the court hotspots, the hoop and three pointer line are clearly demarcated. Furthermore the intensity functions for each player strongly match what would be expected given their playing style—e.g. Arron Affalo is a 'shooting guard' who is expected to spend the majority of his time inside the three pointer line, but has a propensity to shoot from the bottom left of the court. These effects are clearly visible on the heat map, less so on the data (see Figures 2.1 and 2.5).

As is clearly demonstrated in Table 2.5, our structured approach to modelling the basketball point data in a fully Bayesian fashion yields a huge improvement over both the independent SGCP as well as a modern kernel density estimator. Another point worth making is that due to the high data density around the hoop for each player, the traditional approach of thinning (used here as well as in the SGCP) would be prohibitively computationally expen-

sive. We are only able to test on this data due to the method of adaptive thinning introduced in this Chapter.

2.5.5 Evaluation Metrics

Model estimation in the context of inhomogenous Poisson process data is an inherrently tricky problem, as we discussed in the introductory Chapter of this Thesis. As a result, it is worth spending a little time considering where the weak points of our performance evaluation scheme lie, how they compare to the methods used in work produced by the community, and whether there might be a more appropriate (or perhaps exhaustive) way to evaluate inference schemes and models.

We begin by considering the general setting—estimating a single inhomogenous Poisson process, using the adaptive thinning scheme proposed in Section . For synthetic data—i.e. where we are able to generate many realisations of the point process—we feel that little is missed by using the intensity function posterior to evaluate the predictive likelihood of held out test data sets, if our model is As is the case in all Bayesian models, a judiciously chosen set of priors could bias the model to a point where it outperforms on a specific dataset, however to guard against this we use the same prior for all synthetic realisations (and for that matter on the 'real' data tests too).

When it comes to the 'real' data evaluations in Section 2.5.2, however, it is clear that for a single realisation of the dataset (which we then split into training and test subsets), conditioning on a maximum-a-posteriori point estimate for λ^* may heavily bias our evaluation metric (in a positive or negative way). This is because the estimation process for λ^* is high variance (given a single realisation of the point process), and therefore any prior will strongly influence the result.

We are thus faced with a difficult conundrum: the model we fit *does* as part of the process attempt to estimate λ^*, however the high variance of this estimate combined with the bias effect of any prior means that it may be prudent to consider discounting it when evaluating

the model. As was pointed out by Prof. Chris Holmes, this may be achieved by evaluating held out test performance by normalising each posterior intensity function, and simply using the result as a density estimator for the held out data. This would mean that (at least some) effects caused by the estimation process for λ^* were discounted at test time; however, on the other hand we would now be evaluating the model under an inherrently different predictive likelihood function than the one that was used to estimate it.

While this is not the approach taken by the community at present, a pragmatic (although perhaps philosophically controversial) method of evaluating these models may be to: evaluate the model under the correct predictive likelihood exhaustively for synthetic test datasets (much as we did in this Chapter). Then, for the 'real' datasets, for which we only have a single point process realisation, test performance under *both* a density estimation framework, and the true Poisson process likelihood. Although it is the job of the model to both estimate λ^* and the intensity function, it would be highly unlikely that any given model had a consistently better estimation process for λ^* alone, therefore if the approach designed is truly more appropriate for the datasets presented, we would expect increased performance under the density evaluation metric as well as the true likelihood (on average). This testing approach trivially generalises to the case of the dependent Poisson process model, as presented in this Chapter.

2.6 Conclusion

We have introduced a fully generative model for dependent point processes, alongside an efficient, parallelised inference scheme. We have shown the appropriateness of this model on two real datasets, and introduced a new adaptation of thinning which allows the model to scale to larger datasets and in particular higher dimensional spaces.

Despite our improved MCMC driven inference scheme, in order to scale the model to very large (for the Cox process literature), heterogeneous datasets, it would seem sensible at this

juncture to pursue replacing the MCMC inference scheme with one based on variational inference (Hensman et al., 2013).

Chapter 3

Variational Inference for Cox Processes

The material in this chapter is based on the following paper:
C. Lloyd*, T. Gunter*, M. A. Osborne, and S. J. Roberts. Variational Inference for Gaussian Process Modulated Point Processes. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML, 2015,
where our contributions were as outlined in the acknowledgements section of this thesis.

In this chapter we present the first fully variational Bayesian inference scheme for continuous Gaussian-process-modulated Poisson processes. Our scheme: requires no discretisation of the domain; scales linearly in the number of observed events; and is many orders of magnitude faster than previous sampling based approaches. The resulting algorithm is shown to outperform standard methods on synthetic examples, coal mining disaster data, and in the prediction of Malaria incidences in Kenya.

3.1 Introduction

The use of a full Gaussian process to model the Cox process intensity (Adams et al., 2009) incurs prohibitive $\mathcal{O}(N^3)$ computational scaling in the number of data points, N. To tackle this problem in practice, many approaches (Møller et al., 1998; Rathbun and Cressie, 1994) discretise the domain, binning counts within each segment. This approach enabled (Cun-

ningham et al., 2008) to achieve $\mathcal{O}(N \log N)$ performance. However, the discretisation approach suffers from poor scaling with the dimension of the domain and sensitivity to the choice of discretisation.

We introduce a new model for Gaussian-process-modulated Poisson processes that eliminates the requirement for discretisation, while simultaneously delivering $\mathcal{O}(N)$ scaling. We further introduce the first fully variational Bayesian inference scheme for such models, allowing computation many orders of magnitude faster than existing schemes. This approach is shown to provide more accurate prediction than benchmarks on held-out data from datasets including synthetic examples, coal mining disaster data and Malaria incidences in Kenya. The power of our approach suggests many applications: in particular, our fully generative model permits the joint inference of real-valued covariates (such as log-rainfall) and a point process (such as disease outbreaks).

3.2 Notation and Preliminaries

As before, we define the Cox process via an intensity function $\lambda(\mathbf{x}) : \mathcal{X} \to \mathbb{R}^+$. For a domain $\mathcal{X} = \mathbb{R}^R$ of arbitrary dimension R, the number of atomic point masses, $N(\mathcal{T})$, found in a subregion $\mathcal{T} \subset \mathcal{X}$ is Poisson distributed with parameter $\lambda_{\mathcal{T}} = \int_{\mathcal{T}} \lambda(\mathbf{x}) \, \mathrm{d}\mathbf{x}$, where $\mathrm{d}\mathbf{x}$ indicates integration with respect to a Lebesgue measure over the domain. In addition for disjoint subsets $\mathcal{T}_i$ of $\mathcal{X}$, the counts $N(\mathcal{T}_i)$ are independent random variables.

If we restrict our consideration to some bounded region, $\mathcal{T}$, the probability of a set of N observed points, $\mathcal{D} = \{\mathbf{x}^{(n)} \in \mathcal{T}\}_{n=1}^{N}$, conditioned on the rate function $\lambda(\mathbf{x})$ is

$$p(\mathcal{D} \mid \lambda) = \exp\left\{ - \int_{\mathcal{T}} \lambda(\mathbf{x}) \, \mathrm{d}\mathbf{x} \right\} \prod_{n=1}^{N} \lambda(\mathbf{x}^{(n)}). \tag{3.1}$$

We use $\omega(V)$ to denote the measure of the continuous domain V. In this work we will assume $\mathcal{T}$ is a box-bounded subset of $\mathbb{R}^R$ with boundaries $\mathcal{T}_r^{\min}$ and $\mathcal{T}_r^{\max}$ in each dimension r and

$$\omega(\mathcal{T}) = \int_{\mathcal{T}} \mathrm{d}\mathbf{x} = \prod_{r=1}^{R} (\mathcal{T}_r^{\max} - \mathcal{T}_r^{\min}). \tag{3.2}$$

3.2.1 Inferring Intensity Functions

As we saw in Chapter 2, in the SGCP a Gaussian process (Rasmussen and Williams, 2006) is used to construct an intensity function prior by passing a random function, $f \sim \mathcal{GP}$, through a sigmoid transformation and scaling it with a maximum intensity λ^*. The intensity function is therefore $\lambda(x) = \lambda^* \sigma\big(f(x)\big)$, where $\sigma(\cdot)$ is the logistic sigmoid (squashing) function

$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \tag{3.3}$$

To remove the inner intractable integral, the authors augment the variable set to include latent data, such that the joint distribution of the latent and observed data is uniform Poisson over the region $\mathcal{T}$. While this model works well in practice on small, sparse event data in low dimensions, in reality, it scales poorly with both the dimensionality of the domain and the maximum observed density of points. This is due to: the incorporation of latent, or thinned, data, whose number grows exponentially with the dimensionality of the space; and an $\mathcal{O}(N^3)$ cost in the number N of all data (thinned or otherwise).

In Chapter 2 and Gunter* et al. (2014), we go some way towards improving the scalability of the SGCP, by introducing a further set of latent variables such that the entire space need no longer be thinned uniformly. Instead, we thinned to a piecewise uniform Poisson process, maintaining the tractability of the inner integral, and allowing the model to scale to higher dimensional point processes.

In Lasko (2014) the author performs renewal process inference without thinning the domain, by making use of a positively transformed intensity function. The intractability of their chosen approach forces them to resort to numerical integration techniques, however, and Bayesian inference is still performed using computationally expensive sampling.

3.3 Model

We construct our prior over the rate function using a Gaussian process. Rather than using a squashing function, we will assume[1] the intensity function is simply defined as $\lambda(\mathbf{x}) = g^2(\mathbf{x})$ where g is a Gaussian process distributed random function. This yields a non-negative prior

$$g(\mathbf{x}) \sim \mathcal{GP}\big(\mu(\mathbf{x}), \Sigma(\mathbf{x}, \mathbf{x}')\big), \tag{3.4}$$

as in Gunter et al. (2014). Furthermore we will assume that g is conditionally dependent on another Gaussian process u evaluated at a set of *inducing* points $\mathcal{Z} = \{\mathbf{z}^{(m)} \in \mathcal{T}\}_{m=1}^{M}$. We denote the evaluation of u at these points $\mathbf{u}$, and note $\mathbf{u}$ has distribution:

$$\mathbf{u} \sim \mathcal{N}(\vec{1}\bar{u}, \mathbf{K}_{zz}). \tag{3.5}$$

Using this formulation, the mean and covariance functions of g are

$$\mu(\mathbf{x}) = \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{u}, \tag{3.6}$$

$$\Sigma(\mathbf{x}, \mathbf{x}') = \mathbf{K}_{xx'} - \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{k}_{zx'}, \tag{3.7}$$

where $\mathbf{k}_{xz}$, $\mathbf{K}_{xx'}$, $\mathbf{K}_{zz}$ are matrices evaluated at $\mathbf{x}$, $\mathbf{x}'$ and $\mathcal{Z}$ using an appropriate kernel. We use the exponentiated quadratic (also known as the "squared exponential") ARD kernel although a variety of other kernels would also be viable choices (see Appendix A.1 for a detailed specification).

[1] See Section 3.5 for a detailed motivation for this choice.

With this hierarchical formulation the joint distribution over $\mathscr{D}$, g, $\mathbf{u}$ and Θ is

$$p(\mathscr{D}, g, \mathbf{u}, \Theta) = p(\mathscr{D}|\lambda = g^2)p(g|\mathbf{u}, \Theta)p(\mathbf{u}|\Theta)p(\Theta) \tag{3.8}$$

where $p(\Theta)$ is the (optional) prior on the set of model parameters $\Theta = \{\gamma, \alpha_1, \ldots, \alpha_R, \bar{u}\}$. For notational convenience we will usually omit explicitly conditioning on Θ.

3.4 Inference

We will use variational inference to obtain a lower bound on the model evidence $p(\mathscr{D})$. We favour the black-box approach to variational inference, where rather than determine analytic update equations, we specify up front variational posteriors over our unknown variables. We then employ Jensen's rule (Bishop, 2007) to allow us to arrive at a lower bound on the log-marginal likelihood, which may then be maximised with respect to the parameters of the variational posteriors using the optimisation algorithm of your choice.

In this case, we must integrate out g and $\mathbf{u}$, but we must also integrate g^2 over the region $\mathscr{T}$ due to the integral embedded in the likelihood, Equation 3.1.

3.4.1 Variational Bound

We begin by integrating out the latent function u, using a variational distribution $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m}, \mathbf{S})$ over the inducing points. Since only g is conditioned on $\mathbf{u}$ and since $q(\mathbf{u})$ is conjugate to $p(g|\mathbf{u})$, the resulting integral is:

$$\begin{aligned}
\log p(\mathscr{D}|\Theta) &= \log\left[\iint p(\mathscr{D}|g)p(g|\mathbf{u})p(\mathbf{u})\frac{q(\mathbf{u})}{q(\mathbf{u})}\,\mathrm{d}\mathbf{u}\,\mathrm{d}g\right] \\
&\geq \iint p(g|\mathbf{u})q(\mathbf{u})\,\mathrm{d}\mathbf{u}\;\log[p(\mathscr{D}|g)]\,\mathrm{d}g \\
&\quad + \iint p(g|\mathbf{u})q(\mathbf{u})\,\mathrm{d}g\log\left[\frac{p(\mathbf{u})}{q(\mathbf{u})}\right]\,\mathrm{d}\mathbf{u} \\
&= \mathbb{E}_{q(g)}\left[\log p(\mathscr{D}|g)\right] - \mathrm{KL}\big(q(\mathbf{u})||p(\mathbf{u})\big) \\
&\triangleq \mathscr{L}
\end{aligned} \tag{3.9}$$

As $p(g|\mathbf{u})$ is conjugate to $q(\mathbf{u})$, we can write down in closed-form the resulting integral:

$$\begin{aligned}
q(g) &= \int p(g|\mathbf{u})q(\mathbf{u})\mathrm{d}\mathbf{u} = \mathscr{GP}(g;\tilde{\mu},\tilde{\Sigma}), \\
\tilde{\mu}(\mathbf{x}) &= \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{m}, \\
\tilde{\Sigma}(\mathbf{x},\mathbf{x}') &= \mathbf{K}_{xx'} - \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{k}_{zx'} + \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{S}\mathbf{K}_{zz}^{-1}\mathbf{k}_{zx'}.
\end{aligned} \tag{3.10}$$

$\mathrm{KL}\big(q(\mathbf{u})||p(\mathbf{u})\big)$ is simply the KL-divergence between two Gaussians (see Appendix A.3).

$$\begin{aligned}
\mathrm{KL}\big(q(\mathbf{u})||p(\mathbf{u})\big) = \frac{1}{2}\Bigg[&\mathrm{tr}\left(\mathbf{K}_{zz}^{-1}\mathbf{S}\right) + \log\frac{|\mathbf{K}_{zz}|}{|\mathbf{S}|} - M \\
&+ (\vec{1}\bar{u} - \mathbf{m})^\top\mathbf{K}_{zz}^{-1}(\vec{1}\bar{u} - \mathbf{m})\Bigg].
\end{aligned} \tag{3.11}$$

We can now take expectations of the data log-likelihood under $q(g)$:

$$\begin{aligned}
\mathscr{L} &= \mathbb{E}_{q(g)}\left[\log p(\mathscr{D}|g)\right] - \mathrm{KL}\big(q(\mathbf{u})||p(\mathbf{u})\big) \\
&= \mathbb{E}_{q(g)}\left[-\int_{\mathscr{T}} g_x^2\,\mathrm{d}\mathbf{x} + \sum_{n=1}^{N}\log g_n^2\right] \\
&\quad - \mathrm{KL}\big(q(\mathbf{u})||p(\mathbf{u})\big) \\
&= -\int_{\mathscr{T}}\left\{\mathbb{E}_{q(g)}[g_x]^2 + \mathrm{Var}_{q(g)}[g_x]\right\}\mathrm{d}\mathbf{x} \\
&\quad + \sum_{n=1}^{N}\mathbb{E}_{q(g)}\left[\log g_n^2\right] - \mathrm{KL}\big(q(\mathbf{u})||p(\mathbf{u})\big), \qquad (3.12)
\end{aligned}$$

where to keep the notation concise we have introduced the following identities:

$$\begin{aligned}
g_x &\triangleq g(\mathbf{x}), \qquad \tilde{\mu}_x \triangleq \tilde{\mu}(\mathbf{x}), \qquad \tilde{\sigma}_x^2 \triangleq \tilde{\Sigma}(\mathbf{x},\mathbf{x}), \\
g_n &\triangleq g(\mathbf{x}^{(n)}), \quad \tilde{\mu}_n \triangleq \tilde{\mu}(\mathbf{x}^{(n)}), \quad \tilde{\sigma}_n^2 \triangleq \tilde{\Sigma}(\mathbf{x}^{(n)},\mathbf{x}^{(n)}).
\end{aligned}$$

We have used Tonelli's theorem to reverse the ordering of the integrations over the positive integrand $g_x^2 q(g)$. Before arriving at the final model bound, we must first integrate over the region $\mathscr{T}$ and compute the expectations $\mathbb{E}_{q(g)}\left[\log g_n^2\right]$ at the data points.

3.4.2 Integrating Over the Region $\mathscr{T}$

The lower bound allows us to take expectations under $q(g)$ at any specific point, $\mathbf{x}$, of the function value, $g(\mathbf{x})$, since $q(g(\mathbf{x}))$ is Gaussian. The fact that these operations are analytic, arises because: a) we used the conditional GP formulation; b) we have already integrated out the latent function u; and c) we chose a suitable transformation to map the output of the GP into $\mathbb{R}^+$, i.e. $\lambda(\mathbf{x}) = g^2(\mathbf{x})$.

The required statistics for Equation 3.12 are:

$$\mathbb{E}_{q(g)}[g_x]^2 = \tilde{\mu}_x^2 = \mathbf{m}^\top \mathbf{K}_{zz}^{-1} \mathbf{k}_{zx} \mathbf{k}_{xz} \mathbf{K}_{zz}^{-1} \mathbf{m}, \tag{3.13}$$

$$\begin{aligned} \mathrm{Var}_{q(g)}[g_x] = \tilde{\sigma}_x^2 = \mathbf{k}_{xx} - \mathrm{Tr}(\mathbf{K}_{zz}^{-1} \mathbf{k}_{zx} \mathbf{k}_{xz}) \\ + \mathrm{Tr}(\mathbf{K}_{zz}^{-1} \mathbf{S} \mathbf{K}_{zz}^{-1} \mathbf{k}_{zx} \mathbf{k}_{xz}). \end{aligned} \tag{3.14}$$

It is now easy to calculate the integral since only $\mathbf{k}_{zx} = \mathbf{k}_{xz}^\top$ is a function of $\mathbf{x}$, leading to the following terms:

$$\int_{\mathscr{T}} \mathbb{E}_{q(g)}[g_x]^2 \mathrm{d}\mathbf{x} = \mathbf{m}^\top \mathbf{K}_{zz}^{-1} \Psi \mathbf{K}_{zz}^{-1} \mathbf{m}, \tag{3.15}$$

$$\begin{aligned} \int_{\mathscr{T}} \mathrm{Var}_{q(g)}[g_x] \mathrm{d}\mathbf{x} = \gamma |\mathscr{T}| - \mathrm{Tr}(\mathbf{K}_{zz}^{-1} \Psi) \\ + \mathrm{Tr}(\mathbf{K}_{zz}^{-1} \mathbf{S} \mathbf{K}_{zz}^{-1} \Psi). \end{aligned} \tag{3.16}$$

For the exponentiated quadratic ARD kernel, the matrix

$$\Psi = \int K(\mathbf{z}, \mathbf{x}) K(\mathbf{x}, \mathbf{z}') \, \mathrm{d}\mathbf{x} \tag{3.17}$$

can be calculated by re-arranging the product as a single exponentiated quadratic in $\mathbf{x}$ and $\bar{\mathbf{z}}$, where $\bar{\mathbf{z}} = [\bar{z}_1, \ldots, \bar{z}_R]^\top$ has elements $\bar{z}_r = \frac{z_r + z_r'}{2}$, as derived in Appendix A.5.

In addition to the exponentiated quadratic ARD kernel, the matrix Ψ can be computed in closed-form for many other kernels, including polynomial and periodic kernels, as well as sum and product combinations of kernels (including the spectral kernel (Wilson et al., 2014)). The only limitation we should be aware of, is that for many of these kernels the integral remains tractable in arbitrary dimensions *under the assumption* that we remain strictly axis-aligned—this is certainly true for any kernel which includes the Gaussian function as a constituent component.

3.4.3 Expectations at the Data Points

The expectation $\mathbb{E}_{q(g)}[\log g_n^2]$ has an analytical—albeit complicated—solution expressed as

$$\mathbb{E}_{q(g)}[\log g_n^2] = \int_{-\infty}^{\infty} \log(g_n^2)\, \mathcal{N}(g_n, \tilde{\mu}_n, \tilde{\sigma}_n^2)\, \mathrm{d}g_n \tag{3.18}$$

$$= -\tilde{G}\left(-\frac{\tilde{\mu}_n^2}{2\tilde{\sigma}_n^2}\right) + \log\left(\frac{\tilde{\sigma}_n^2}{2}\right) - C, \tag{3.19}$$

where $C \approx 0.57721566$ is the Euler-–Mascheroni constant and $\tilde{G}$ is defined via the confluent hyper-geometric function (assuming we allow $r \coloneqq -\frac{\tilde{\mu}_n^2}{2\tilde{\sigma}_n^2}$):

$${}_1F_1\,(a, b, r) = \sum_{k=0}^{\infty} \frac{(a)_k r^k}{(b)_k\; k!}, \tag{3.20}$$

where $(\cdot)_k$ denotes the rising Pochhammer series

$$(a)_0 = 1, \quad (a)_k = a(a+1)(a+2)\ldots(a+k-1).$$

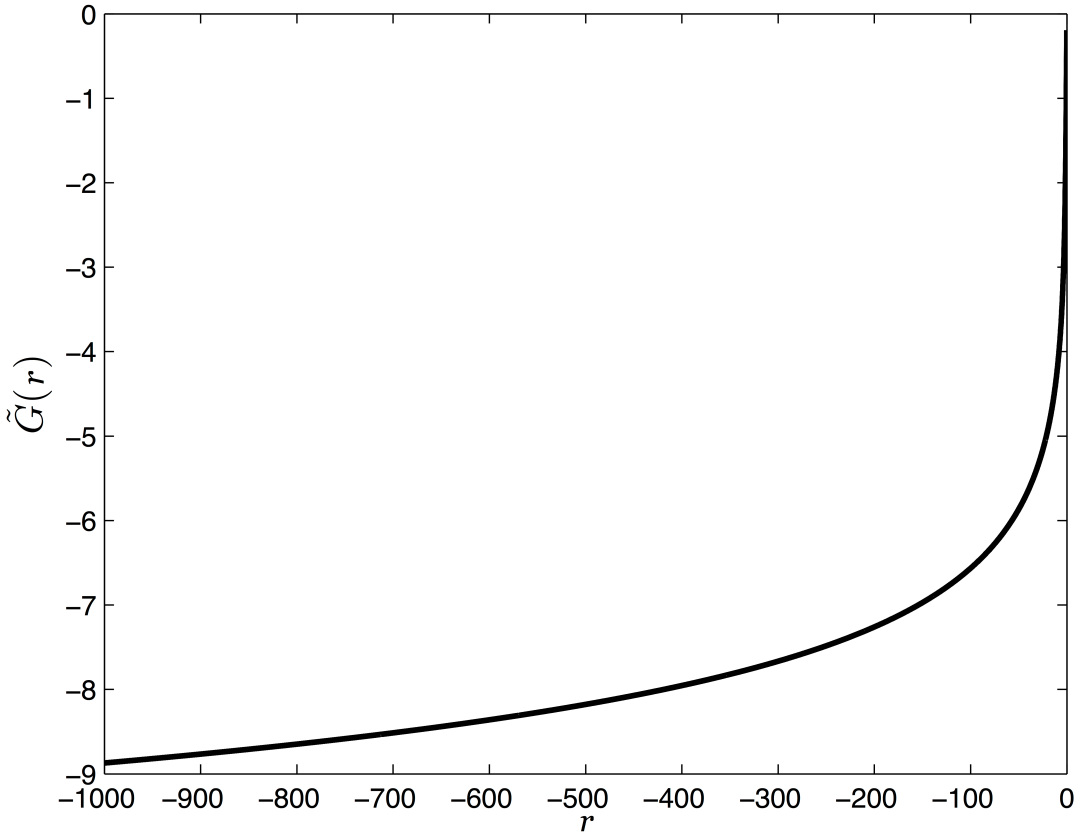
Specifically $\tilde{G}$ is a specialised version of the partial derivative of ${}_1F_1$ with respect to its first argument and can be computed using the method of Ancarani and Gasaneo (2008), which has a particular solution at $a = 0$, leading to the following definition of $\tilde{G}$:

$$\tilde{G}(r) = {}_1F_1{}^{(1,0,0)}\left(0, \frac{1}{2}, r\right) = 2r \sum_{j=0}^{\infty} \frac{j!\;\; r^j}{(2)_j (1\tfrac{1}{2})_j}. \tag{3.21}$$

Naive implementations of Equation 4.10 have poor numerical stability, although this can be improved somewhat using an iterative scheme. In practice we use a large multi-resolution look-up table of precomputed values obtained from a numerical-package. As shown in Figure 3.1, this function decreases very slowly as its argument becomes increasingly negative,

so we can easily compute accurate evaluations of $\tilde{G}(r)$ for any r by linear interpolation of our lookup table and, as a by-product, we also obtain $\tilde{G}'$ (see also Appendix A.4).

Fig. 3.1 A plot of $\tilde{G}(r)$ across a wide range of r. This part of the model lower bound, while not closed form in expression, is very smooth and therefore well approximated by a partial expansion or simple function approximation.



3.4.4 Optimising the Bound

Given this final lower bound on the marginal likelihood, $\mathscr{L}$, we maximise with respect to the variational parameters $\mathbf{m}$, $\mathbf{S}$ and the model parameters Θ. To optimise these simultaneously we construct an augmented vector $\mathbf{y} = [\Theta^\top, \mathbf{m}^\top, \text{vech}(\mathbf{L})^\top]^\top$—where vech($\mathbf{L}$) is the vectorisation of the lower triangular elements of $\mathbf{L}$, such that $\mathbf{S} = \mathbf{L}\mathbf{L}^\top$.

We can compute the maximum-likelihood (ML) solution by optimising $\mathscr{L}(\mathscr{D}; \mathbf{y})$ or the maximum-a-posterior (MAP) estimate by maximising $\mathscr{L}(\mathscr{D}; \mathbf{y}) + \log p(\Theta)$.

To optimise the inducing point locations we use the change of variables

$$z_r^{(m)} = \frac{\mathscr{T}_r^{\min} + \mathscr{T}_r^{\max}}{2} - \frac{\mathscr{T}_r^{\min} - \mathscr{T}_r^{\max}}{2} \sin\left(\omega_r^{(m)}\right), \tag{3.22}$$

and optimise in $\omega_r^{(m)} \in [-\pi, \pi]$, which ensures the inducing points always remain within the region $\mathscr{T}$.

3.4.5 Locating the Inducing Points

We have so far neglected to specify the number and location of inducing points. In principle, for a given set of parameters Θ, we will obtain a lower bound for the true GP likelihood for any number of inducing points in any configuration of locations. We consider two possible approaches: firstly, treating the inducing points as optimisation parameters and, secondly, fixing them on a regular grid.

If the locations of the inducing points are optimised, this suggests—in common with other sparse GP models—that we might achieve good performance using only a small number of well-placed inducing points. Optimisation of the inducing points is particularly computationally expensive, however, because of the necessity to recompute $\mathbf{K}_{zz}^{-1}$ for each dimension of each inducing point ($M \times R$ in total) and since $\mathbf{K}_{zz}$ affects every term in the bound.

Regular grids offer a competing set of advantages and disadvantages. In contrast to standard sparse GP regression—the accuracy of our solution is not only governed by the distance between the inducing points and the data points. This is due to our Cox process likelihood, and choice of square transformation applied to the GP. The variance of $g(\mathbf{x})$ increases as $\mathbf{x}$ becomes further from the inducing points. However, the rate function, λ, is a function of both the mean *and* the variance of g. Since we are integrating λ over $\mathscr{T}$, we need inducing points distributed across $\mathscr{T}$ and not just in regions close to the data. An evenly-spaced grid is one way to ensure this is achieved.

Regularly sampled grids also afford potential computational advantages. When the grid points are evenly spaced, the kernel matrix has Toeplitz structure, and hence allows matrix inversion (and linear solving) in $\mathcal{O}(M \log^2 M)$ time, a fact previously utilised for efficient point processes by Cunningham et al. (2008). Furthermore, when the kernel function is separable across the dimensions (as specified by Equation (A.1)), the kernel matrix has Kronecker structure which can further reduce the cost of matrix inversion (Osborne et al., 2012b). The latter is relevant to all sparse GP applications based on inducing points, however, it is particularly relevant for this application as we are motivated by the doubly intractable nature of Equation 2.1. In our implementation, we use naïve inversion of the inducing point kernel matrix, $\mathbf{K}_{zz}$, resulting in computational complexity of $\mathcal{O}(NM^3)$. Hence the computation times reported below could be improved with a relatively small amount of additional implementation effort.

3.4.6 Predictive Distribution

To form the predictive distribution we assume our optimised variational distribution $q^*(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m}^*, \mathbf{S}^*)$ approximates the posterior $p(\mathbf{u}|\mathcal{D})$. Analogously to Equation 3.10 we next compute $q^*(g) \approx p(g|\mathcal{D})$. This allows us to provide a lower bound of the (approximate) predictive log-likelihood on a held-out test dataset $\mathcal{H}$:

$$\begin{aligned} \log p(\mathcal{H}|\mathcal{D}, \Theta^*) &= \log \mathbb{E}_{p(g|\mathcal{D})}[p(\mathcal{H}|g)] \\ &\approx \log \mathbb{E}_{q^*(g)}[p(\mathcal{H}|g)] \triangleq \mathcal{M}_p \\ &\geq \mathbb{E}_{q^*(g)}[\log p(\mathcal{H}|g)] \triangleq \mathcal{L}_p. \end{aligned} \tag{3.23}$$

The derivation of $\mathcal{L}_p$ follows Equations 3.12-3.19. The resulting bound is the same as $\mathcal{L}$ except $\mathbf{m}$, $\mathbf{S}$ are replaced with $\mathbf{m}^*$ and $\mathbf{S}^*$, and there is no KL divergence term. Kernel matrices are computed using Θ^*.

The tightness of this final bound—i.e. how close we are to evaluating the true predictive likelihood—will be a function of how well the inducing variables $\mathbf{u}$ define the function f. Intuitively, when the variance at the inducing points is large, the entropy of g will be large, as it is unconstrained over much of the domain. From an information theoretic perspective, we can say that the tightness of the bound will be a function of the entropy of g: $H(g) = \frac{1}{2}\log|\tilde{\Sigma}| + \text{const.}$

Given this knowledge we define a second, tightened, lower bound $\mathscr{L}_0$, where we allow the variance of function values at the inducing points to collapse to zero: $p(\mathbf{u}|\mathscr{D}) \approx \delta(\mathbf{m})$. This reduces the conditional entropy of g given $\mathbf{u}$ by shrinking the final term in the definition of $\tilde{\Sigma}$ (Equation 3.10), resulting in a tighter bound for a slightly restricted class of models. As it is a slightly more restricted model, we expect the ground truth $\mathscr{M}_0 = \log\left(\mathbb{E}_{q(g|\mathbf{u}=\mathbf{m})}\left[p(\mathscr{H}|g)\right]\right)$ to be lower than the ground truth $\mathscr{M}_p$. In practice, however, because the variational $\mathscr{L}_0$ is so much tighter, we use this to give results for approximate predictive likelihood when comparing against other approaches. This is fair, as we are not adapting the model in any way, but instead simply ensuring we accurately measure the predictive likelihood provided by the model on held-out test data. In Figure 3.7 we demonstrate this empirically for the coal mining dataset by evaluating the true predictive bounds on a held-out 50% of the data via $10,000$ MCMC samples, and shading between these and the variational approximations. We do this for a range of inducing point grid densities, both with and without optimisation of the inducing point locations. In Figures 3.5 through 3.7, we plot all of the bounds described above as the number of inducing points increase. We note that for the relatively smooth coal mining data, (Figure 3.7), all bounds do not benefit from more than about 10 inducing points, while in the case of the twitter data, the faster dynamics call for increased numbers. In all three Figures the tightness of $\mathscr{L}_0$ is evident as compared to $\mathscr{L}_p$.

3.5 Alternative GP transformations

At this point it is worth considering why we have chosen the function transformation $\lambda(\mathbf{x}) = g^2(\mathbf{x})$ in preference to other alternatives we might have used. An obvious first choice would be

$$\lambda(\mathbf{x}) = \exp\big(g(\mathbf{x})\big). \tag{3.24}$$

This transformation is undesirable for two reasons. The more obvious of these is that after taking expectations under $q(g)$ we are left with the integral

$$-\int_{\mathscr{T}} \exp\left(\tilde{\mu}_x + \frac{\tilde{\sigma}_x^2}{2}\right) \mathrm{d}\mathbf{x} \tag{3.25}$$

which cannot be computed in closed form. We could approximate the integral using a series expansion, however this would be very difficult with more than a couple of terms and furthermore, since the function is concave, this approximation would not be a lower bound.

The second—and more subtle—reason is that in using this transformation, when we take expectations under $q(g)$ of the data, we obtain

$$\mathbb{E}_{q(g)}\left[\log\left\{\exp(g_n)\right\}\right] = \tilde{\mu}_n. \tag{3.26}$$

Since the mean, $\tilde{\mu}_n$, is not a function of $\mathbf{S}$, the variance of the variational distribution $q(\mathbf{u})$, we have effectively decoupled the data from the uncertainty of our variational approximation; this is clearly undesirable.

Another possible candidate is the probit function, $\lambda(x) = \Phi(g(x))$. This can be integrated analytically against the GP prior, however we are again left with a difficult integral over $\mathscr{T}$

$$-\int_{\mathscr{T}} \Phi\left(\frac{\tilde{\mu}_x}{\sqrt{1+\tilde{\sigma}_x^2}}\right) \mathrm{d}\mathbf{x}. \tag{3.27}$$

As the range of this transformation is $(0, 1)$ we would also require additional machinery to infer a scaling variable.

In contrast the square transform presented allows the integral over the region $\mathscr{T}$ to be computed in closed form and $\mathbb{E}_{q(g)}[\log g_n^2]$ may be computed analytically. Importantly this transformation also maintains the connection between the data and the variational uncertainty.

Although the square transform is not a one-to-one function—any rate function λ may have been generated by g^2 or $(-g)^2$—this sign ambiguity is integrated out in a Bayesian sense, Equation 3.18.

3.6 Relationship to Sparse GP Models

The use of inducing points in this model relates it to a wide range of sparse Gaussian process models, e.g. the Sparse Pseudo-input Gaussian Process (SPGP) (Snelson and Ghahramani, 2005). The variational sparse Gaussian process framework was introduced by Titsias (2009), however the bound we develop is more akin to the "Big-Data" GP bound (Hensman et al., 2013), since we explicitly maintain the variational distribution $q(\mathbf{u})$. The variable Ψ that results from integrating the kernel over the input domain is similar to the so-called "Ψ-statistic" which arises when integrating out the uncertainty of latent variables in the variational Gaussian Process Latent Variable Model (GPLVM) (Titsias and Lawrence, 2010).

3.7 Experiments

To evaluate the performance of this algorithm, we benchmarked against a frequentist kernel smoothing approach, described below, (testing both Kernel Smoothing with Edge Correction (KS+EC) and Kernel Smoothing without Edge Correction (KS–EC)), as well as a fully Bayesian SGCP MCMC sampler. Our test data sets are generative data from the SGCP model, as well as several real-world data sets.

3.7.1 Benchmarks

The kernel smoothing method we choose is similar to standard kernel density estimation, except we use truncated normal kernels to account for our explicit knowledge of the domain—the latter is referred to as "end-correction" in the literature (Diggle, 1985). The kernel smoother optimises a diagonal covariance, Σ^*, by maximising the leave-one-out training objective

$$\Sigma^* = \operatorname*{argmax}_{\Sigma} \sum_{i=1}^{N} \log \sum_{j \neq i=1}^{N} \mathcal{N}_T(\mathbf{x}^{(i)}; \mathbf{x}^{(j)}, \Sigma). \tag{3.28}$$

We can construct the predictive distribution by combining the maximum-likelihood estimates of the size and spatial location of the point process. For the test data set $\mathscr{H}$ (with $K!$ permutations) this distribution is

$$p(\mathscr{H}|\mathscr{D}) = K!\ p(K|\mathscr{D}) \prod_{k=1}^{K} p(\tilde{\mathbf{x}}^{(k)}|\mathscr{D}) \tag{3.29}$$

where the location density

$$p(\tilde{\mathbf{x}}^{(k)}|\mathscr{D}) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{N}_{\mathcal{T}}(\tilde{\mathbf{x}}^{(k)}; \mathbf{x}^{(n)}, \Sigma^*) \tag{3.30}$$

Table 3.1 Results for 2D synthetic data (drawn from the SGCP generative process).

	SGCP			KS+EC			VBPP ($\mathscr{L}_0$)		
Function	Avg. LL	RMS	Time(s)	Avg. LL	RMS	Time(s)	Avg. LL	RMS	Time(s)
1	446.1	1.37	7547.83	389.8	1.48	0.34	392.9	1.21	3.26
2	-61.1	0.38	1039.65	-78.3	0.46	0.02	-76.1	0.38	2.00
3	122.4	0.88	3173.91	84.3	1.04	0.12	92.6	0.81	2.44
4	175.8	1.71	3773.75	147.0	1.26	0.05	148.3	1.14	2.58
5	446.1	2.94	6368.44	413.6	2.02	0.21	415.5	1.81	2.83

is computed using using the previously described method and the distribution of the number of points

$$p(K|\mathscr{D}) = \frac{N^K}{K!}\exp(-N) \tag{3.31}$$

is simply a Poisson distribution with parameter N. It is straight forward to show that Equation 3.29 is equivalent to Equation 3.1 since we can interpret the rate function as

$$\lambda(\mathbf{x}) = \sum_{n=1}^{N} \mathscr{N}_{\mathscr{T}}(\mathbf{x}; \mathbf{x}^{(n)}, \Sigma^*) \tag{3.32}$$

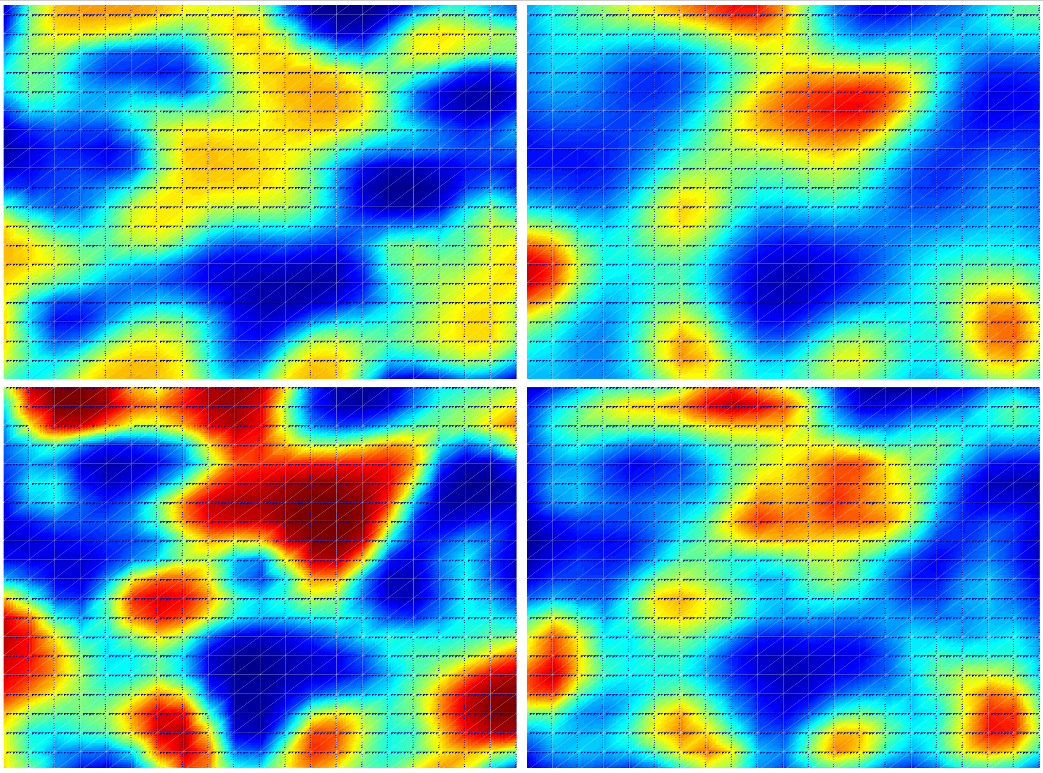
and since $\int_T \lambda(\mathbf{x})\mathrm{d}\mathbf{x} = N$.

Our SGCP sampler is based on (Adams et al., 2009). This implementation differs by using elliptical slice sampling to infer the latent functions function g and we perform hyper-parameter inference using HMC. We also use the "adaptive-thinning" method described in Chapter 2 and Gunter* et al. (2014) in order to reduce the number of thinning points required.

3.7.2 Synthetic Data

For the synthetic data sets, we first generate a 2D function from a high resolution grid using a Gaussian process and sigmoid link function, and then, conditioned on that function, we draw a training dataset and multiple test datasets. We give average performances results for these test datasets in Table 3.1. Figure 3.2 visualises an inferred 2D intensity conditioned

Fig. 3.2 2D Synthetic Data. Clockwise from top left: Ground truth, VBPP, KS+EC, SGCP.



on $\sim$ 500 observations. Although the SGCP sampler gives better predictive performance than the Variational Bayes model for point processes (VBPP) $\mathscr{L}_0$ bound, it should be noted that the sampler uses well tuned hyper-parameters, uses the same link function as the generative process and is much more computationally expensive. VBPP outperforms kernel smoothing in terms of both predictive likelihood and RMS error.

In general it is clear that the variational method performs approximately equivalently to the SGCP in the majority of cases, and typically surpasses the results delivered by both KDE methodologies. VBPP arrives at a full posterior over the function in a fraction of the time the SGCP does, albeit typically slower than the maximum-likelihood solution delivered by KDE.

3.7.3 Real Data

We next investigate three real world data sets. For these data sets we create training and test subsets by allocating each point to either subset with probability 0.5. Since true rates are

Fig. 3.3 Coal mining data and predictions.

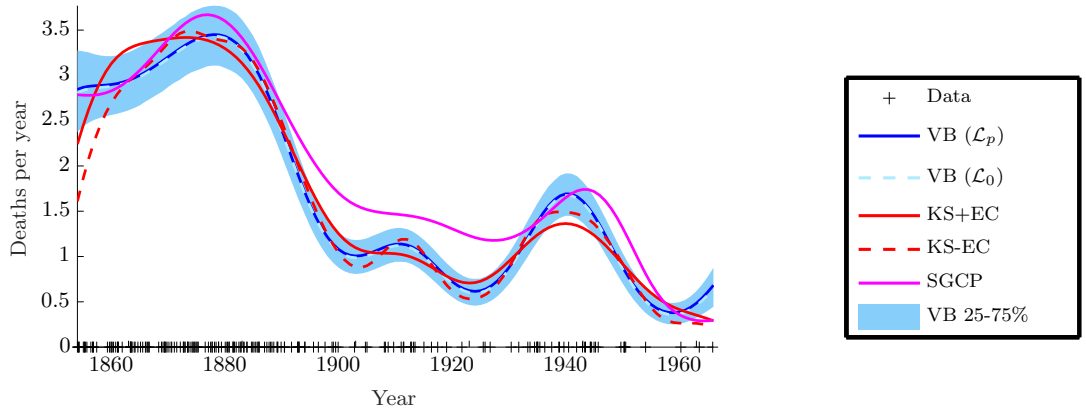


Fig. 3.4 Twitter data.

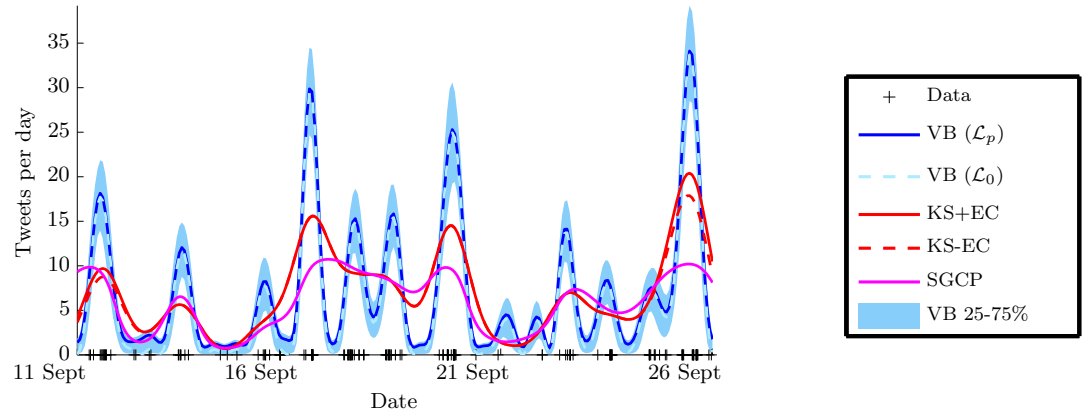


Fig. 3.5 Twitter data (Z on a fixed grid): The difference between sampling $\mathscr{M}_0$ and $\mathscr{M}_p$, and the corresponding lower bounds, $\mathscr{L}_0$ and $\mathscr{L}_p$ (see Section 3.4.6). $\mathscr{L}$ represents the variational lower bound achieved at training time: Equation 3.12.

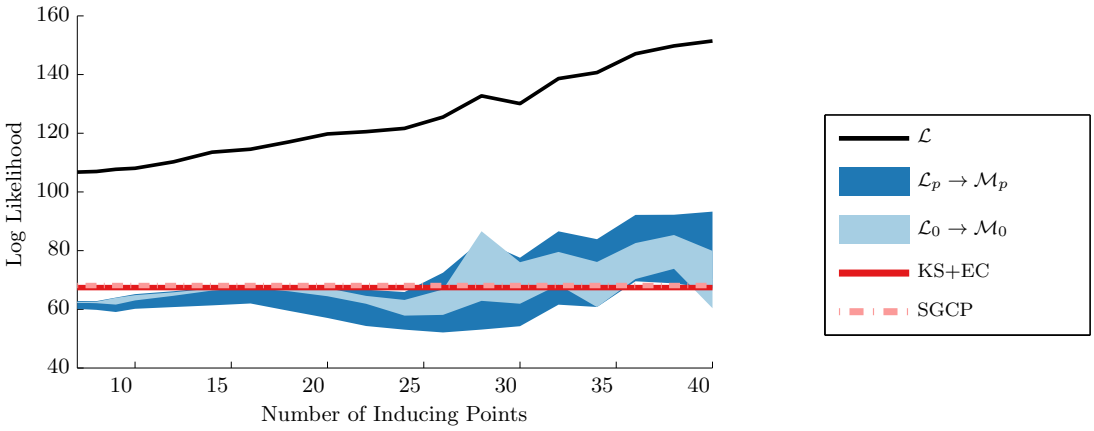
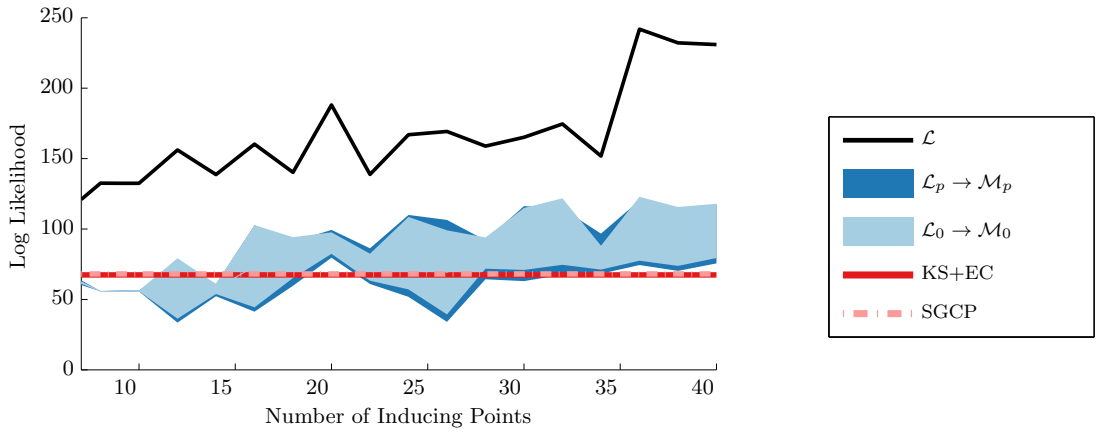


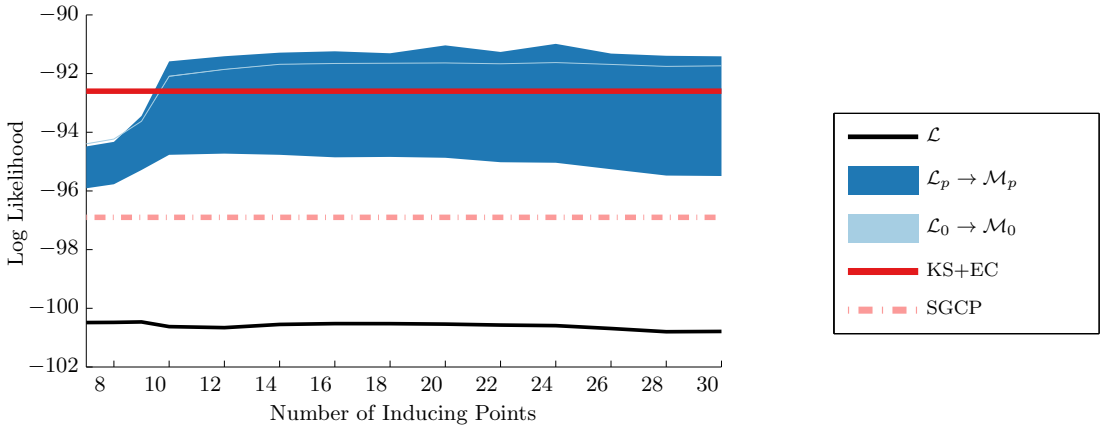
Fig. 3.6 Twitter data (Z optimised): The difference between sampling $\mathscr{M}_0$ and $\mathscr{M}_p$, and the corresponding lower bounds, $\mathscr{L}_0$ and $\mathscr{L}_p$ (see Section 3.4.6). $\mathscr{L}$ represents the variational lower bound achieved at training time: Equation 3.12.



unknown for these datasets we rely on held-out predictive likelihood as the only performance metric.

Coal Mining Disaster Data

Fig. 3.7 Coal Mining Data set: The difference between sampling $\mathscr{M}_0$ and $\mathscr{M}_p$, and the corresponding lower bounds, $\mathscr{L}_0$ and $\mathscr{L}_p$ (see Section 3.4.6). $\mathscr{L}$ represents the variational lower bound achieved at training time: Equation 3.12. The figure clearly demonstrates the tightness of the $\mathscr{L}_0$ bound.



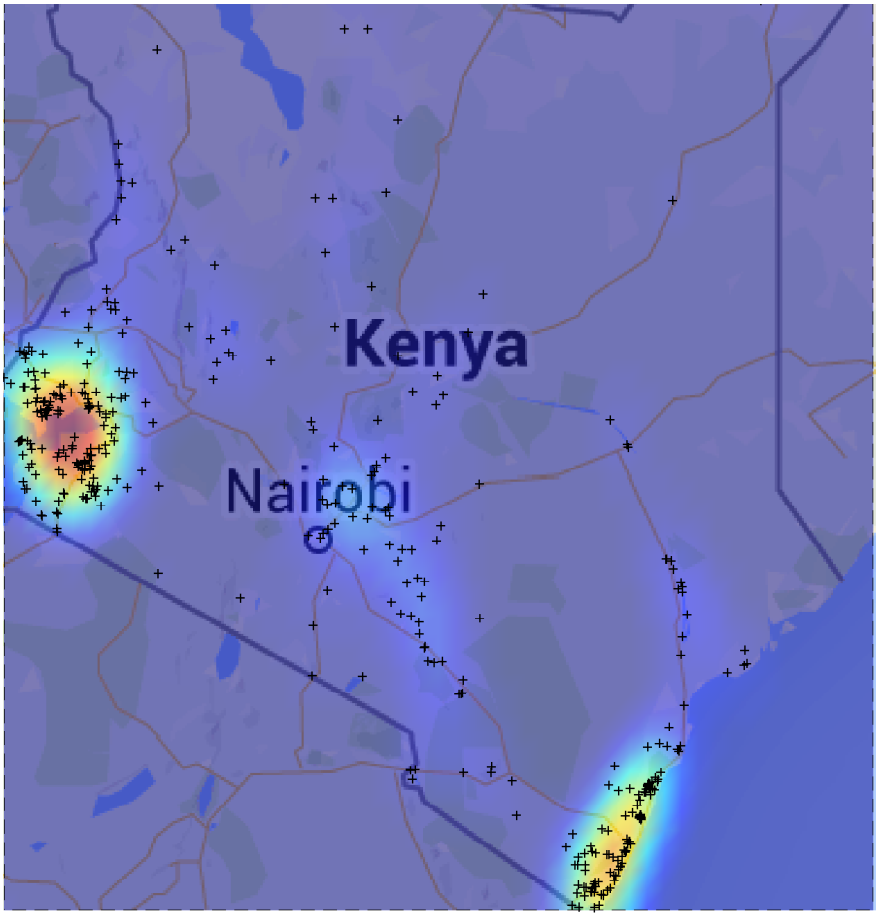
Our first real dataset comprises 190 events recorded from March 15, 1851 to March 22, 1962; each represents a coal-mining disaster that killed at least ten people in the United Kingdom. These data, first analysed in this form in 1979 (Jarrett, 1979), have often been

tackled with nonhomogeneous Poisson processes, (Adams et al., 2009), as the rate of such disasters is expected to vary according to known historical developments. The events are indicated by the rug plot along the axis of Figure 3.3. Our inferred intensity of disasters correlates with the historical introduction of safety regulation, as noted in previous work on this data (Carlin et al., 1992; Fearnhead, 2006). Firstly, our results depict a decline in the rate of such disasters throughout 1870–1890, a period that saw the UK parliament passing several acts with the aim of improving safety for mine workers, including the Coal Mines Regulation Acts of 1872 and 1887. Our inferred intensity also declines after 1950, likely related to the imposition of further safety regulation with the Mines and Quarries Act, 1954. Predictive log-likelihood values on held out data (Figure 3.7) are also encouraging. VBPP outperforms kernel smoothing and SGCP with as few as 10 inducing points; more inducing points yielding no further benefit.

Twitter Data

Next, we ran the models on the tweet profile of the chairman of the 'Better Together Campaign', Alistair Darling, one week either side of the Scottish independence election (189 tweets). Results are shown in Figure 3.4 and Table 3.2, where half the data was held out and a regular 31 point grid was used. Figures 3.5 and 3.6 compare the performance of regularly spaced and optimised inducing points, and show optimisation yields considerably improved performance on this dataset. The $\mathscr{L}_0$ and $\mathscr{L}_p$ bounds become less tight as the number of inducing points is increased, suggesting there is less uncertainty represented in the variational parameter $\mathbf{S}$ and more uncertainty captured by a reduced kernel length scale. This transition is observed for fewer inducing points when inducing point optimisation is employed. Both with and without inducing point optimisation, VBPP $\mathscr{M}_0$ and $\mathscr{M}_p$ outperform both the SGCP and kernel smoothing by a wide margin, suggesting the square link function is an appropriate model for this data.

Fig. 3.8 A sample of 741 malaria incidences in Kenya, which occured over the course of 1985-2010, and the associated VBPP intensity function. 20×20 inducing points.



Malaria Data

We expect that a major application of the contributions presented in this chapter is the joint modelling of disease incidence with correlating factors, in a fully Bayesian, scalable framework. For example, those studying the spread of malaria often wish to use continuous rainfall measurements to better inform their epidemiological models. We use examples from the Malaria Atlas Project (map, 2014) to test our scheme. We extracted 741 incidences of malaria outbreak documented in Kenya between 1985 and 2010, and ran our VBPP algorithm and kernel smoothing on approximately half of the resulting dataset, holding out the

Table 3.2 Run times in seconds for 1D data sets.

Method	**Coal Mining**	**Twitter**
VBPP	0.7	0.5
KS+EC	0.0	0.3
KS-EC	0.0	0.2
SGCP	417.6	230.0

Table 3.3 Test log-likelihood for 2D Malaria data.

KS-EC	**KS+EC**	**VBPP($\mathscr{M}_p$)**	**VBPP($\mathscr{L}_0$)**
855.0	867.2	869.7	855.9

remainder for testing. Test log-likelihood results, given in Tables 3.3, show VBPP performs comparably to kernel smoothing.

3.8 Further Work

Although the performance of the variational Bayesian point process inference algorithm described in this Chapter improves upon standard methods when used in isolation, we expect that it is in its extensions that its utility will be fully realised. In Chapter 2 (Gunter* et al., 2014) we showed that hierarchical modelling of point processes—structured point processes—can significantly improve predictive accuracy. In these multi-output models, statistical strength is shared across multiple rate processes via latent driving processes. The method presented here provides a likelihood model for point-process data that can be incorporated as a probabilistic building-block into these larger interconnected models. That is, our fully generative model can readily be extended to additionally incorporate other observation modalities (not just other Cox processes). For example, real-valued observations such as (log-) household income could be modelled along with the intensity function over crime incidents using a variational multi-output GP framework. In the next chapter we extend this variational inference scheme to cope with multiple, correlated, Cox processes; we in fact solve a particular case of the problem posed in Chapter 2.

Chapter 4

Latent Point Process Allocation

The material in this chapter is based on the following paper:
C. Lloyd, T. Gunter, T. Nickson, M.A. Osborne, and S.J. Roberts. Latent Point Process Allocation (LPPA). In *The 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016,
where our contributions were as outlined in the acknowledgements section of this thesis.

In this chapter we use the efficient variational inference scheme derived in Chapter 3 to revisit something closely related to the dependent point-process model explored in Chapter 2. We show that the model brings a means of incorporating structure in point process inference beyond the state-of-the-art, and does so with significantly lower computational overhead than was previously possible. We furthermore test this as an approach for inferring structure in an event process defined by the arrival of wild birds at discrete geographical feeder locations through continuous time.

4.1 Introduction

In Chapter 2 we saw several examples where modelling multiple dependent Cox processes was desirable, and introduced an appropriate model and associated inference scheme.

In this chapter we extend the variational inference method developed in Chapter 3, allowing it to cope with multi-output structured point processes. In this case, individual output point processes may share similar characteristics (where what is meant by 'similar' is flexible and will be defined later on—cf. Chapter 2). We name this algorithm Latent Poisson Process Allocation (LPPA), alluding to the structural similarities with Latent Dirichlet Allocation (LDA) (Blei et al., 2003), often used to build topic models of document corpora. LDA assumes a document is generated via a mixture of topics, each of which is defined as an atomic distribution over the vocabulary. During inference, each word in a given document is assigned a topic from a set of latent shared topics. Analogously LPPA points are assigned to latent rate functions that are shared across multiple observed point processes. Thus LPPA is conceptually a topic model for Poisson point processes.

LPPA is a continuous analogue of Non-negative Matrix Factorisation (NMF) (Lee and Seung, 2001) and, in particular, bears a resemblance to the fully Bayesian NMF (BNMF) model of Cemgil (2009), since both BNMF and LPPA exploit the infinite divisibility property of the Poisson distribution to apportion data to multiple explanatory factors. However LPPA infers continuous rate functions and benefits from a spatial prior over the continuous latent factors.

In comparison with Chapter 2, we here impose non-negativity constraints on the latent functions themselves (hence the link to NMF), and also in effect assume the observed rate processes are the result of an instantaneous mixing of the latent functions, leaving out the convolutions from our earlier model. Due to this lack of convolution, LPPA is also related to the Semi-Parametric Factor Model (SPFM) for multi-task GP regression (Teh et al., 2005). However, while SPFM uses the Informative Vector Machine (IVM) machinery to achieve computational tractability, LPPA exploits a sparse variational approach related to that found in Hensman et al. (2013). Furthermore, the LPPA offers efficient inference for point processes, enabling it to be applied to problems for which the SPFM is unsuitable.

LPPA, like the Log-Gaussian Cox Process (LGCP) (Møller et al., 1998) and the SGCP (Adams et al., 2009) use transformed Gaussian processes (Rasmussen and Williams, 2006) to construct the prior over the rate function, however LPPA uses the square link function, which results in more tractable integrals.

This work shares a similar motivation to the works of Lian et al. (2014); Miller et al. (2014). Compared to the approach of Miller et al. (2014), LPPA provides a single integrated and generative model for both rate process smoothing and rate process factorisation.

In Lian et al. (2014) we find an approach which is conceptually similar to LPPA. Both methods attempt to factor continuous point processes using a positive linear combination of latent functions. The model used to drive the latent processes and the inference approach employed differ significantly, (binary semi-Markov Jump Processes (BSMJP) and Forward Filtering and Backward Sampling (FFBS) respectively in the case of Lian et al. (2014)), which together would seem to limit the approach to 1-dimensional time series. A GP based intensity function allows LPPA to extend naturally to higher dimensions and mixed-continuous-and-discrete latent co-ordinate spaces.

LPPA, and the associated efforts in our earlier chapters is weakly related to a body of work on cascading-Poisson processes, otherwise known as Hawkes processes, (see for example Iwata et al. (2013), Simma and Jordan (2010) and Linderman and Adams (2014)). In such processes, events trigger spikes in the future intensity function, and the challenge is to determine this intra-temporal structure: LPPA as specified is not capable of modelling these self-excitatory processes.

As we saw in Chapter 2, models which enable efficient joint inference over a set of related Cox processes are applicable in a wide variety of domains. We explore a few applications of LPPA in Section 4.4.

4.1.1 Multivariate Marked Cox Processes

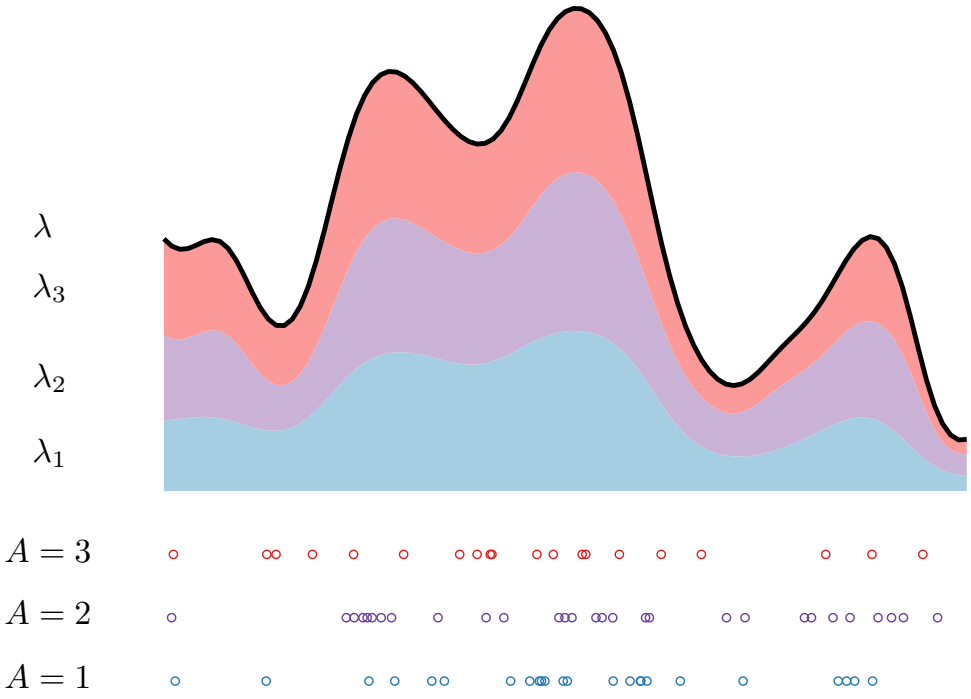
To set the context for our extension into the marked point process domain: formally a Cox process—or doubly stochastic inhomogenous Poisson process—over events $\mathscr{X} \triangleq \{\mathbf{x}^{(n)} \in \mathbb{R}^R\}$ is defined via a stochastic intensity function $\lambda(\mathbf{x}) : \mathbb{R}^R \to \mathbb{R}^+$, with an arbitrary domain of dimension R. The number of points, $N(\mathscr{X}_i)$, found in any subregion $\mathscr{X}_i \subset \mathbb{R}^R$ is Poisson distributed with parameter $\lambda_{\mathscr{X}_i} \triangleq \int_{\mathscr{X}_i} \lambda(\mathbf{x})\,\mathrm{d}\mathbf{x}$—where $\mathrm{d}\mathbf{x}$ indicates integration with respect to the Lebesgue measure over the subregion.

A *marked* Cox process over events $\mathscr{M} \triangleq \left\{ \left(\mathbf{x}^{(n)}, A(\mathbf{x}^{(n)})\right) \mid \mathbf{x}^{(n)} \in \mathbb{R}^R,\ A(\mathbf{x}^{(n)}) \in \mathscr{T} \right\}$ extends a Cox process by associating with each point, $\mathbf{x}$, an additional piece of information $A(\mathbf{x}) \in \mathscr{T}$ which hereon out we refer to as a 'mark'. The form of mark may vary widely; it can be a discrete random variable, real valued random variable or indeed another point process. Furthermore, the mark assigned at any point can depend on the location and the value of the rate function at that location.

We consider a multivariate (discrete) mark set $\mathscr{T} \triangleq \{1, \ldots, T\}$. As the marks are discrete there will be a rate process, $\lambda_t(\mathbf{x})$, associated with each mark t, describing the event of observing a point with mark t. Furthermore, since the set of all points, $\mathscr{M}$, is equal to the union of all sets $\mathscr{M}_t \triangleq \left\{ \left(\mathbf{x}^{(n)}, t\right) \mid \mathbf{x}^{(n)} \in \mathbb{R}^R \right\}$, the overall rate process must be the sum of the individual rate processes. Therefore $\lambda(\mathbf{x}) = \sum_{t=1}^{T} \lambda_t(\mathbf{x})$ and, using the Poisson-multinomial connection, the probability of mark t at a point $\mathbf{x}$ is $p(t;\mathbf{x}) = \lambda_t(\mathbf{x})/\lambda(\mathbf{x})$. In general the individual *latent* rate functions λ_t need not be independent, although in the model we develop in the next section we will assume they are.

In this framework, the probability density of a set of N observed marked points $\mathscr{M} = \{(\mathbf{x}^{(n)}, A^{(n)})\}_{n=1}^{N}$, where $A^{(n)} = A(\mathbf{x}^{(n)})$, in some bounded region, $\mathscr{X}$, factorises conditioned

Fig. 4.1 Rate functions are correlated via a convolution process. Credit to Chris Lloyd for this diagram.



on the rate processes[1]

$$p(\mathscr{M} \mid \lambda_{1:T}) = p(\{\mathbf{x}^{(n)}\}_{n=1}^{N} \mid \lambda_{1:T}) \times p\big(\{A^{(n)}\}_{n=1}^{N} \mid \{\mathbf{x}^{(n)}\}_{n=1}^{N}, \lambda_{1:T}\big). \tag{4.1}$$

The probability density of the points is

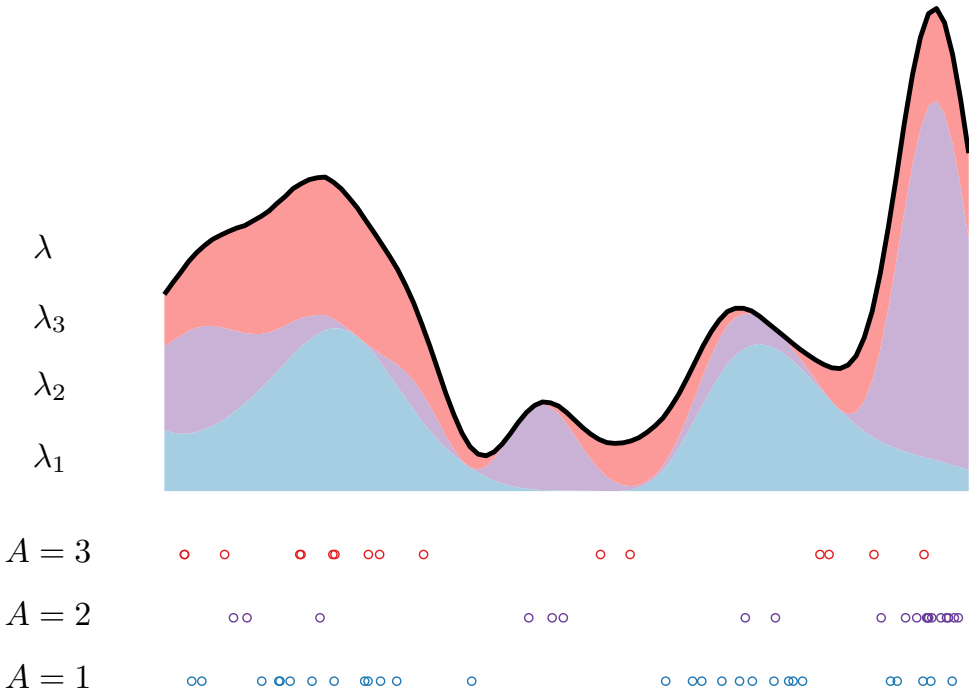
$$p(\{\mathbf{x}^{(n)}\}_{n=1}^{N} \mid \lambda_{1:T}) = \exp\left\{-\int_{\mathscr{X}} \lambda(\mathbf{x})\,\mathrm{d}\mathbf{x}\right\} \prod_{n=1}^{N} \lambda(\mathbf{x}^{(n)}). \tag{4.2}$$

and, using the Poisson-multinomial connection, the probability of the N labels is

$$p\big(\{A^{(n)}\}_{n=1}^{N} \mid \{\mathbf{x}^{(n)}\}_{n=1}^{N}, \lambda_{1:T}\big) = \prod_{n=1}^{N} \frac{\lambda_{A^{(n)}}(\mathbf{x}^{(n)})}{\lambda(\mathbf{x}^{(n)})}. \tag{4.3}$$

[1] We use the MATLAB-like notation $p(a_{1:I} \mid b_{1:J})$ to denote $p(a_1, \ldots, a_I \mid b_1, \ldots, b_J)$.

Fig. 4.2 Rate functions are uncorrelated square-transform Gaussian processes. Credit to Chris Lloyd for this diagram.



A standard inference task for multiple point processes would be to learn the labelling distribution and unobserved point process intensity λ. For independent rate functions, this reduces to simply inferring the rate functions λ_t from subset of points marked with the corresponding mark t.

If the rate functions are not independent, then we may seek to refine our model by building a structural prior which allows statistical strength to be shared between the point processes associated with each label. The latter approach was validated by Gunter* et al. (2014), using a convolution process to tie the rate processes together, as shown in Figure 4.1.

In this work we will assume the marks $\mathcal{T}$ are unobserved latent variables to be inferred. Due to a lack of observability, this is impossible for a single point process. To make this possible, we need to observe multiple point process in which the latent rate functions are present in linearly independent proportions. We will designate each of these point processes a separate *output* yielding a dataset $\mathcal{D}_s \triangleq \{\mathbf{x}^{(s,n)} \in \mathcal{X}\}_{n=1}^{N_s}$ and each of which will be tied

to its own set of output rate functions $\lambda_{s,t}$. Since these outputs are themselves marks in $\mathcal{S} = \{1, \ldots, S\}$, their superposition is a marked point process. There are therefore two distinct sets of marks: observed marks in $\mathcal{S}$ corresponding to the outputs and unobserved marks in $\mathcal{T}$ corresponding to the latent function.

4.1.2 Permanental Point Processes

There are a variety of options for constructing the strictly non-negative stochastic rate function $\lambda(x)$, which drives the point process of a given mark. One common approach is to transform a Gaussian process through a link-function, where typical choices include the exponential (Kom-Samo and Roberts, 2015) and the sigmoid function (Adams et al., 2009). As in the earlier chapters, we use the square transform.

Constructing a rate function as a sum of square transformed, zero-mean independent Gaussian processes results in a particular sub-class of Cox processes known in the mathematical probability literature as *permanental* point processes (Eisenbaum and Kaspi, 2009; Hough et al., 2006).

As the name suggests, positive correlation between point counts in disjoint subsets of the underlying space is produced through defining the likelihood of any given configuration as being proportional to the permanent of a kernel Gram matrix. Computing the permanent is significantly more costly than the $\mathcal{O}(N^3)$ leading-order term associated with the matrix determinant, but fortunately a duality exists between such point processes and a Cox process with intensity function constructed as the sum of independent squared Gaussian processes (Hough et al., 2006), $\lambda(x) = f_1^2(x) + \ldots + f_k^2(x)$, where $f_i(x)$ is a Gaussian process. Looking at Figures 4.1 and 4.2, we can see that this makes intuitive sense: the increased dynamic range afforded by the quadratic transform leads to a better ability to model very high and very low function values, which in turn enables strong local correlations between neighboring subsets of the space—a characteristic of the permanental point process. As a result, we envisage our approach being highly appropriate for performing Bayesian inference on permanental data.

We are further motivated to choose the square transform as a link function for the following reasons: As discussed in Chapter 3, the square transform allows efficient variational Bayesian inference machinery that is entirely tractable—this is not the case for the other two transforms. The square transform spreads prior probability mass more evenly over the set of translation invariant Cox processes than the other two link-functions. This is because as we break the independence assumption inherent in the homogenous Poisson process and instead introduce an increasing amount of positive correlation between disjoint neighbouring subsets of the space, the resulting point process will exhibit ever stronger clustering behaviour for a given configuration of points. In order to model a range of point processes from nearly homogenous through to strongly clustered, we need a link function that results in a transformed prior which has good dynamic range, but can also achieve both very high and very low function values, without breaking dependence between nearby function values. The squared transform has high dynamic range, and furthermore is unique amongst the set of link functions listed in being able to easily model low values—including numerical zero if necessary.

4.2 Model

We will assume the intensity of the t^{th} topic of the s^{th} output is $\lambda_{s,t}(\mathbf{x}) = \gamma_{s,t} f_t^2(\mathbf{x})$, where the functions f_t are independent Gaussian process distributed random functions. We constrain the output length scales $\gamma_{s,t}$ to being positive and thus the rate functions are a positive mixture of positive valued latent functions f_t^2.

We condition each latent function f_t at a set of inducing points $\mathcal{Z} \triangleq \{\mathbf{z}^{(m)} \in \mathcal{X}\}_{m=1}^{M}$ and we denote the evaluation of f_t at these points $\mathbf{u}_t \sim \mathcal{N}(\vec{1}\bar{u}_t, \mathbf{K}_{zz})$. Therefore $f_t|\mathbf{u}_t \sim \mathcal{GP}\big(\mu_t(\mathbf{x}), \Sigma_t(\mathbf{x}, \mathbf{x}')\big)$ is a Gaussian process with mean function $\mu_t(\mathbf{x}) = \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{u}_t$ and covariance function $\Sigma_t(\mathbf{x}, \mathbf{x}') = \mathbf{K}_{xx'} - \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{k}_{zx'}$, where $\mathbf{k}_{xz} = \mathbf{k}_{zx}^{\top}$, $\mathbf{K}_{xx'}$, $\mathbf{K}_{zz}$ are matrices evaluated at $\mathbf{x}$, $\mathbf{x}'$ and $\mathcal{Z}$ using a suitable kernel function. For notational convenience, we

have assumed that all latent functions f_t share the same set of inducing points $\mathcal{Z}$, although it is possible to relax this constraint. As before, we use the exponentiated quadratic (also known as the "squared exponential") Automatic Relevance Determination (ARD) kernel (Duvenaud, 2014), although we once again remind the reader that a variety of integrable ARD kernels may be used as drop in replacements.

Combining the Cox process likelihood, Equation 4.2, and mark likelihood, Equation 4.3, using indicator variables, square transform Gaussian process rate functions and output (square) length scales $\gamma_{s,t}$ gives[2]

$$p(\mathcal{D}_s, A_s \mid f_{1:T}, \Theta) = \prod_t \exp\left(-\int \gamma_{s,t} f_t^2(\mathbf{x}) \mathrm{d}\mathbf{x}\right) \times \prod_n \left[\gamma_{s,t} f_t^2\left(\mathbf{x}^{(s,n)}\right)\right]^{\mathbb{1}\{A_s^{(n)}=t\}}. \tag{4.4}$$

The joint distribution of $\mathcal{D}_{1:S}$, $f_{1:T}$, $\mathbf{u}_{1:T}$ and $A_{1:S}$ in this hierarchy is

$$p(\mathcal{D}_{1:S}, A_{1:S}, f_{1:T}, \mathbf{u}_{1:T} \mid \Theta) = \prod_t p(f_t \mid \mathbf{u}_t)\, p(\mathbf{u}_t) \times \prod_s p(\mathcal{D}_s, A_s \mid f_{1:T}), \tag{4.5}$$

where $\Theta \triangleq \{\Gamma, \alpha_{1:R}, \bar{u}_{1:T}\}$ is the set of model parameters and $\Gamma \in \mathbb{R}_+^{S\times T}$ is a matrix of output length scales with elements $\gamma_{s,t}$. For notational convenience we will often omit conditioning on Θ.

4.3 Variational inference

We roughly follow the process in Chapter 3, allowing us to obtain a lower bound on the model evidence $p(\mathcal{D}_{1:S})$. In this case we order our operations as follows: first integrate out the inducing points $\mathbf{u}_t$ and marginalise the rate functions f_t^2 (Section 4.3.1) to obtain an uncollapsed lower bound; then integrate this uncollapsed bound over the region $\mathcal{X}$ (Section 4.3.2); before collapsing out the indicator variables A_s (Section 4.3.3).

[2]We use $\sum_n$ as shorthand for $\sum_{n=1}^{N_s}$ and $\sum_t$ for $\sum_{t=1}^{T}$ and $\sum_s$ for $\sum_{s=1}^{S}$ and analogously for products.

4.3.1 The Uncollapsed Bound

We begin by integrating out the latent function the variables $\mathbf{u}_{1:T}$ using a variational distribution $q(\mathbf{u}_{1:T}) = \prod_t q(\mathbf{u}_t)$. In contrast to standard variational inference approaches we bring both $q(\mathbf{u}_{1:T})$ and $p(f_{1:T} \mid \mathbf{u}_{1:T})$ outside of the logarithm giving the uncollapsed bound: (also see Appendix A.2):

$$\begin{aligned}\log p(\mathscr{D}_{1:S}, A_{1:S}|\Theta) &= \mathbb{E}_{q(f_{1:T})}\big[\log p(\mathscr{D}_{1:S}, A_{1:S} \mid f_{1:T})\big] \\ &\quad - \mathrm{KL}\big(q(\mathbf{u}_{1:T}) \,\big\|\, p(\mathbf{u}_{1:T})\big) \qquad (4.6)\\ &\triangleq \mathscr{L}(\mathscr{D}_{1:S}, A_{1:S}; \Theta). \qquad (4.7)\end{aligned}$$

Since the likelihood is not directly dependent on $\mathbf{u}_{1:T}$ we can integrate it out before taking expectations. As $p(f_t \mid \mathbf{u}_t)$ and $q(\mathbf{u}_t)$ are conjugate, we can write

$$q(f_t) = \int p(f_t \mid \mathbf{u}_t)\, q(\mathbf{u}_t) \mathrm{d}\mathbf{u}_t = \mathscr{GP}\big(f_t; \tilde{\mu}_t(\mathbf{x}), \tilde{\Sigma}_t(\mathbf{x}, \mathbf{x}')\big), \tag{4.8}$$

where $\tilde{\mu}_t(\mathbf{x}) = \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{m}_t$ and $\tilde{\Sigma}_t(\mathbf{x}, \mathbf{x}') = \mathbf{K}_{xx'} - \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{k}_{zx'} + \mathbf{k}_{xz}\mathbf{K}_{zz}^{-1}\mathbf{S}_t\mathbf{K}_{zz}^{-1}\mathbf{k}_{zx'}$ and $q(f_{1:T}) = \prod_t q(f_t)$. The last term in (4.6) is the Kullback-Leibler divergence between T pairs of independent Gaussian distributions. (Appendix A.3).

We expand Equation 4.6 using Equation 4.4 to give[3]

$$\begin{aligned}\mathbb{E}_{q(f_{1:T})}\big[\log p(\mathscr{D}_{1:S}, A_{1:S} \mid f_{1:T})\big] = \sum_s \Big[&- \sum_t \gamma_{s,t} \int_{\mathscr{X}} \Big(\mathbb{E}_{q(f_t)}[f_{t,x}]^2 + \mathrm{Var}_{q(f_t)}[f_{t,x}]\Big)\mathrm{d}\mathbf{x} \\ &+ \sum_n \sum_t \mathbb{1}\{A_s^{(n)} = t\}\big(\log(\gamma_{s,t}) + \mathbb{E}_{q(f_t)}[\log f_{s,t,n}^2]\big)\Big].\end{aligned} \tag{4.9}$$

[3] We use the following shorthand definitions: $f_{t,x} \triangleq f_t(\mathbf{x})$, $\tilde{\mu}_{t,x} \triangleq \tilde{\mu}_t(\mathbf{x})$, $\tilde{\sigma}_{t,x}^2 \triangleq \tilde{\Sigma}_t(\mathbf{x}, \mathbf{x})$, $f_{s,t,n} \triangleq f_t(\mathbf{x}^{(s,n)})$, $\tilde{\mu}_{s,t,n} \triangleq \tilde{\mu}_t(\mathbf{x}^{(s,n)})$, $\tilde{\sigma}_{s,t,n}^2 \triangleq \tilde{\Sigma}_t(\mathbf{x}^{(s,n)}, \mathbf{x}^{(s,n)})$

The integral $\mathbb{E}_{q(f_t)}[\log f^2_{s,t,n}]$ has an analytic solution

$$\begin{aligned}\mathbb{E}_{q(f_t)}[\log f^2_{s,t,n}] &= -\tilde{G}\left(-\frac{\tilde{\mu}^2_{s,t,n}}{2\tilde{\sigma}^2_{s,t,n}}\right) + \log\left(\frac{\tilde{\sigma}^2_{s,t,n}}{2}\right) - C \\ &= \mathfrak{G}_{s,t,n} \end{aligned} \tag{4.10}$$

where once again $C \approx 0.5772156$ is the Euler-–Mascheroni constant and $\tilde{G}$ is a specialised version of a partial derivative of the confluent hyper-geometric function (Ancarani and Gasaneo, 2008), Appendix A.4.

4.3.2 Integrating over the region $\mathscr{X}$

Equation 4.9 demands the following integral over the region $\mathscr{X}$, where $|\mathscr{X}| = \int_{\mathscr{X}} \mathrm{d}\mathbf{x}$:

$$\int_{\mathscr{X}} \mathbb{E}_{q(f_t)}[f_{t,x}]^2 \mathrm{d}\mathbf{x} \;= \mathbf{m}_t^\top \mathbf{K}_{zz}^{-1} \Psi_{zz} \mathbf{K}_{zz}^{-1} \mathbf{m}_t, \tag{4.11}$$

$$\begin{aligned}\int_{\mathscr{X}} \mathrm{Var}_{q(f_t)}[f_{t,x}]\mathrm{d}\mathbf{x} &= |\mathscr{X}| - \mathrm{Tr}(\mathbf{K}_{zz}^{-1}\Psi_{zz}) \\ &\quad + \mathrm{Tr}(\mathbf{K}_{zz}^{-1}\mathbf{S}_t\mathbf{K}_{zz}^{-1}\Psi_{zz}).\end{aligned} \tag{4.12}$$

For the ARD kernel used in this work the matrix $\Psi(\mathbf{z}, \mathbf{z}') = \int_{\mathscr{X}} K(\mathbf{z}, \mathbf{x})K(\mathbf{x}, \mathbf{z}')\mathrm{d}\mathbf{x}$ is given by:

$$\begin{aligned}\Psi(\mathbf{z}, \mathbf{z}') = \prod_{r=1}^{R} \frac{\sqrt{\pi\alpha_r}}{2} \exp\left(-\frac{(z_r - z'_r)^2}{4\alpha_r}\right) \\ \times \left[\mathrm{erf}\left(\frac{\bar{z}_r - \mathscr{X}_r^{\mathrm{Min}})}{\sqrt{\alpha_r}}\right) - \mathrm{erf}\left(\frac{\bar{z}_r - \mathscr{X}_r^{\mathrm{Max}}}{\sqrt{\alpha_r}}\right)\right],\end{aligned} \tag{4.13}$$

where $\bar{z}_r = \frac{1}{2}(z_r + z'_r)$. Ψ can also be computed for other kernels, including the ARD spectral kernel (Wilson et al., 2014).

4.3.3 Collapsing the Bound

The bound defined by Equation (4.7) contains a large number of multivariate indicator variables $A_s^{(n)}$. The standard variational inference approach to this problem would be to marginalise these variables using a variational distribution $q(A_{1:S})$ and to update $q(\mathbf{u}_{1:T})$, $q(A_{1:S})$ and Θ alternately using co-ordinate ascent 'E' and 'M'-steps. Instead we prefer to collapse out the indicator variables before updating the variational parameters and model parameters through pursuing numerical optimisation of the lower bound. Collapsing out the indicator variables may be particularly desirable in cases where the number of latent functions and observed point processes are large, as we significantly reduce the dimensionality of the resulting optimisation problem.

In general, collapsed variational Bayes has a couple of benefits as compared to the uncollapsed alternative: firstly it reduces the number of variables which must be explicitly updated via a marginal gradient step at each iteration; secondly, as we have analytically marginalised (in this case a large subset of) the unknown variables, the implicit updates of those unknown variables will occur with greater efficiency, in the sense that they will converge to a solution in fewer iterations (Hensman et al., 2012).

To do this we first note that we can write the bound (4.7) as the sum of a set of variables $\mathfrak{A}_{s,t,n} = \log(\gamma_{s,t}) + \mathfrak{G}_{s,t,n}$ which multiply the indicator variables $A_s^{(n)}$ and a term $\mathfrak{B}$, that does not, resulting in the compact definition:

$$\mathscr{L}(\mathscr{D}_{1:S}, A_{1:S}; \Theta) = \mathfrak{B} + \sum_{s,t,n} \mathbb{1}\{A_s^{(n)} = t\}\mathfrak{A}_{s,t,n}. \tag{4.14}$$

To collapse the bound we sum over all the possible assignments to each of the allocation variables [4]:

$$\begin{aligned}\log p(\mathscr{D}_{1:S}|\Theta) &= \log \sum_{A_{1:S}} p(\mathscr{D}_{1:S}, A_{1:S}|\Theta) \\ &\geq \mathfrak{B} + \sum_s \sum_n \log \sum_t \exp \mathfrak{A}_{s,t,n} \\ &\triangleq \mathscr{L}(\mathscr{D}_{1:S}; \Theta)\end{aligned}$$

4.3.4 Kronecker Structure

Since our GP kernel is by necessity separable across input dimensions, e.g. Equation A.1, we can construct our kernel matrices to have Kronecker structure. Such an approach was previously used with Poisson-likelihood GP models by Flaxman et al. (2015), albeit outside of a variational framework. To achieve this structuring we begin by introducing a separate set of inducing points $\mathscr{Z}_r \triangleq \left\{z_r^{(m)} \in [\mathscr{X}_r^{\text{Min}}, \mathscr{X}_r^{\text{Max}}]\right\}_{m=1}^{M_r}$ for each dimension r, so that the overall inducing point set is the cross product of these sets, i.e. $\mathscr{Z} = \mathscr{Z}_1 \times \ldots \times \mathscr{Z}_R$ and $M = \prod_r M_r$.

Using these inducing points we can construct the matrices $\mathbf{K}_{zz}$, Ψ_{zz} as the Kronecker product of R matrices:

$$\mathbf{K}_{zz} = \bigotimes_{r=1}^{R} \mathbf{K}_{z_r z_r}, \qquad \Psi_{zz} = \bigotimes_{r=1}^{R} \Psi_{z_r z_r} \tag{4.15}$$

where the functions used to construct $\mathbf{K}_{z_r z_r}$ and $\Psi_{z_r z_r}$ are the same as Equations A.1 and A.10 without the product over R. We must also give the covariance of the variational distributions, $\mathbf{S}_t$, Kronecker structure:

$$\mathbf{S}_t = \bigotimes_{r=1}^{R} \mathbf{S}_{t,r} \tag{4.16}$$

[4] A more complete derivation is given in Appendix A.6.

We could also similarly structure the means, $\mathbf{m}_t$, however our implementation left these as general full vectors, since we can still exploit Kronecker structure when multiplying Kronecker matrices with full vectors. In fact, we never need to construct any full $M \times M$ matrix to compute the collapsed lower bound, nor the derivatives, and instead we store each of the constituent matrices separately. For example, using straight-forward applications of the Kronecker matrix identities for inversion, multiplication and trace, we can compute the following term of Equation 4.12: $\mathrm{Tr}(\mathbf{K}_{zz}^{-1}\boldsymbol{\Psi}_{zz})$, as

$$\mathrm{Tr}\left(\bigotimes_{r=1}^{R}\mathbf{K}_{z_r z_r}^{-1}\boldsymbol{\Psi}_{z_r z_r}\right) = \prod_r \mathrm{Tr}\left(\mathbf{K}_{z_r z_r}^{-1}\boldsymbol{\Psi}_{z_r z_r}\right),$$

which only requires multiplication and inversion of M_r sized matrices. We also need to maintain Kronecker structure of the cross-kernel terms $\mathbf{k}_{zx} = \mathbf{k}_{xz}^{\top}$:

$$\mathbf{k}_{zx}^{(s,n)} = \bigotimes_{r=1}^{R}\mathbf{k}_{z_r x_r}^{(s,n)}. \tag{4.17}$$

To allow efficient use of low-level matrix libraries, it is important to keep all the constituent vectors $\mathbf{k}_{z_r x_r}^{(s,n)}$ stacked together in contiguous memory as follows:

$$\mathbf{K}_{z_r x_r}^{(s,1:N_s)} = [\mathbf{k}_{z_r x_r}^{(s,1)}, \ldots, \mathbf{k}_{z_r x_r}^{(s,N_s)}]. \tag{4.18}$$

This allows us to compute, for example, $\mathbf{K}_{xz}^{(s,1:N_s)}\mathbf{K}_{zz}^{-1}$ using only R Basic Linear Algebra Subprogram (BLAS) calls rather than $N_s \times R$ calls separately.

Using these Kronecker tricks places two additional constraints on our model: The first is that inducing points cannot be moved independently, we can only control the $z_r^{(m)}$ each of which controls the r^{th} co-ordinate of M/M_r inducing points in $\mathcal{Z}$. The other important consequence is the restriction of $\mathbf{S}_t$ to have Kronecker structure. This results in a necessarily less flexible variational approxmation $q(\mathbf{u}_t)$ than the full matrix equivalent. We can therefore

expect that in general, the tightest variational bound achievable in the full matrix case will always be at least as good as the one arrived at after making the Kronecker approxmations.

4.3.5 Computational Complexity

The computational complexity of LPPA is a function of the total number of data points in all outputs $N = \sum_s N_s$, the number of latent functions T and the number of inducing points M (or for Kronecker structured kernel matrices $\max_r(M_r)$). As can be seen from Equation 4.9, the compuational complexity is linear in N and T. The most significant computational costs are associated with inverting $M \times M$ (or $M_r \times M_r$) matrices, with complexity $\mathcal{O}(M^3)$ if computed via Gauss-Jordan elimination, and with matrix-matrix multiplications with complexity $\mathcal{O}(NM^2)$.

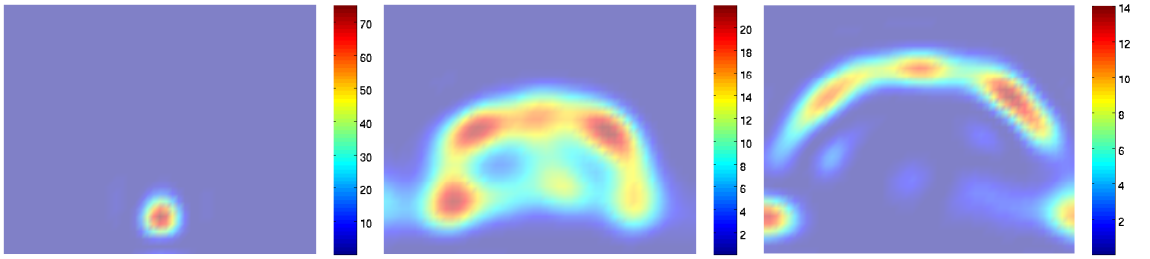
We note that both $\mathbf{K}_{zz}$ and $\mathbf{\Psi}_{zz}$ meet the requirements for inversion and matrix-vector multiplication using the Inverse Fast Multipole Method (IFMM) (Ambikasaran and Darve, 2014)—when computed using translation invariant and smooth covariance functions—which has $\mathcal{O}(M)$ complexity for both operations. However, the headline complexity is still governed by matrix-matrix multiplications of $\mathbf{S}$ in either case and is $\mathcal{O}(TNM^2)$.

4.3.6 Predictive Distribution

To evaluate the performance of LPPA we will follow the example set in Chapter 3 and compute a lower bound on the predictive log-likelihood, $\mathscr{L}_p \leq \log p(\mathscr{H}_{1:S}|\mathscr{D}_{1:S})$, of held out sources with $\mathscr{H}_s = \{\tilde{\mathbf{x}}^{(s,n)}\}_{n=1}^{\tilde{N}_s}$. The derivation of $\mathscr{L}_p$ begins by assuming the posterior distribution of the latent functions at the inducing points, $p(\mathbf{u}_{1:T}|\mathscr{D}_{1:S})$, is well approximated by the optimised variational distribution $q(\mathbf{u}_{1:T})$. The remaining steps follow the derivation of the collapsed bound, however lack the KL term.

When evaluating L_p there are two distinct use cases corresponding to whether we believe the held out data has the same rate as the training data, or whether they merely have the same latent functions f_t^2, albeit in different proportions. The former case corresponds to reusing

Fig. 4.3 Bases computed by LPPA for the Basketball data set (seen in Chapter 2) using 20 players. Note the sparsity of the three factors (due to the non-negativity constraint on the latent functions), and the fact that they strongly conform to three 'modes' of shot typically attempted by players. From left to right: the slam-dunk, the two-pointer shot, and finally the three-point boundary line and corners (we are advised that the corners are tactically advantageous).



the same learned output length scales, Γ, for test, and the latter corresponds to allowing Γ to be adapted, whilst holding the remaining parameters fixed.

4.4 Experiments

We now empirically test the performance of LPPA on several real world datasets.

4.4.1 Benchmark

We benchmark against an algorithm combining Kernel Smoothing (KS) and Poisson-NMF (Lee and Seung, 2001) We first smooth each data set using truncated normal densities to construct a rate process, $\lambda_s(\mathbf{x}) = \sum_{n=1}^{N_s} \mathcal{N}_{\mathcal{X}}(\mathbf{x}; \mathbf{x}^{(s,n)}, \Sigma_s^*)$, for each data source s with diagonal covariances, Σ_s^* fit using leave-one-out cross validation (Hastie et al., 2001). Next we integrate the rate functions, λ_s, over D grid-cells, each of dimension $\Delta\mathbf{x}$, and denote l_{sd} as the result of the integral over the d^{th} cell. We then factorise the matrix $\mathbf{L} \in \mathbb{R}_+^{S\times D}$, with entries l_{sd}, using NMF as $\mathbf{L} \sim \mathcal{P}(\mathbf{AB})$, where $\mathbf{A} \in \mathbb{R}_+^{S\times T}$ is the so-called "activation" matrix, and $\mathbf{B} \in \mathbb{R}_+^{T\times D}$ is the "template" matrix.

The result of this two stage procedure is a predictive rate function that is a positive weighted sum of piece-wise constant functions. We can construct the test log-likelihood

for the held-out datasets as

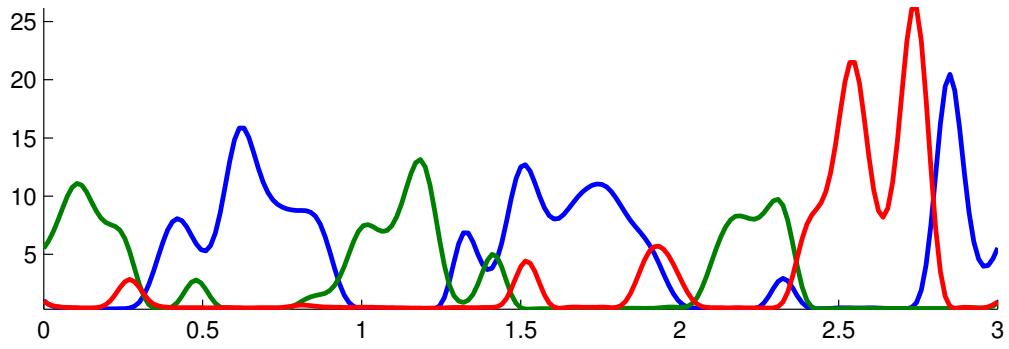
$$\log p(\mathscr{H}_{1:S}|\mathscr{D}_{1:S}, \Sigma^*_{1:S}) = \sum_{s=1}^{S}\sum_{n=1}^{\tilde{N}_h} \log \sum_{t=1}^{T} a_{s,t}\, b_{t,m(h,n)} - |\Delta \mathbf{x}| \sum_{s=1}^{S}\sum_{t=1}^{T}\sum_{b=1}^{B} a_{s,t} b_{t,b} \tag{4.19}$$

where $m(h,n)$ is a function that maps a test data point $\tilde{\mathbf{x}}^{(h,n)}$ into the d^{th} grid-cell.

4.4.2 Twitter Data

Our first application is a simple 1-dimensional time series from Twitter. We pulled a selection of the twitter streams of $\sim$ 50 politicians each from Australia, the U.K. and the U.S. We select a 72-hour time window; from midnight on the second of June 2015 through to midnight on the fourth of June, and log the time stamp of each tweet for each person. The resulting dataset may be viewed as $\sim$ 150 related event-processes. We then employ LPPA to infer the joint underlying structure, and compare against the benchmark.

Fig. 4.4 Bases computed by LPPA for Twitter data.



The three base functions inferred are periodic and it is clear that they do not fully overlap, the phase lag can be attributable to the 8-12 hour time difference between each of the three time-zones. The predictive log-likelihood performance, Table 4.1, is very strong as compared to the competing methods.

Table 4.1 Twitter held out predictive log-likelihoods.

LPPA ($\mathscr{L}_p$)	KS+NMF	IND. KS
-209	-363	-6853

4.4.3 Basketball Data

For our next experiment we re-investigate the basketball point shot location problem seen in Chapter 2, and also previously analysed by Miller et al. (2014) and Gunter* et al. (2014). Although each individual shot is made from a unique location, the rules of basketball make certain areas on the court more strategic than others. In point process language, we might suggest that the point shot intensity function will vary over the space according to some notional basketball utility function, where shots are taken so as to maximise the long term expected score. This is clearly visible in Figure 4.3.

As described in Section 4.3.6 we can consider two use cases: 1) The common rate (CR) case, which corresponds to the prediction of a held out set of shot data from players used to train the model, (i.e. held out shot locations), and 2), the common topic (CT) case, corresponds to the construction of rate processes for players not previously seen using new output length scales (or weight matrix $\mathbf{A}$ for the benchmark), learned independently via convex optimisation. In other words 1) corresponds to same set of players, missing shot data, and 2) corresponds to entirely new player.

We selected 40 players at random and $\sim$25 shots per player, from which we created 10 test/train splits of equal size. We used three latent functions and ran 250 iterations of gradient descent on the LPPA model, at approximately 1 iteration per second on a fairly standard desktop computer. The inducing point were fixed to an evenly spaced 13$\times$17 grid. We used the same number of grid cells for the benchmark. We then averaged the predictive performance over the 10 splits. The results are shown in Table 4.2.

Table 4.2 Basketball held out predictive log-likelihoods.

LPPA($\mathscr{L}_p$)		KS+NMF		IND. KS
CR	CT	CR	CT	CR
-3673.9	**-3651.9**	-3935.2	-3983.6	-4940.5

4.4.4 Wild Bird Data

In this experiment we use the wild bird dataset previously investigated by Psorakis et al. (2012). The data set contains the times tagged wild birds arrive at a number of Radio Frequency Identification (RFID) equipped bird feeders distributed across Wytham Great Wood, near Oxford. The dataset contains hundreds birds and hundreds of thousands of arrivals at dozens of locations shown in Figure 4.5. We selected a subset of the data containing 14,742 arrivals by 274 birds at 37 locations over a 7 day period.

We model each bird as a separate output and latent functions are defined over a mixed continuous discrete co-ordinate space consisting of arrival time and feeder ID. Although the location identifiers are discrete, the kernel between feeders reflect their geographic proximity.[5] We used six of these continuous-discrete latent functions, which are shown in Figure 4.6.

Factorisation reveals likely communities of birds, each of which has a distinct arrival intensity for a given feeder. We found that most birds attached strongly to one latent function (which is interpretable as a 'community' (Psorakis et al., 2012)), potentially indicating that much of the community structure has been captured.

4.5 Conclusion

We have presented a Bayesian factor model for continuous Poisson process intensities, together with an efficient variational inference engine. The approach yields sparse, smooth

[5]The presence of this discrete dimension means that one of the integrals used to compute Equation A.10 becomes a finite sum. See Appendix A.8.

Fig. 4.5 Feeding stations in Wytham Great Wood.

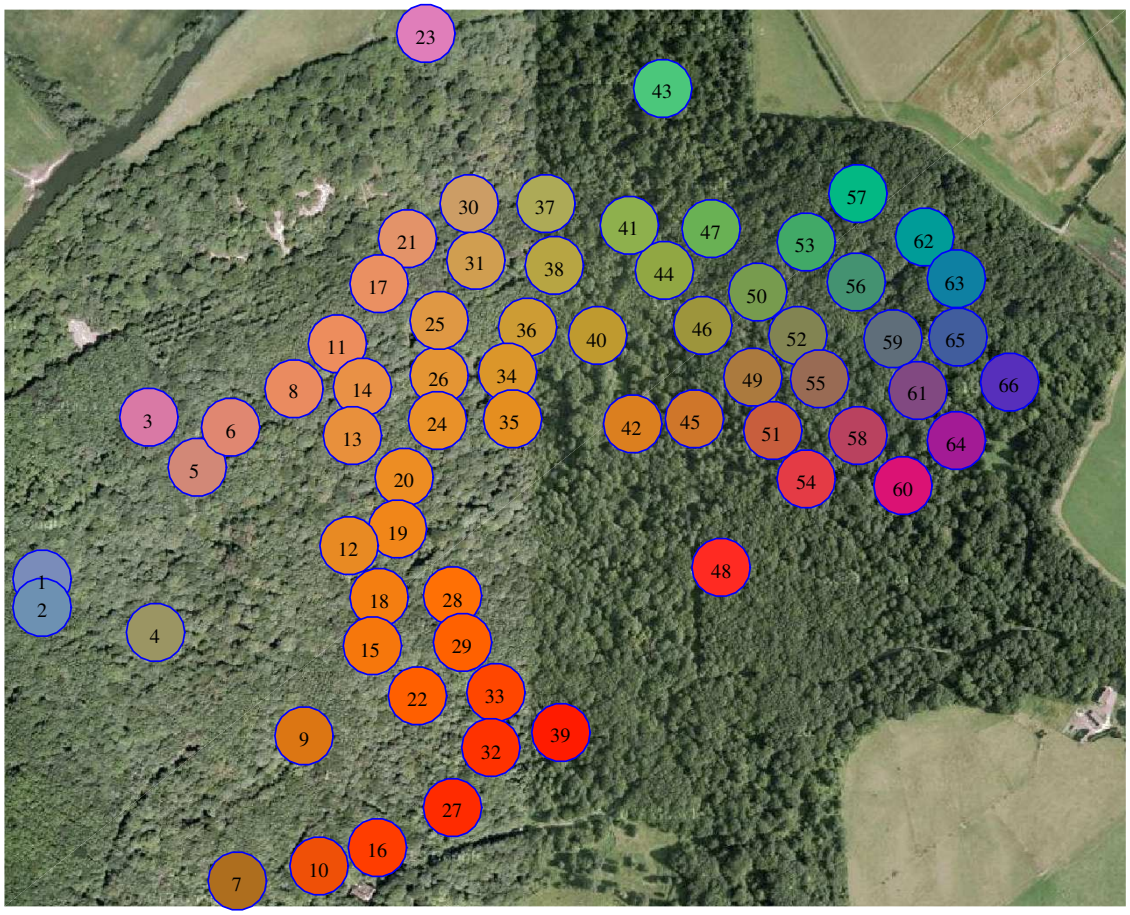
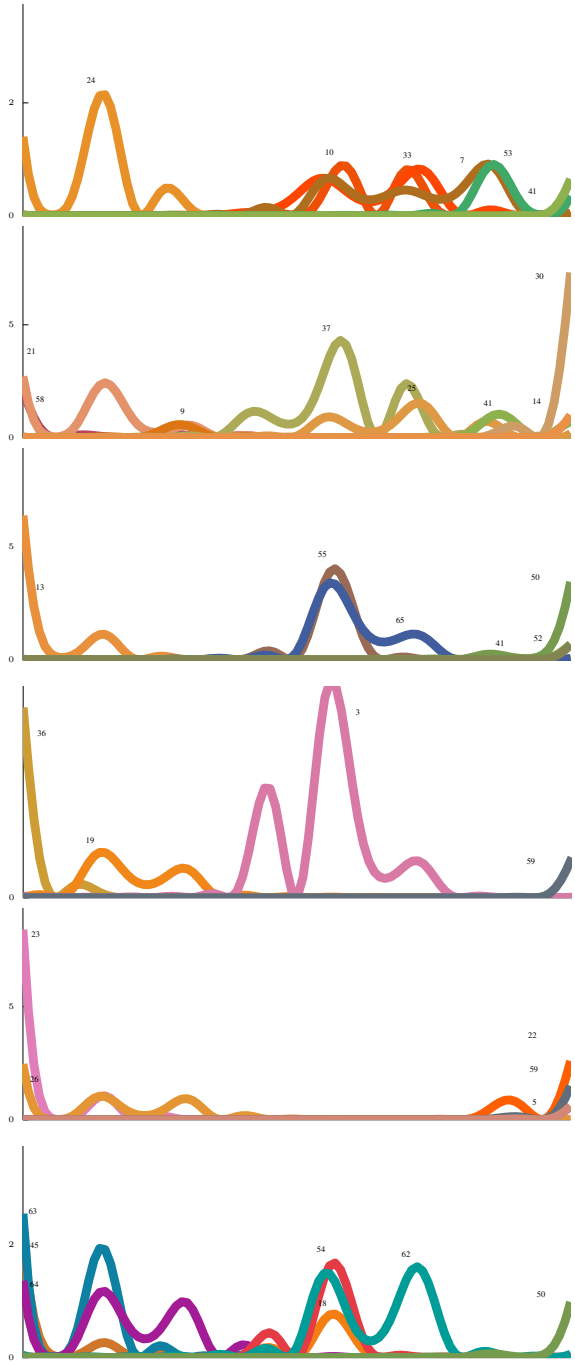


Fig. 4.6 Bases computed by LPPA for the Wild Bird dataset. Each sub-plot represents the spatio-temporal distribution of putative communities of birds. The numbers and colour coding correspond those in Figure 4.5.



and interpretable latent factors. We have demonstrated the validity and usefulness of the model on real-world datasets.

This work has several possible extensions and there are numerous other practical applications worthy of further exploration. These include the following:

Modelling pairwise communications: communications between people or computers is a dynamic network in which the link is present during the instant the communication is active. This can be thought as a point process in a continuous temporal, or spatio-temporal domain. In many case we may expect there to be shared common structure between the activities on each link, based on the community structure of the senders and receivers. With a few modifications LPPA could be adapted for this purpose.

Mixed likelihoods: there are many important applications in which one may want to jointly model continuous, point process and binary data.

Chapter 5

Fast Active Bayesian Quadrature

The material in this chapter is based on the following paper:
T. Gunter, M. A. Osborne, R. Garnett, P. Hennig, and S.J. Roberts. Sampling for inference in probabilistic models with fast Bayesian quadrature. In C. Cortes and N. Lawrence, editors, *Advances in Neural Information Processing Systems (NIPS)*, 2014,
where our contributions were as outlined in the acknowledgements section of this thesis.

Referring back to the introduction, and the theme of information efficiency carried throughout this thesis, in this chapter we make a contribution to the body of work known as Bayesian computation. Specifically, we propose a novel sampling framework for inference in probabilistic models: an active learning approach that converges more quickly (in wall-clock time) than MCMC benchmarks. The central challenge in probabilistic inference is numerical integration, to average over ensembles of models or unknown (hyper-)parameters (for example to compute the marginal likelihood or a partition function). MCMC has provided approaches to numerical integration that deliver state-of-the-art inference, but can suffer from sample inefficiency and poor convergence diagnostics. Bayesian quadrature techniques offer a model-based solution to such problems, but their uptake has been hindered by prohibitive computation costs. We introduce a warped model for probabilistic integrands (likelihoods) that are known to be non-negative, permitting a cheap active learning scheme to optimally

select sample locations. Our algorithm is demonstrated to offer faster convergence (in wall-clock time) relative to simple Monte Carlo and annealed importance sampling on both synthetic and real-world examples.

5.1 Introduction

As we have seen in earlier chapters, Bayesian approaches to machine learning problems inevitably call for the frequent approximation of computationally intractable integrals of the form

$$Z = \langle \ell \rangle = \int \ell(\mathbf{x})\, \pi(\mathbf{x})\, \mathrm{d}\mathbf{x}, \tag{5.1}$$

where both the likelihood $\ell(\mathbf{x})$ and prior $\pi(\mathbf{x})$ are non-negative. Such integrals arise when marginalising over model parameters or variables, calculating predictive test likelihoods and computing model evidences. In all cases the function to be integrated—the integrand—is naturally constrained to be non-negative, as the functions being considered define probabilities.

While the techniques developed here are naturally applicable to any problem where the integrand is constrained to be non-negative, in what follows we will primarily consider the computation of model evidence, Z. In this case, $\ell(\mathbf{x})$ defines the unnormalised likelihood over a D-dimensional parameter set, $x_1, ..., x_D$, and $\pi(\mathbf{x})$ defines a prior density over $\mathbf{x}$. Many techniques exist for estimating Z, such as annealed importance sampling AIS (Neal, 2001), nested sampling (Skilling, 2004), and bridge sampling (Meng and Wong, 1996). These approaches are based around a core Monte Carlo estimator for the integral, and make minimal effort to exploit prior information about the likelihood surface. Monte Carlo convergence diagnostics are also unreliable for partition function estimates (Brooks and Roberts, 1998; Cowles et al., 1999; Neal, 1993). More advanced methods—e.g., AIS—also require parameter tuning, and will yield poor estimates with misspecified parameters.

The Bayesian Quadrature (BQ) (Diaconis, 1988; Kennedy, 1998; O'Hagan, 1991; Rasmussen and Ghahramani, 2003) approach to estimating model evidence is inherently model based. That is, it involves specifying a prior distribution over likelihood functions in the form of a GP (Rasmussen and Williams, 2006). This prior may be used to encode beliefs about the likelihood surface, such as smoothness or periodicity. Given a set of samples from $\ell(\mathbf{x})$, posteriors over both the integrand and the integral may in some cases be computed analytically (see below for discussion on other generalisations). Active sampling (Osborne et al., 2012a) can then be used to select function evaluations so as to maximise the reduction in entropy of either the integrand or integral. Such an approach has been demonstrated to improve sample efficiency, relative to naïve randomised sampling (Osborne et al., 2012a).

In a big-data setting, where likelihood function evaluations are prohibitively expensive, BQ is demonstrably better than Monte Carlo approaches (Osborne et al., 2012a; Rasmussen and Ghahramani, 2003). As the cost of the likelihood decreases, however, BQ no longer achieves a higher effective sample rate per second, because the computational cost of maintaining the GP model and active sampling becomes relevant, and many Monte Carlo samples may be generated for each new BQ sample. Our goal was to develop a cheap and accurate BQ model alongside an efficient active sampling scheme, such that even for low cost likelihoods BQ would be the scheme of choice. Our contributions extend existing work in two ways:

Square-root GP: Foundational work (Diaconis, 1988; Kennedy, 1998; O'Hagan, 1991; Rasmussen and Ghahramani, 2003) on BQ employed a GP prior directly on the likelihood function, making no attempt to enforce non-negativity a priori. Osborne et al. (2012a) introduced an approximate means of modelling the logarithm of the integrand with a GP (whilst also sampling actively). This involved making a first-order approximation to the exponential function, so as to maintain tractability of inference in the integrand model. In this work, we choose another classical transformation to preserve non-negativity—the square-root. By placing a GP prior on the square-root of the integrand, we arrive at a model which both goes

some way towards dealing with the high dynamic range of most likelihoods, and enforces non-negativity without the approximations resorted to in Osborne et al. (2012a).

Fast Active Sampling: Whereas most approaches to BQ use either a randomised or fixed sampling scheme, Osborne et al. (2012a) targeted the reduction in the expected variance of Z. Here, we sample where the expected posterior variance of the integrand after the quadratic transform is at a maximum. This is a cheap way of balancing exploitation of known probability mass and exploration of the space in order to approximately minimise the entropy of the integral.

We compare our approach, termed Warped Sequential Active Bayesian Integration (WSABI), to non-negative integration with standard Monte Carlo techniques on simulated and real examples. Crucially, we make comparisons of error against ground truth *given a fixed compute budget*.

5.2 Bayesian Quadrature

Given a non analytic integral $\langle \ell \rangle \coloneqq \int \ell(\mathbf{x})\pi(\mathbf{x})\,\mathrm{d}\mathbf{x}$ on a domain $\mathcal{X} = \mathbb{R}^D$, Bayesian quadrature is a model based approach of inferring both the functional form of the integrand and the value of the integral conditioned on a set of sample points. Typically the prior density is assumed to be a Gaussian, $\pi(\mathbf{x}) \coloneqq \mathcal{N}(\mathbf{x}; \boldsymbol{\nu}, \boldsymbol{\Lambda})$; however, via the use of an importance reweighting trick, $q(\mathbf{x}) = (q(\mathbf{x})/\pi(\mathbf{x}))\,\pi(\mathbf{x})$, any prior density $q(\mathbf{x})$ may be integrated against. For clarity we will henceforth notationally consider only the $\mathcal{X} = \mathbb{R}$ case, although all results trivially extend to $\mathcal{X} = \mathbb{R}^d$.

Typically a GP prior is chosen for $\ell(x)$, although it may also be directly specified on $\ell(x)\pi(x)$. This is parameterised by a mean $\mu(x)$ and scaled Gaussian covariance $K(x, x') \coloneqq \lambda^2 \exp\left(-\frac{1}{2}\frac{(x-x')^2}{\sigma^2}\right)$. The output length-scale λ and input length-scale σ control the standard deviation of the output and the autocorrelation range of each function evaluation respectively, and will be jointly denoted as $\theta = \{\lambda, \sigma\}$. Conditioned on samples $x_d = \{x_1, ..., x_N\}$ and

associated function values $\ell(x_d)$, the posterior mean is

$$m_{\mathscr{D}}(x) \coloneqq \mu(x) + K(x, x_d)K^{-1}(x_d, x_d)\big(\ell(x_d) - \mu(x_d)\big), \tag{5.2}$$

and the posterior covariance is

$$C_{\mathscr{D}}(x, x') \coloneqq K(x, x) - K(x, x_d)K(x_d, x_d)^{-1}K(x_d, x), \tag{5.3}$$

where $\mathscr{D} \coloneqq \big\{x_d, \ell(x_d), \theta\big\}$.

When a GP prior is placed directly on the integrand in this manner, the posterior mean and variance of the integral can be derived analytically through the use of Gaussian identities, as in Rasmussen and Ghahramani (2003). This is because the integration is a linear projection of the function posterior onto $\pi(x)$, and joint Gaussianity is preserved through any arbitrary affine transformation. The mean and variance estimate of the integral are given as follows:

$$\mathbb{E}_{\ell|\mathscr{D}}\big[\langle\ell\rangle\big] = \int m_{\mathscr{D}}(x)\,\pi(x)\,\mathrm{d}x, \tag{5.4}$$

and,

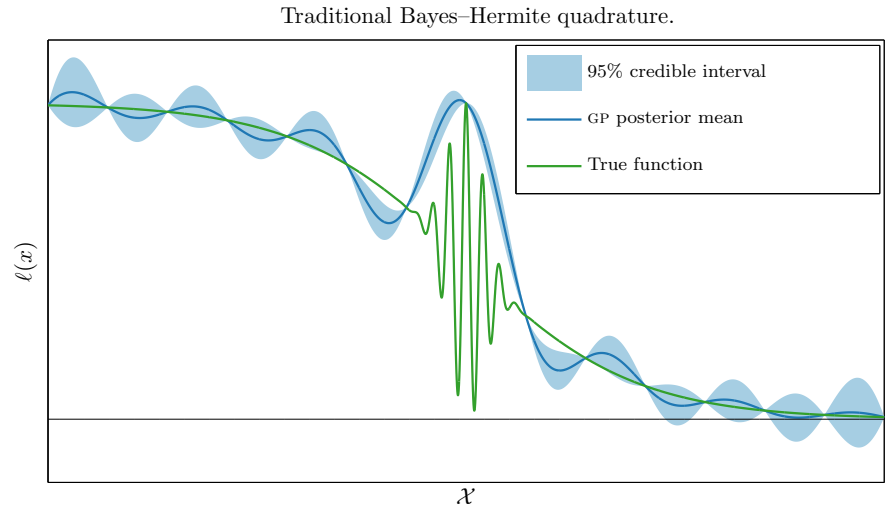
$$\mathbb{V}_{\ell|\mathscr{D}}\big[\langle\ell\rangle\big] = \iint C_{\mathscr{D}}(x, x')\,\pi(x)\,\mathrm{d}x\,\pi(x')\,\mathrm{d}x'. \tag{5.5}$$

Both mean and variance are analytic when $\pi(x)$ is Gaussian, a mixture of Gaussians, or a polynomial (amongst other functional forms).

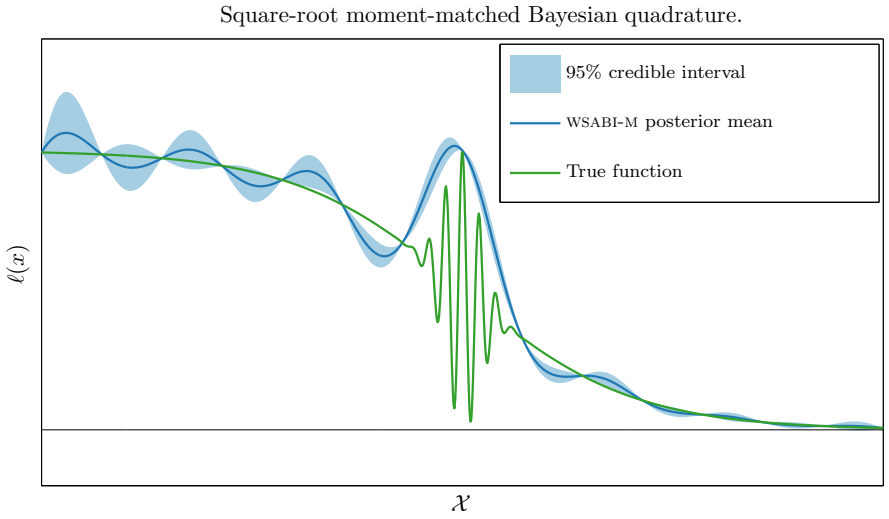
If the GP prior is placed directly on the likelihood in the style of traditional Bayes–Hermite quadrature, the optimal point to add a sample (from an information gain perspective) is dependent only on x_d—the locations of the previously sampled points. This means that given a budget of N samples, the most informative set of function evaluations is a design that can be pre-computed, completely uninfluenced by any information gleaned from function values (Minka, 2000). In Osborne et al. (2012a), where the log-likelihood is modelled by a GP,

a dependency is introduced between the uncertainty over the function at any point and the function value at that point. This means that the optimal sample placement is now directly influenced by the obtained function values.

Fig. 5.1 Figure 5.1a depicts the integrand as modelled directly by a GP, conditioned on 15 samples selected on a grid over the domain. Figure 5.1b shows the moment-matched approximation—note the larger relative posterior variance in areas where the function is high. The linearised square-root GP performed identically on this example, and is not shown.



(a)



(b)

An illustration of Bayes–Hermite quadrature is given in Figure 5.1a. Conditioned on a grid of 15 samples, it is visible that any sample located equidistant from two others is

equally informative in reducing our uncertainty about $\ell(x)$. As the dimensionality of the space increases, exploration can be increasingly difficult due to the curse of dimensionality. A better designed BQ strategy would create a dependency structure between function value and informativeness of sample, in such a way as to appropriately express prior bias towards exploitation of existing probability mass.

5.3 Square-Root Bayesian Quadrature

Crucially, likelihoods are non-negative. This is a fact neglected by traditional Bayes–Hermite quadrature. In Osborne et al. (2012a) the logarithm of the likelihood was modelled, and the authors approximated the posterior of the integral via a linearisation trick. We choose a different member of the power transform family—the square-root.

The square-root transform halves the dynamic range of the function we model. This helps deal with the large variations in likelihood observed in a typical model, and has the added benefit of extending the autocorrelation range (or the input length-scale) of the GP, yielding improved predictive power when extrapolating away from existing sample points.

Let $\tilde{\ell}(x) \coloneqq \sqrt{2\big(\ell(x)-\alpha\big)}$, such that $\ell(x) = \alpha + \frac{1}{2}\tilde{\ell}(x)^2$, where α is a small positive scalar.[1] We then take a GP prior on $\tilde{\ell}(x)$: $\tilde{\ell} \sim \mathcal{GP}(0, K)$. We can then write the posterior for $\tilde{\ell}$ as

$$p(\tilde{\ell} \mid \mathcal{D}) = \mathcal{GP}\big(\tilde{\ell}; \tilde{m}_{\mathcal{D}}(\cdot), \tilde{C}_{\mathcal{D}}(\cdot,\cdot)\big); \tag{5.6}$$

$$\tilde{m}_{\mathcal{D}}(x) \coloneqq K(x, x_d)K(x_d, x_d)^{-1}\tilde{\ell}(x_d); \tag{5.7}$$

$$\tilde{C}_{\mathcal{D}}(x, x') \coloneqq K(x, x') - K(x, x_d)K(x_d, x_d)^{-1}K(x_d, x'). \tag{5.8}$$

The square-root transformation renders analysis intractable with this GP: we arrive at a process whose marginal distribution for any $\ell(x)$ is a non-central χ^2 (with one degree of free-

[1] α was taken as $0.8\times\min \ell(x_d)$ in all experiments; our investigations found that performance was insensitive to the choice of this parameter.

dom). Given this process, the posterior for our integral is not closed-form. We now describe two alternative approximation schemes to resolve this problem.

5.3.1 Linearisation

We firstly consider a local linearisation of the transform $f : \tilde{\ell} \mapsto \ell = \alpha + \frac{1}{2}\tilde{\ell}^2$. As GPs are closed under linear transformations, this linearisation will ensure that we arrive at a GP for ℓ given our existing GP on $\tilde{\ell}$. Generically, if we linearise around $\tilde{\ell}_0$, we have $\ell \simeq f(\tilde{\ell}_0) + f'(\tilde{\ell}_0)(\tilde{\ell} - \tilde{\ell}_0)$. Note that $f'(\tilde{\ell}) = \tilde{\ell}$: this simple gradient is a further motivation for our transform, as described further in Section 5.3.3. We choose $\tilde{\ell}_0 = \tilde{m}_{\mathcal{D}}$; this represents the mode of $p(\tilde{\ell} \mid \mathcal{D})$. Hence we arrive at

$$\ell(x) \simeq \left(\alpha + \tfrac{1}{2}\tilde{m}_{\mathcal{D}}(x)^2\right) + \tilde{m}_{\mathcal{D}}(x)\left(\tilde{\ell}(x) - \tilde{m}_{\mathcal{D}}(x)\right) = \alpha - \tfrac{1}{2}\tilde{m}_{\mathcal{D}}(x)^2 + \tilde{m}_{\mathcal{D}}(x)\,\tilde{\ell}(x). \quad (5.9)$$

Under this approximation, in which ℓ is a simple affine transformation of $\tilde{\ell}$, we have

$$p(\ell \mid \mathcal{D}) \simeq \mathcal{GP}\left(\ell; m^{\mathcal{L}}_{\mathcal{D}}(\cdot), C^{\mathcal{L}}_{\mathcal{D}}(\cdot,\cdot)\right); \quad (5.10)$$

$$m^{\mathcal{L}}_{\mathcal{D}}(x) \coloneqq \alpha + \tfrac{1}{2}\tilde{m}_{\mathcal{D}}(x)^2; \quad (5.11)$$

$$C^{\mathcal{L}}_{\mathcal{D}}(x,x') \coloneqq \tilde{m}_{\mathcal{D}}(x)\tilde{C}_{\mathcal{D}}(x,x')\tilde{m}_{\mathcal{D}}(x'). \quad (5.12)$$

5.3.2 Moment Matching

Alternatively, we consider a moment-matching approximation: $p(\ell \mid \mathcal{D})$ is approximated as a GP with mean and covariance equal to those of the true χ^2 (process) posterior. This gives $p(\ell \mid \mathcal{D}) \coloneqq \mathcal{GP}\left(\ell; m^{\mathcal{M}}_{\mathcal{D}}(\cdot), C^{\mathcal{M}}_{\mathcal{D}}(\cdot,\cdot)\right)$, where

$$m^{\mathcal{M}}_{\mathcal{D}}(x) \coloneqq \alpha + \tfrac{1}{2}\left(\tilde{m}^2_{\mathcal{D}}(x) + \tilde{C}_{\mathcal{D}}(x,x)\right); \quad (5.13)$$

$$C^{\mathcal{M}}_{\mathcal{D}}(x,x') \coloneqq \tfrac{1}{2}\tilde{C}_{\mathcal{D}}(x,x')^2 + \tilde{m}_{\mathcal{D}}(x)\tilde{C}_{\mathcal{D}}(x,x')\tilde{m}_{\mathcal{D}}(x'). \quad (5.14)$$

We will call these two approximations WSABI-L (for "linear") and WSABI-M (for "moment-matched"), respectively. Figure 5.2 shows a comparison of the approximations on synthetic data. The likelihood function, $\ell(x)$, was defined to be $\ell(x) = \exp(-x^2)$, and is plotted in red. We placed a GP prior on $\tilde{\ell}$ and conditioned this on seven observations spanning the interval $[-2, 2]$. We then drew 50 000 samples from the true χ^2 posterior on $\tilde{\ell}$ along a dense grid on the interval $[-5, 5]$ and used these to estimate the true density of $\ell(x)$, shown in blue shading. Finally, we plot the means and 95% confidence intervals for the approximate posterior. Notice that the moment-matching results in a higher mean and variance far from observations, but otherwise the approximations largely agree with each other and the true density.

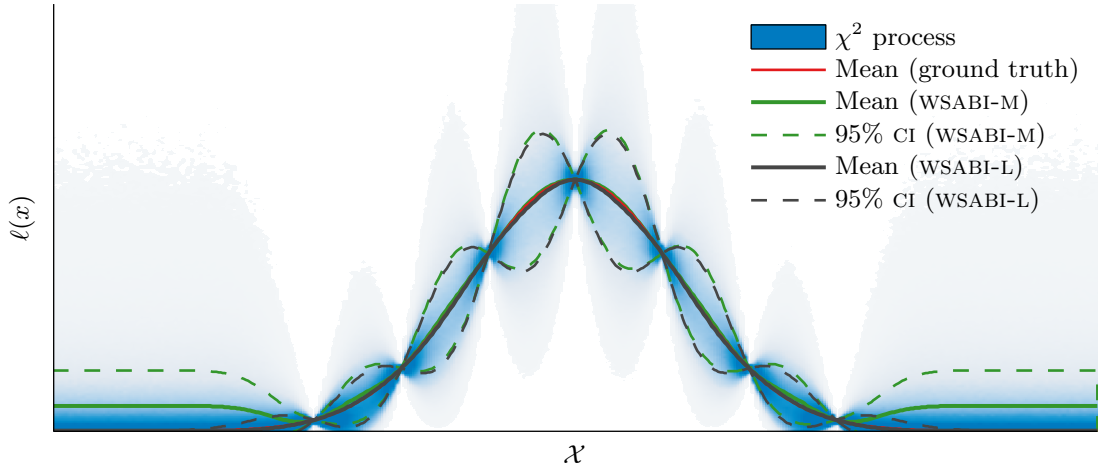
5.3.3 Quadrature

As we have assumed a zero mean function $\mu(x) = 0$ and a scaled Gaussian covariance function $K(x, x') \coloneqq \lambda^2 \exp\left(-\frac{1}{2}\frac{(x-x')^2}{\sigma^2}\right)$, $\tilde{m}_{\mathcal{D}}$ and $\tilde{C}_{\mathcal{D}}$ are both mixtures of un-normalised Gaussians K. As such, the expressions for posterior mean and covariance under either the linearisation ($m^{\mathscr{L}}_{\mathcal{D}}$ and $C^{\mathscr{L}}_{\mathcal{D}}$, respectively) or the moment-matching approximations ($m^{\mathscr{M}}_{\mathcal{D}}$ and $C^{\mathscr{M}}_{\mathcal{D}}$, respectively) are also mixtures of un-normalised Gaussians. Substituting these expressions (under either approximation) into (5.4) and (5.5) yields closed-form expressions (omitted due to their length) for the mean and variance of the integral $\langle \ell \rangle$. This result motivated our initial choice of transform: for linearisation, for example, it was only the fact that the gradient $f'(\tilde{\ell}) = \tilde{\ell}$ that rendered the covariance in (5.12) a mixture of un-normalised Gaussians. The discussion that follows is equally applicable to either approximation.

It is clear that the posterior variance of the likelihood model is now a function of both the expected value of the likelihood at that point, and the distance of that sample location from the rest of x_d. This is visualised in Figure 5.1b.

Comparing Figures 5.1a and 5.1b we see that conditioned on an identical set of samples, WSABI both achieves a closer fit to the true underlying function, and associates minimal prob-

Fig. 5.2 The χ^2 process, alongside moment-matched (WSABI-M) and linearised approximations (WSABI-L). Notice that the WSABI-L mean is nearly identical to the ground truth. Credit to Roman Garnett for providing this figure.



ability mass with negative function values. These are desirable properties when modelling likelihood functions—both arising from the use of the square-root transform.

5.4 Active Sampling

Given a full Bayesian model of the likelihood surface, it is natural to call on the framework of Bayesian decision theory, selecting the next function evaluation so as to optimally reduce our uncertainty about either the total integrand surface or the integral. Let us define this next sample location to be x_*, and the associated likelihood to be $\ell_* := \ell(x_*)$. Two utility functions immediately present themselves as natural choices, which we consider below. Both options are appropriate for either of the approximations to $p(\ell)$ described above.

5.4.1 Minimising Expected Entropy

One possibility would be to follow (Osborne et al., 2012a) in minimising the expected entropy of the integral, by selecting $x_* = \arg\min_x \langle \mathbb{V}_{\ell|\mathcal{D},\ell(x)}\left[\langle \ell \rangle\right] \rangle$, where

$$\left\langle \mathbb{V}_{\ell|\mathcal{D},\ell(x)}\left[\langle \ell \rangle\right] \right\rangle = \int \mathbb{V}_{\ell|\mathcal{D},\ell(x)}\left[\langle \ell \rangle\right] \mathcal{N}\left(\ell(x); m_{\mathcal{D}}(x), C_{\mathcal{D}}(x,x)\right) \mathrm{d}\ell(x). \tag{5.15}$$

Unfortunately this is typically highly non-tractable, and so we must instead sample from either an approximation, or a different utility function altogether, but one which expresses the goal we are interested in achieving.

5.4.2 Uncertainty Sampling

To approximate the behaviour of integral entropy minimisation, we can instead target the reduction in entropy of the total integrand $\ell(x)\pi(x)$ instead, by targeting

$$x_* = \arg\max_x \mathbb{V}_{\ell|\mathcal{D}}\left[\ell(x)\pi(x)\right] \tag{5.16}$$

(this is known as *uncertainty sampling*), where

$$\mathbb{V}^{\mathcal{M}}_{\ell|\mathcal{D}}\left[\ell(x)\pi(x)\right] = \pi(x) C_{\mathcal{D}}(x,x)\pi(x) = \pi(x)^2 \tilde{C}_{\mathcal{D}}(x,x)\left(\tfrac{1}{2}\tilde{C}_{\mathcal{D}}(x,x) + \tilde{m}_{\mathcal{D}}(x)^2\right), \tag{5.17}$$

in the case of our moment-matched approximation, and, under the linearisation approximation,

$$\mathbb{V}^{\mathcal{L}}_{\ell|\mathcal{D}}\left[\ell(x)\pi(x)\right] = \pi(x)^2 \tilde{C}_{\mathcal{D}}(x,x)\tilde{m}_{\mathcal{D}}(x)^2. \tag{5.18}$$

The uncertainty sampling option reduces the entropy of our GP approximation to $p(\ell)$ rather than the true (intractable) distribution. The computation of either Equation (5.17) or (5.18) is considerably cheaper and more numerically stable than that of Equation (5.15).

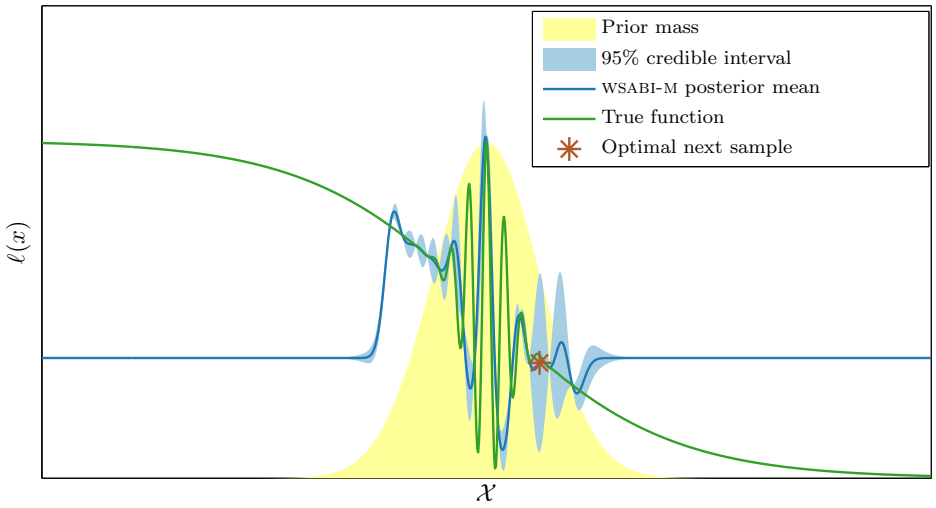
Notice that as our model builds in greater uncertainty in the likelihood where it is high, it will naturally balance sampling in entirely unexplored regions against sampling in regions where the likelihood is expected to be high. Our model (the square-root transform) is more suited to the use of uncertainty sampling than the model taken in Osborne et al. (2012a). This is because the approximation to the posterior variance is typically poorer for the extreme log-transform than for the milder square-root transform. This means that, although the log-transform would achieve greater reduction in dynamic range than any power transform, it would also introduce the most error in approximating the posterior predictive variance of $\ell(x)$. Hence, on balance, we consider the square-root transform superior for our sampling scheme.

It is worth noting that while we do not consider it further here, Bayesian Active Learning by Disagreement Houlsby N. and Husczár F. and Ghahramani Z. and Lengyel M. (20111) may further improve the simple uncertainty sampling based approach we describe above, by incorporating information from the model hyperparameter (and possibly the integral) priors into the sample selection criteria. Naturally one could also extend any of these approaches to be non-myopic (considering the optimal set of samples N steps into the future), however in practice this typically incurs several additional intractabilities, the navigation of which may lead to adverse point-set selection overall.

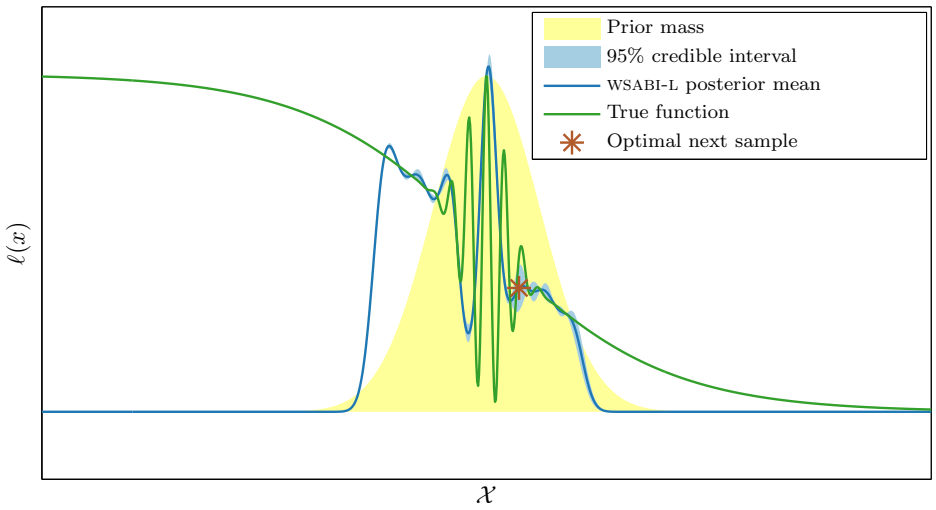
Figures 5.3a–5.3b illustrate the result of square-root Bayesian quadrature, conditioned on 15 samples selected sequentially under utility functions (5.17) and (5.18) respectively. In both cases the posterior mean has not been scaled by the prior $\pi(x)$ (but the variance has). This is intended to exaggerate the contributions to the mean made by WSABI-M.

A good posterior estimate of the integral has been achieved, and this set of samples is more informative than a grid under the utility function of minimising the integral error. In all active-learning examples a Covariance Matrix Adaptive Evolution Strategy (CMAES) (Hansen et al., 2003) global optimiser was used to explore the utility function surface before

Fig. 5.3 Examples of active sampling under both linear and moment-matched approximations to the square-root GP posterior.



(a) Square-root Bayesian quadrature with active sampling according to utility function (5.17) and corresponding moment-matched model. Note the non-zero expected mean everywhere.



(b) Square-root Bayesian quadrature with active sampling according to utility function (5.18) and corresponding linearised model. Note the zero expected mean away from samples.

selecting the next sample. This is sub-optimal, and we would suggest that performance improvements could be realised by researching more involved options.

It is worth noting that in both cases, the posterior function estimates are poor in the regions where the function is extrapolated. This is partially because we are using a stationary prior, where perhaps an assumption of non-stationarity would be more appropriate, and partially because the function has been fitted according to a utility which emphasises the importance of modelling areas of the function which maximally impact the integral estimate. As ever, in the small sample regime, (which is what we consider here), a good posterior estimate comes down to a judicious and appropriate choice of prior, and a sensible sample selection scheme.

5.5 Results

Given this new model and fast active sampling scheme for likelihood surfaces, we now test for speed against standard Monte Carlo techniques on a variety of problems.

5.5.1 Synthetic Likelihoods

We generated 16 likelihoods in four-dimensional space by selecting K normal distributions with K drawn uniformly at random over the integers 5–14. The means were drawn uniformly at random over the inner quarter of the domain (by area), and the covariances for each were produced by scaling each axis of an isotropic Gaussian by an integer drawn uniformly at random between 21 and 29. The overall likelihood surface was then given as a mixture of these distributions, with weights given by partitioning the unit interval into K segments drawn uniformly at random—'stick-breaking'. This procedure was chosen in order to generate 'lumpy' surfaces. We budgeted 500 samples for our new method per likelihood, allocating the same amount of time to Simple Monte Carlo (SMC).

Naturally the computational cost per evaluation of this likelihood is effectively zero, which afforded SMC just under 86 000 samples per likelihood on average. WSABI was on av-

erage faster to converge to 10^{-3} error (Figure 5.4a), and it is visible in Figure 5.4b that the likelihood of the ground truth is larger under this model than with SMC. This concurs with the fact that a tighter bound was achieved.

5.5.2 Marginal Likelihood of GP Regression

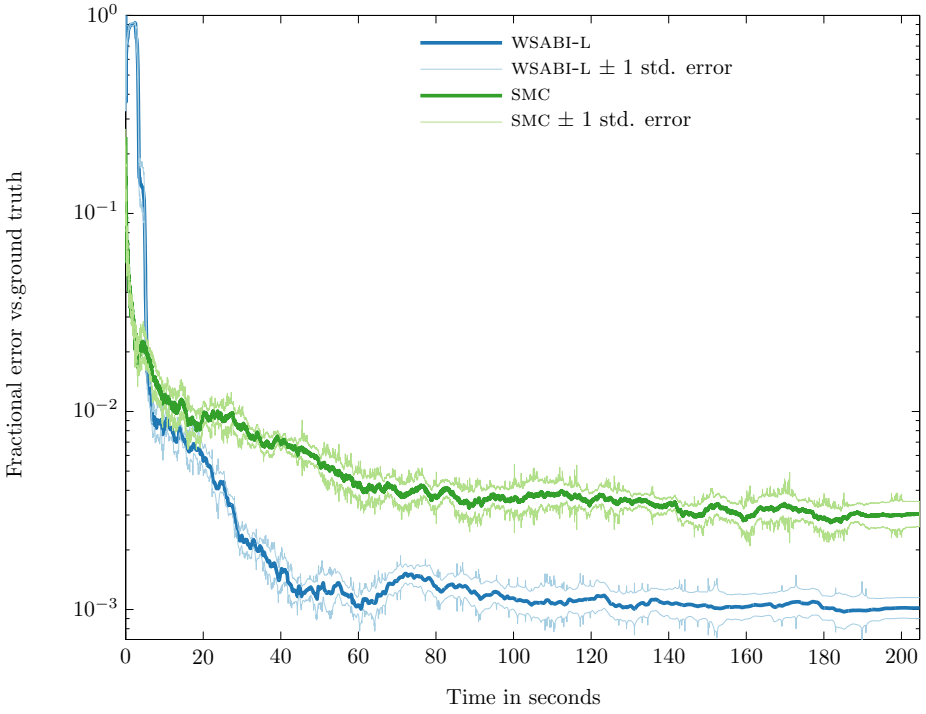
As an initial exploration into the performance of our approach on real data, we fitted a Gaussian process regression model to the *yacht hydrodynamics* benchmark dataset (Gerritsma et al., 1981). This has a six-dimensional input space corresponding to different properties of a boat hull, and a one-dimensional output corresponding to drag coefficient. The dataset has 308 examples, and using a squared exponential ARD covariance function a single evaluation of the likelihood takes approximately 0.003 seconds.

Marginalising over the hyperparameters of this model is an eight-dimensional non-analytic integral. Specifically, the hyperparameters were: an output length-scale, six input length-scales, and an output noise variance. We used a zero-mean isotropic Gaussian prior over the hyperparameters in log space with variance of 4. We obtained ground truth through exhaustive SMC sampling, and budgeted 1 250 samples for WSABI. The same amount of compute-time was then afforded to SMC, AIS (which was implemented with a Metropolis–Hastings sampler), and Bayesian Monte Carlo (BMC). SMC achieved approximately 375 000 samples in the same amount of time. We ran AIS in 10 steps, spaced on a log-scale over the number of iterations, hence the AIS plot is more granular than the others (and does not begin at 0). The 'hottest' proposal distribution for AIS was a Gaussian centered on the prior mean, with variance tuned down from a maximum of the prior variance.

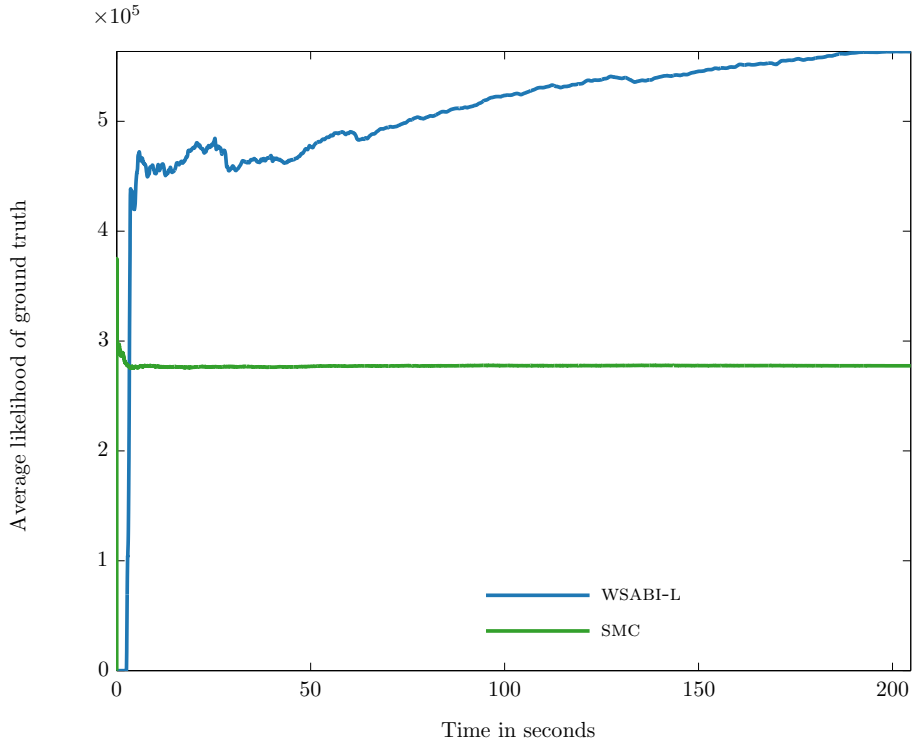
Figure 5.5 shows the speed with which WSABI converges to a value very near ground truth compared to the rest. AIS performs rather disappointingly on this problem, despite our best attempts to tune the proposal distribution to achieve higher acceptance rates.

Although the first datapoint (after 10 000 samples) is the second best performer after WSABI, further compute budget did very little to improve the final AIS estimate. BMC is by

Fig. 5.4 Experimental results of using WSABI-L to compute the model evidence for a synthetic problem.

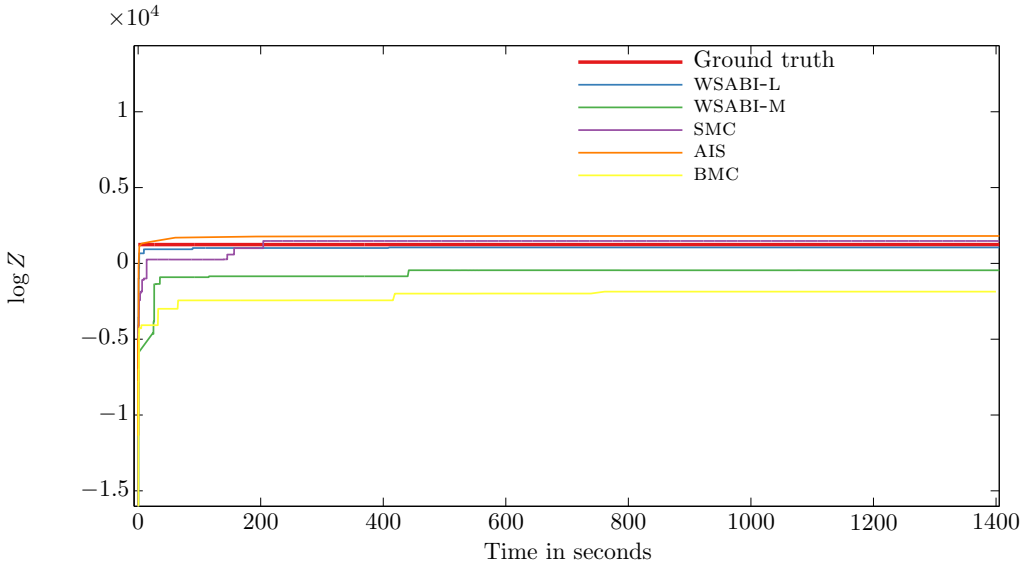


(a) Time in seconds versus average fractional error compared to the ground truth integral, as well as empirical standard error bounds, derived from the variance over the 16 runs. WSABI-M performed slightly better.



(b) Time in seconds versus average likelihood of the ground truth integral over 16 runs. WSABI-M has a significantly larger variance estimate for the integral as compared to WSABI-L.

Fig. 5.5 Log-marginal likelihood of GP regression on the yacht hydrodynamics dataset.



far the worst performer. This is because it has relatively few samples compared to SMC, and those samples were selected completely at random over the domain. It also uses a GP prior directly on the likelihood, which due to the large dynamic range will have a poor predictive performance.

5.5.3 Marginal Likelihood of GP Classification

We fitted a Gaussian process classification model to both a one dimensional synthetic dataset, as well as real-world binary classification problem defined on the nodes of a citation network (Garnett et al., 2012). The latter had a four-dimensional input space and 500 examples. We use a probit likelihood model, inferring the function values using a Laplace approximation. Once again we marginalised out the hyperparameters.

5.5.4 Synthetic Binary Classification Problem

We generate 500 binary class samples using a 1D input space. The GP classification scheme implemented in Gaussian Processes for Machine Learning Matlab Toolbox (GPML) (Rasmussen and Nickisch, 2010) is employed using the inference and likelihood framework described above. We marginalised over the three-dimensional hyperparameter space of: an

output length-scale, an input length-scale and a 'jitter' parameter. We again tested against BMC, AIS, SMC and, additionally, Doubly Bayesian Quadrature (BBQ) (Osborne et al., 2012a). Ground truth was found through 100 000 SMC samples.

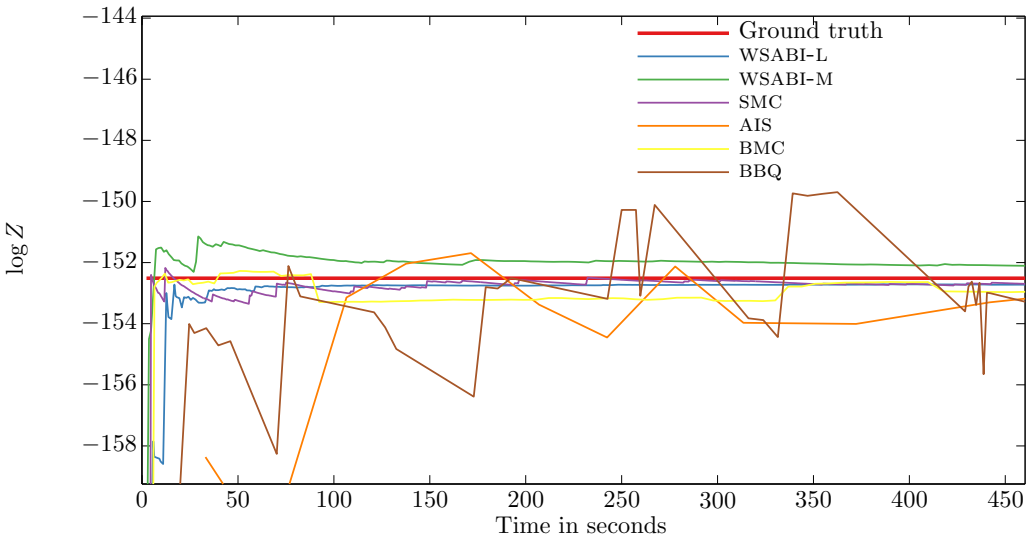
This time the acceptance rate for AIS was significantly higher, and it is visibly converging to the ground truth in Figure 5.6a, albeit in a more noisy fashion than the rest. WSABI-L performed particularly well, almost immediately converging to the ground truth, and reaching a tighter bound than SMC in the long run. BMC performed well on this particular example, suggesting that the active sampling approach did not buy many gains on this occasion. Despite this, the square-root approaches both converged to a more accurate solution with lower variance than BMC. This suggests that the square-root transform model adds value, even without an active sampling scheme. The computational cost of selecting samples under Doubly Bayesian Quadrature BBQ prevents rapid convergence.

5.5.5 Real Binary Classification Problem

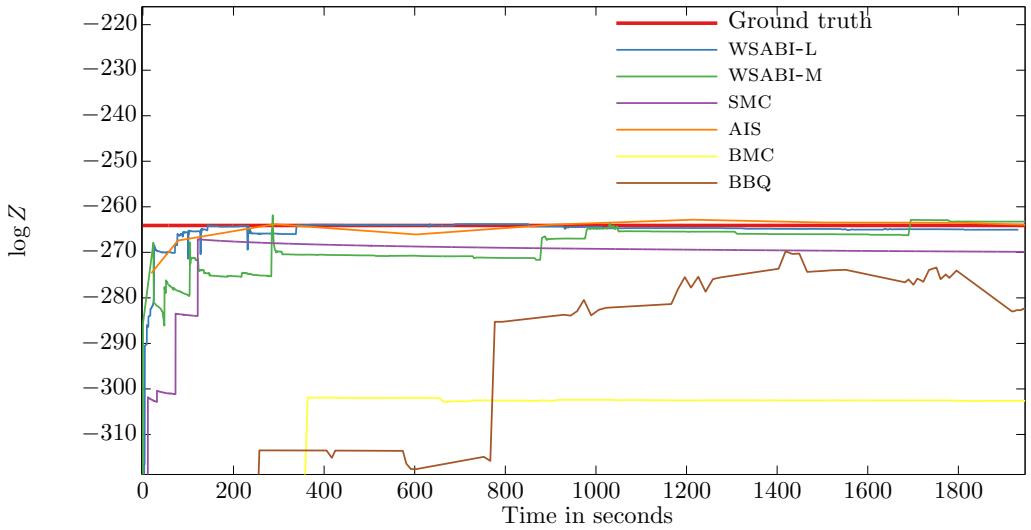
For our next experiment, we again used our method to calculate the model evidence of a GP model with a probit likelihood, this time on a real dataset.

The dataset, first described in Garnett et al. (2012), was a graph from a subset of the CiteSeer$^{\text{x}}$ citation network. Papers in the database were grouped based on their venue of publication, and papers from the 48 venues with the most associated publications were retained. The graph was defined by having these papers as its nodes and undirected citation relations as its edges. We designated all papers appearing in the Neural Information Processing Systems (NIPS) proceedings as positive observations. To generate Euclidean input vectors, the authors performed "graph principal component analysis" on this network (Fouss et al., 2007); here, we used the first four graph principal components as inputs to a GP classifier. The dataset was subsampled down to a set of 500 examples in order to generate a cheap likelihood, half of which were positive.

Fig. 5.6 Log-marginal likelihoods for GP classification on two different problems.



(a) Log-marginal likelihood for GP classification—synthetic dataset.



(b) Log-marginal likelihood for GP classification—graph dataset.

Across all our results, it is noticeable that WSABI-M typically performs worse relative to WSABI-L as the dimensionality of the problem increases. This is due to an increased propensity for exploration as compared to WSABI-L. WSABI-L is the fastest method to converge on all test cases, apart from the synthetic mixture model surfaces where WSABI-M performed slightly better. These results suggest that an active-sampling policy which aggressively exploits areas of probability mass before exploring further afield may be the most appropriate approach to Bayesian quadrature for real likelihoods.

WSABI-M optimistically maintains some non-zero prior expectation of likelihood mass over the whole domain. For most real problems, we suspect this is not appropriate, and naturally leads the estimator to converge on the ground truth from 'above'. The WSABI-L approximation meanwhile, maintains a zero prior expectation, however also suffers from a heavily deflated prior variance in areas away from existing sample locations—a property which directly leads to greedy exploitation of existing probability mass before exploring.

5.6 Conclusions

We introduced the first fast Bayesian quadrature scheme, using a novel warped likelihood model and a novel active sampling scheme. Our method, WSABI, demonstrates faster convergence (in wall-clock time) for regression and classification benchmarks than the Monte Carlo state-of-the-art.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

6.1.1 Relating to the Point Process Work:

We presented approaches for expanding the structural modelling capacity of Gaussian-process-driven Cox process models. To ensure that these models are useful for practitioners, we also derived efficient inference schemes—both exact MCMC based methods, as well as approximate but highly efficient variational techniques.

We envisage that these new models and associated inference schemes will allow for the principled inclusion of point-set observation data in a number of problem domains, where previously coarse bin-count type approximations were used as pre-processing steps to map from event data to piece-wise constant values over the domain. In particular, we would hope that researchers in epidemiology and related fields will be able to use our models to incorporate disease incidence alongside other exogenous variables, in doing so taking advantage of the benefits afforded by incorporating other variables with dependency structure.

We derived a new MCMC algorithm, applicable to a variety of Gaussian-process-driven point process models (including renewal processes), which demonstrably outperforms the

existing approaches in effective samples per second, and in fact by doing so enables inference on a wide range of point data on domains with dimensionality of 3 or higher.

We then derived the first fully variational inference scheme for Gaussian-process-driven Cox processes (to the best of our knowledge), and were in fact able to do so without making additional approximations. When extended to the marked point process domain, this resulted in a variational inference algorithm for permanental point processes. Once again all innovations were tested on both synthetic and real data, and were convincing in their performance as compared to existing approaches.

6.1.2 Relating to the Quadrature Work:

We also presented work relating to the computational scaling of existing techniques for active Bayesian quadrature. While this is unlikely to be the final form of such an algorithm, the author believes that in order to do full general Bayesian inference at scale, incorporating prior structural knowledge whilst sampling actively are both techniques of demonstrable value, and should be carried forward. It is not unreasonable to suggest that where accurate and fast inference is required, it is highly desirable to both: 1) place as much prior mass on the true solution as possible, and 2) acquire observations in an information efficient manner as possible. We tested two different versions of the approach on both real world and synthetic problems, empirically matching the state of the art even after adjusting for compute budget.

There are a couple of enabling 'tricks' employed throughout this thesis which are seemingly under-utilised within the Gaussian process community and for no definitive reason. In particular, modelling the square-root of a strictly non-negative function using a Gaussian process, while a trivial step, yields many interesting properties—both in theoretical and practical terms. Several analytic manipulations commonly required to enable Bayesian, (or approximate Bayesian), inference are made fully tractable, and this transformation also in-

curs interesting and potentially useful side effects in terms of resulting regressor properties (such as an ability to deal with high dynamic range while simultaneously being capable of modelling numerical zero).

6.2 Related Work

Naturally there is a wealth of existing literature on modelling point processes, drawn from fields including: statistics, computer science, machine learning and applied mathematics. Of particular relevance to Bayesian inference in fully-continuous Gaussian-process-driven Cox processes is the early work by Lewis (1979), in which an auxiliary variable technique is described for simulating from an inhomogenous Poisson process. Most modern MCMC based inference schemes rely on this insight (and ours is no different in that respect).

Work which was developed in parallel to our own includes a MCMC based inference scheme for Gaussian-process-driven Cox processes (Kom-Samo and Roberts, 2015), which was the first such approach to avoid using the auxiliary variable trick of Lewis (1979), and hence achieved significantly improved computational scaling. A contribution by G. de G. Matthews et al. (2016) confirmed that our variational approach does indeed lead to a consistent and theoretically sound marginal objective function. The same author also extends on existing contributions made by McCullagh and Yang (2006) in G. de G. Matthews and Ghahramani (2014), where the marked Cox process is reimagined as a classification algorithm and a novel inference scheme is devised.

There are a few application papers which target the datasets we use to benchmark our approach, of note is Miller et al. (2014). In this work, the authors use a multi-stage approach to factorise point shot location maps in the NBA. They do not specify a full generative model, but instead rely on a two-stage data processing pipeline, which is defined to be a binned log-Gaussian Cox process, followed by non-negative matrix factorisation (NMF) in order to arrive at the maximally descriptive basis set. Most Cox process application work does in fact

rely on the binned log-Gaussian Cox process model, and therefore requires discretisation of the domain as a pre-processing step. Discretisation is an inherently difficult procedure, as naturally different binning schemes may yield radically different piecewise constant rate functions over the domain. Beyond a uniform grid, we must also ask questions as to whether the bin size should be fixed across dimensions, or whether it should vary—perhaps spatially as well.

In general we believe there is evidence to support the fact that inhomogenous point processes are a relatively under-explored area of research within the machine learning community. This is particularly so given the fundamental nature of the topic, general availability of data, and when we measure output relative to work on other non-parametric stochastic processes.

With reference to Bayesian quadrature, most recent application driven work is by Osborne: Osborne et al. (2012c) specifies an algorithm designed to perform inference directly on the value of the ratio of two integrals, a problem often encountered during model selection. Garnett et al. (2012) deploys an active learning scheme and a log Gaussian process to infer model evidence integrals whilst coping well with the high dynamic range of typical likelihood functions. Earlier foundational work by O'Hagan et al. as well as these new contributions are well described in Osborne (2010).

Oates, Briol and Girolami are pursuing a more formal understanding of the theoretical properties of existing approaches for probabilistic integration (including but not limited to Bayesian quadrature). Briol et al. (2015a) demonstrates that under a restriction to kernels in finite dimensional reproducing kernel Hilbert spaces, the convergence rate of the resulting Bayesian quadrature rule will be exponential, and the posterior concentration rate will be super exponential in the number of sample points. A more general overview paper, which also relates probabilistic integration to the emergent field of probabilistic numerics may be found in Briol et al. (2015b).

6.3 Future Work

For the point process work, there are two immediate directions of further research:

On the application side, it would be useful to merge our variational approach for Cox processes into a generalised Gaussian process framework for dealing with any number of heterogenous input data streams. From discussions with epidemiology researchers, it would seem that such a model may bring additional value to the study of Dengue fever movement.

More generally, there are cases where we might view Cox processes in a more abstract fashion, and use them to construct structured models over data which may not traditionally be interpreted as a point process. For example, in the absence of any other information, a typical network may be interpreted only through the node-node communication events that occur at semi-random points in time (and potentially space or other covariates). There are manifold ways to specify such a model, but it is reasonable to expect that an appropriate choice might achieve state of the art performance on extracting structure from message data in an unsupervised fashion, much as we were able to in LPPA.

For Bayesian quadrature, one can reiterate most of the points made in the concluding chapter of Osborne (2010), specifically those relating to the design of priors which incorporate more integrand structure than simply smoothness and non-negativity. It would be interesting to see uptake of a system automating (or at least attempting to) the role of the quadrature rule 'designer'—a niche profession, and one where a human constructs an optimal quadrature rule for a specific problem in industry.

Joint inference of both kernel structure and non-stationarity is an alternative thread to follow—here we would hope to arrive at some middle ground between a raw delta function based approximation to an integral, and one where a heavily structured prior is employed as a starting density.

On the theory side, we expect to see extended convergence work for Bayesian quadrature in the general setting, and hope that concerns highlighted in Szabó et al. (2015) do not impede the development of concentration results for Gaussian process based rules constructed using kernels defined on infinite dimensional reproducing kernel Hilbert spaces.

6.4 Final Thoughts

While the lofty goals set out in the introductory chapters of this thesis may not quite have been achieved, we retain our conviction that scalable (semi black-box) Bayesian inference is still important for modern data science and machine learning. We hope that our contributions towards this larger goal have been explained and demonstrated fully, and that they have also been set in their proper context.

As a more general note to position our work with respect to the current state of the machine learning field: The majority of statistical machine learning requires some form of function approximation. While we favour Bayesian kernel approaches, many have turned to deep neural networks for use as a base regressor. It is clear that great leaps in performance have been achieved on single task high signal-to-noise large data problems using deep neural networks, however it is not clear whether: 1. The fact that the mapping imposed by a deep neural network architecture is coarsely restricted to be a function of strictly polynomial type is inappropriate for some problems and use cases—empirically it would seem that in most instances this is not the case, and some make arguments based on the prevalence of polynomial energy functions in Physics that the natural world might be 'polynomial friendly' (Lin and Tegmark, 2016). 2. These innovations really do bring us any closer to solving either general 'intelligence' or statistical prediction—as opposed to simply helping resolve the machine perception problem. Attempts to extend these techniques into the world of generative modelling (which we do not suggest equate to intelligence), for example, while interesting

have been (relatively speaking) unsuccessful to date (Goodfellow et al., 2014; Kingma and Welling, 2014).

It is relatively certain that that there will always be low signal-to-noise prediction problems to solve, of which Cox processes form a class of examples. In these cases it will continue to be important to: 1. Derive well calibrated uncertainty estimates, and 2. incorporate as much prior structure as possible while restricting the space of possible solutions as little as possible—both of which may be achieved via the judicious application of efficient Bayesian inference.

Appendix A

Details on Variational Inference for Cox Processes

A.1 Automatic Relevance Determination Kernel

In this work we use the exponentiated quadratic (also known as the "squared exponential") ARD kernel:

$$K(\mathbf{x}, \mathbf{x}') = \prod_{r=1}^{R} \exp\left(-\frac{(x_r - x_r')^2}{2\alpha_r}\right). \tag{A.1}$$

A.2 Derivation of the Lower Bound

The lower bound Equation 4.7 is derived as follows:

$$\log p(\mathscr{D}_{1:S}, A_{1:S}|\Theta) = \log \left[\iint p(\mathscr{D}_{1:S}, A_{1:S}|f_{1:T}) \mathrm{d}p(f_{1:T}|\mathbf{u}_{1:T}) p(\mathbf{u}_{1:T}) \frac{q(\mathbf{u}_{1:T})}{q(\mathbf{u}_{1:T})} \, \mathrm{d}\mathbf{u}_{1:T} \right] \tag{A.2}$$

$$\geq \int \prod\nolimits_t \int \mathrm{d}p(f_t \mid \mathbf{u}_t)\, q(\mathbf{u}_t)\, \mathrm{d}\mathbf{u}_t \, \log\left[p(\mathscr{D}_{1:S}, A_{1:S} \mid f_{1:T})\right] + \iint \mathrm{d}p(f_{1:T} \mid \mathbf{u}_{1:T})\, q(\mathbf{u}_{1:T}) \, \log\left[\frac{p(\mathbf{u}_{1:T})}{q(\mathbf{u}_{1:T})}\right] \, \mathrm{d}\mathbf{u}_{1:T} \tag{A.3}$$

$$= \mathbb{E}_{q(f_{1:T})}\left[\log p(\mathscr{D}_{1:S}, A_{1:S} \mid f_{1:T})\right] - \mathrm{KL}\left(q(\mathbf{u}_{1:T}) \,\middle\|\, p(\mathbf{u}_{1:T})\right) \tag{A.4}$$

$$\triangleq \mathscr{L}(\mathscr{D}_{1:S}, A_{1:S}; \Theta). \tag{A.5}$$

A.3 Definition of the KL-divergence between two Multivariate Gaussians

The KL term in Equations 4.7 and 3.10 is the Kullback–Leibler divergence between T pairs of independent Gaussian distributions, and is defined by:

$$\mathrm{KL}\left(q(\mathbf{u}_{1:T}) \,\middle\|\, p(\mathbf{u}_{1:T})\right) = \frac{1}{2} \sum\nolimits_t \left[\mathrm{tr}\left(\mathbf{K}_{zz}^{-1}\mathbf{S}_t\right) + (\vec{1}\bar{u}_t - \mathbf{m})^\top \mathbf{K}_{zz}^{-1}(\vec{1}\bar{u}_t - \mathbf{m}) - M + \log \frac{|\mathbf{K}_{zz}|}{|\mathbf{S}_t|} \right]. \tag{A.6}$$

A.4 Definition of $\tilde{G}$

The function $\tilde{G}$ that appears in the expectation

$$\mathbb{E}_{q(f_t)}[\log f_{s,t,n}^2] = \int_{-\infty}^{\infty} \log(f_{s,t,n}^2)\, \mathcal{N}(f_{s,t,n}; \tilde{\mu}_{s,t,n}, \tilde{\sigma}_{s,t,n}^2)\, \mathrm{d}f_{s,t,n}, \tag{A.7}$$

and Equation 4.10, is a specialised version of the partial derivative of the confluent hypergeometric function,

$$ {}_1F_1\left(a, b, z\right) = \sum_{k=0}^{\infty} \frac{(a)_k z^k}{(b)_k\, k!}, \tag{A.8} $$

with respect to its first argument and is defined by:

$$ \tilde{G}(z) = {}_1F_1{}^{(1,0,0)}\left(0, \frac{1}{2}, z\right) = 2z \sum\nolimits_{j=0}^{\infty} \frac{j!\ z^j}{(2)_j (1\tfrac{1}{2})_j}, \tag{A.9} $$

where $(\cdot)_j$ denotes the rising Pochhammer series $(a)_0 = 1,\ (a)_j = a(a+1)(a+2)\ldots(a+j-1)$.

A.5 Definition of the Marginalised Inducing Covariance

For the ARD Kernel the function $\Psi(\mathbf{z}, \mathbf{z}') = \int_{\mathcal{X}} K(\mathbf{z}, \mathbf{x}) K(\mathbf{x}, \mathbf{z}') \mathrm{d}\mathbf{x}$ can be computed in closed form:

$$ \Psi(\mathbf{z}, \mathbf{z}') = \prod\nolimits_{r=1}^{R} \frac{\sqrt{\pi \alpha_r}}{2} \exp\left(-\frac{(z_r - z_r')^2}{4\alpha_r}\right) \left[\operatorname{erf}\left(\frac{\bar{z}_r - \mathcal{X}_r^{\mathrm{Min}})}{\sqrt{\alpha_r}}\right) - \operatorname{erf}\left(\frac{\bar{z}_r - \mathcal{X}_r^{\mathrm{Max}}}{\sqrt{\alpha_r}}\right)\right], \tag{A.10} $$

where $\bar{z}_r = \frac{1}{2}(z_r + z_r')$.

A.6 Detailed Derivation of the Collapsed Bound

The set of all possible assignments is:

$$\left\{\{A_1^{(1)}=1,\dots,A_S^{(N_S)}=1\},\dots,\{A_1^{(1)}=T,\dots,A_S^{(N_S)}=T\}\right\},$$

In the collapsed bound we sum over all the possible assignments to each of the allocation variables:

$$\begin{aligned}
\log p(\mathscr{D}_{1:S}|\Theta) &= \log \sum\nolimits_{A_{1:S}} p(\mathscr{D}_{1:S}, A_{1:S}|\Theta) && \text{(A.11)}\\
&\geq \log \sum\nolimits_{A_{1:S}} \exp\big(\mathscr{L}(\mathscr{D}_{1:S}, A_{1:S};\Theta)\big) && \text{(A.12)}\\
&= \log \sum\nolimits_{A} \exp\left(\mathfrak{B}+\sum\nolimits_s\sum\nolimits_n\sum\nolimits_t \mathbb{1}\{A_s^{(n)}=t\}\mathfrak{A}_{s,t,n}\right) && \text{(A.13)}\\
&= \log\left[\exp(\mathfrak{B})\times\sum\nolimits_{A_1^{(1)}=1}^{T}\cdots\sum\nolimits_{A_S^{(N_S)}=1}^{T}\prod\nolimits_s\prod\nolimits_n\exp\left(\sum\nolimits_t\mathbb{1}\{A_s^{(n)}=t\}\mathfrak{A}_{s,t,n}\right)\right] && \text{(A.14)}\\
&= \log\left[\exp(\mathfrak{B})\times\prod\nolimits_s\prod\nolimits_n\sum\nolimits_{A_s^{(n)}=1}^{T}\exp\left(\sum\nolimits_t\mathbb{1}\{A_s^{(n)}=t\}\mathfrak{A}_{s,t,n}\right)\right] && \text{(A.15)}\\
&= \log\left[\exp(\mathfrak{B})\times\prod\nolimits_s\prod\nolimits_n\sum\nolimits_t\exp\left(\mathfrak{A}_{s,t,n}\right)\right] && \text{(A.16)}\\
&= \mathfrak{B}+\sum\nolimits_s\sum\nolimits_n\log\sum\nolimits_t\exp\mathfrak{A}_{s,t,n} && \text{(A.17)}\\
&\triangleq \mathscr{L}(\mathscr{D}_{1:S};\Theta) && \text{(A.18)}
\end{aligned}$$

A.7 Benchmark

The benchmark kernel smoother optimises the leave-one-out training objective:

$$\Sigma_s^* = \underset{\Sigma}{\operatorname{argmax}} \sum\nolimits_{i=1}^{N_s}\log\sum\nolimits_{j\neq i=1}^{N_s}\mathscr{N}_{\square}(\mathbf{x}^{(s,i)};\mathbf{x}^{(s,j)},\Sigma). \tag{A.19}$$

We can construct the test log-likelihood for the held-out datasets as:

$$\log p(\mathscr{H}_{1:S}|\mathscr{D}_{1:S}, \Sigma^*_{1:S}) = \sum_{s=1}^{S}\sum_{n=1}^{\tilde{N}_h} \log \sum_{t=1}^{T} a_{s,t}\, b_{t,m(h,n)} - |\Delta \mathbf{x}| \sum_{s=1}^{S}\sum_{t=1}^{T}\sum_{b=1}^{B} a_{s,t} b_{t,b}$$

where $m(h,n)$ is a function that maps a test data point $\tilde{\mathbf{x}}^{(h,n)}$ into the d^{th} grid-cell. For the CT case the weight matrix $\mathbf{A}$ is optimised for the test data.

A.8 Mixed Continuous Discrete Co-ordinate Spaces

This Ψ-function in the mixed co-ordinate space case is $\Psi(z_r, z'_r) = \sum_{x_r} K(z_r, x_r)K(x_r, z_r)$. When using Kronecker structure $\Psi_{z_2 z_2}$ is simply $\mathbf{K}_{z_2 z_2}\mathbf{K}_{z_2 z_2}$ if $\mathscr{Z}$ contains all feeding station locations and the discrete dimension is $r = 2$.

References

Malaria Atlas Project. http://www.map.ox.ac.uk/explore/data-modelling/, 2014. [Online; accessed 24-October-2014].

R. P. Adams, I. Murray, and D. J. C. MacKay. Tractable Nonparametric Bayesian Inference in Poisson Processes with Gaussian Process Intensities. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML, pages 9–16, New York, NY, USA, 2009. ACM.

M. A. Álvarez and N. D. Lawrence. Computationally Efficient Convolved Multiple Output Gaussian Processes. *Journal of Machine Learning Research*, 12:1459–1500, July 2011. ISSN 1532-4435.

M. A. Álvarez, D. Luengo, and N. D. Lawrence. Latent Force Models. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, AISTATS, pages 9–16, 2009.

S. Amari and H. Nagaoka. *Methods of Information Geometry*. Oxford University Press, 2007.

S. Ambikasaran and E. Darve. The inverse fast multipole method. *arXiv pre-print 1407.1572*, 2014.

L. U. Ancarani and G. Gasaneo. Derivatives of any order of the confluent hypergeometric function ${}_1F_1(a, b, z)$ with respect to the parameter a or b. *Journal of Mathematical Physics*, 49(6), 2008.

O. E. Barndorff-Nielsen. *Information and exponential families: in statistical theory*. Wiley series in probability and mathematical statistics: Tracts on probability and statistics. Wiley, 1978. ISBN 9780471995456.

J. O. Berger. *Statistical decision theory and Bayesian analysis : with 23 illustrations*. Springer series in statistics. Springer, New York, Berlin, Heidelberg, 1985. ISBN 3-540-96098-8. URL http://opac.inria.fr/record=b1092781. Autre tirage : 2010.

S. Bhatt, D. J. Weiss, E. Cameron, D. Bisanzio, B. Mappin, U. Dalrymple, K. E. Battle, C. L. Moyes, A. Henry, P. A. Eckhoff, E. A. Wenger, O. Briët, M. A. Penny, T. A. Smith, A. Bennett, J. Yukich, T. P. Eisele, J. T. Griffin, C. A. Fergus, M. Lynch, F. Lindgren, J. M. Cohen, C. L. J. Murray, D. L. Smith, S. I. Hay, R. E. Cibulskis, and P. W. Gething. The effect of malaria control on Plasmodium falciparum in Africa between 2000 and 2015. *Nature*, 526:207–211, October 2015. doi: 10.1038/nature15535.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 1st edition 2006 edition, October 2007. ISBN 0387310738.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003. ISSN 1532-4435.

W. J. Borucki, D. G. Koch, G. Basri, N. Batalha, A. Boss, T. M. Brown, D. Caldwell, J. Christensen-Dalsgaard, W.D. Cochran, E. DeVore, et al. Characteristics of Kepler planetary candidates based on the first data set. *The Astrophysical Journal*, 728:117, 2011.

Z. I. Botev, J. F. Grotowski, and D. P. Kroese. Kernel Density Estimation via Diffusion. *The Annals of Statistics*, 38(5):2916–2957, 2010.

FX. Briol, C. J. Oates, M. Girolami, and M. A. Osborne. Frank-Wolfe Bayesian Quadrature: Probabilistic Integration with Theoretical Guarantees. *Advances in Neural Information Processing Systems*, 2015a. URL http://arxiv.org/abs/1506.02681.

FX. Briol, C. J. Oates, M. Girolami, M. A. Osborne, and D. Sejdinovic. Probabilistic Integration: A Role for Statisticians in Numerical Analysis? *arXiv:1512.00933 [cs, math, stat]*, December 2015b. URL http://arxiv.org/abs/1512.00933.

S. P. Brooks and G. O. Roberts. Convergence assessment techniques for Markov chain Monte Carlo. *Statistics and Computing*, 8(4):319–335, 1998.

B. P. Carlin, A. E. Gelfand, and A. F. M. Smith. Hierarchical Bayesian analysis of change-point problems. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 41 (2):389–405, Jan 1992.

A. T. Cemgil. Bayesian Inference for Non-negative Matrix Factorisation Models. *Computational Intelligence and Neuroscience*, 2009.

J. Chilès and P. Delfiner. *Kriging*, pages 147–237. John Wiley & Sons, Inc., 2012. ISBN 9781118136188. doi: 10.1002/9781118136188.ch3. URL http://dx.doi.org/10.1002/9781118136188.ch3.

M. K. Cowles, G. O. Roberts, and J. S. Rosenthal. Possible biases induced by Markov Chain Monte Carlo convergence diagnostics. *Journal of Statistical Computation and Simulation*, 64(1):87, 1999.

R. T. Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1):1–13, 1946. URL http://link.aip.org/link/?AJP/14/1/1.

J. P. Cunningham, K. V. Shenoy, and M. Sahani. Fast Gaussian process methods for point process intensity estimation. In *Proceedings of the 25th international conference on Machine learning*, ICML, pages 192–199. ACM, 2008.

P. Diaconis. Bayesian numerical analysis. In S. Gupta J. Berger, editor, *Statistical Decision Theory and Related Topics IV*, volume 1, pages 163–175. Springer-Verlag, New York, 1988.

P. Diggle. A kernel method for smoothing point process data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 34(2):pp. 138–147, 1985. ISSN 00359254.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216 – 222, 1987. ISSN 0370-2693.

D. Duvenaud. *Automatic Model Construction with Gaussian Processes*. PhD thesis, Computational and Biological Learning Laboratory, University of Cambridge, 2014.

N. Eisenbaum and H. Kaspi. On permanental processes. *Stochastic Processes and their Applications*, 119(5):1401 – 1415, 2009. ISSN 0304-4149.

P. Fearnhead. Exact and efficient Bayesian inference for multiple changepoint problems. *Statistics and Computing*, 16(2):203–213, Jun 2006. doi: 10.1007/s11222-006-8450-8.

S. Flaxman, A. Wilson, D. Neill, H. Nickisch, and A. Smola. Fast Kronecker Inference in Gaussian Processes with non-Gaussian Likelihoods. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML, 2015.

F. Fouss, A. Pirotte, J-M Renders, and M. Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE *Transactions on Knowledge and Data Engineering*, 19(3):355–369, 2007.

A. G de G Matthews. *Scalable Gaussian Process Inference using Variational Methods*. PhD thesis, Computational and Biological Learning Laboratory, University of Cambridge, 2016.

A. G. de G. Matthews and Z. Ghahramani. Classification using log Gaussian Cox processes. http://arxiv.org/abs/1405.4141, 2014.

A. G. de G. Matthews, J. Hensman, R. E Turner, and Z. Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Statistics*, AISTATS, 2016.

R. Garnett, Y. Krishnamurthy, X. Xiong, J. Schneider, and R. P. Mann. Bayesian optimal active search and surveying. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning*, ICML. Omnipress, Madison, WI, USA, 2012.

J. Gerritsma, R. Onnink, and A. Versluis. Geometry, resistance and stability of the Delft systematic yacht hull series. *International shipbuilding progress*, 28(328), 1981.

P. J. Gmytrasiewicz and E. H. Durfee. Rational Coordination in Multi-Agent Environments. *Autonomous Agents and Multi-Agent Systems*, 3(4):319–350, 2000.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. 2014.

G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, 1985.

G. Grimmett and D. Stirzaker. *One Thousand Exercises in Probability*. Oxford University Press, 2001.

T. Gunter*, C. Lloyd*, M. A. Osborne, and S. J. Roberts. Efficient Bayesian Nonparametric Modelling of Structured Point Processes. In *Uncertainty in Artificial Intelligence (UAI)*, 2014.

T. Gunter, M. A. Osborne, R. Garnett, P. Hennig, and S.J. Roberts. Sampling for inference in probabilistic models with fast Bayesian quadrature. In C. Cortes and N. Lawrence, editors, *Advances in Neural Information Processing Systems (NIPS)*, 2014.

N. Hansen, S. D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation. *Evolutionary Computation*, 11(1):1–18, 2003.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

A. G. Hawkes. Spectra of Some Self-Exciting and Mutually Exciting Point Processes. *Biometrika*, 58(1):83–90, 1971. ISSN 00063444. doi: 10.2307/2334319. URL http://dx.doi.org/10.2307/2334319.

J. Hensman, M. Rattray, and N. D. Lawrence. Fast Variational Inference in the Conjugate Exponential Family. In *Advances in Neural Information Processing Systems*, NIPS, 2012.

J. Hensman, N. Fusi, and N. D Lawrence. Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intellegence*, pages 282–290. auai.org, 2013.

J. B. Hough, M. Krishnapur, Y. Peres, and B. Virág. Determinantal processes and independence. *Probability Surveys*, 2006. doi: 10.1214/154957806000000078.

Houlsby N. and Husczár F. and Ghahramani Z. and Lengyel M. Bayesian Active Learning for classification and preference learning., 20111. URL http://arxiv.org/abs/1112.5745.

T. Iwata, A. Shah, and Z. Ghahramani. Discovering Latent Influence in Online Social Activities via Shared Cascade Poisson Processes. In *Proceedings of the 19th Conference on Knowledge Discovery and Data Mining*, KDD, 2013.

T. S. Jaakkola. Tutorial on variational approximation methods. In *Advanced Mean Field Methods: Theory and Practice*, pages 129–159. MIT Press, 2000.

R. G. Jarrett. A note on the intervals between coal-mining disasters. *Biometrika*, 66(1): 191–193, 1979. doi: 10.1093/biomet/66.1.191.

M. I Jordan, Z. Ghahramani, T. S Jaakkola, and L. K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

M. Potters J.P. Bouchaud. Financial Applications of Random Matrix Theory: a short review. https://arxiv.org/abs/0910.1205, 2009.

J. N. Kapur. *Maximum-entropy Models in Science and Engineering*. Wiley, 1989. ISBN 9788122402162. URL https://books.google.co.uk/books?id=LuNIAp3QorUC.

M. Kennedy. Bayesian quadrature with non-normal approximating functions. *Statistics and Computing*, 8(4):365–375, 1998.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations*, ICLR, 2014.

J. F. C. Kingman. *Poisson Processes (Oxford Studies in Probability)*. Oxford University Press, January 1993. ISBN 0198536933.

YL. Kom-Samo and S. J. Roberts. Scalable Nonparametric Bayesian Inference on Point Processes with Gaussian Processes. In *Proceedings of the International Conference on Machine Learning*, ICML, 2015.

T. A. Lasko. Efficient Inference of Gaussian Process Modulated Renewal Processes with Application to Medical Event Data. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, UAI, 2014.

G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. SIAM, 1999.

D. D. Lee and S. H. Seung. Algorithms for Non-negative Matrix Factorization. In *Advances in Neural Information Processing Systems*, NIPS, 2001.

P. A. W. & Shedler G.S. Lewis. Simulation of a Nonhomogeneous Poisson Process by Thinning. *Naval Research Logistics Quarterly*, 26:403–413, 1979.

W. Lian, V. Rao, B. Eriksson, and L. Carin. Modeling Correlated Arrival Events with Latent Semi-Markov Processes. In *Proceedings of the 31st International Conference on Machine Learning*, ICML, 2014.

H. W. Lin and M. Tegmark. Why does deep and cheap learning work so well?, 2016. URL http://arxiv.org/abs/1608.08225.

S. Linderman and R. P. Adams. Discovering Latent Network Structure in Point Process Data. In *Proceedings of the 31st International Conference on Machine Learning*, ICML, 2014.

C. Lloyd*, T. Gunter*, M. A. Osborne, and S. J. Roberts. Variational Inference for Gaussian Process Modulated Point Processes. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML, 2015.

C. Lloyd, T. Gunter, T. Nickson, M.A. Osborne, and S.J. Roberts. Latent Point Process Allocation (LPPA). In *The 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

J. R. Lloyd, D. Duvenaud, R. Grosse, J. B. Tenenbaum, and Z. Ghahramani. Automatic construction and natural-language description of nonparametric regression models. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2014.

D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002. ISBN 0521642981.

P. McCullagh and J. Yang. Stochastic classification models. In *Proceedings of the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, pages 669–686, 2006.

X. Meng and W. H. Wong. Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*, 6(4):831–860, 1996.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

A. Miller, L. Bornn, R. P. Adams, and K. Goldsberry. Factorized Point Process Intensities: A Spatial Analysis of Professional Basketball. In *Proceedings of the 31st Annual International Conference on Machine Learning*, ICML, 2014.

T. P. Minka. Deriving quadrature rules from Gaussian processes. Technical report, Statistics Department, Carnegie Mellon University, 2000.

T. 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.

I. Murray, R. P. Adams, and D. J. C. MacKay. Elliptical Slice Sampling. *Journal of Machine Learning Research - Proceedings Track*, 9:541–548, 2010.

J. Møller, A. R. Syversveen, and R. P. Waagepetersen. Log Gaussian Cox Processes. *Scandinavian Journal of Statistics*, 25(3):451–482, 1998. ISSN 1467-9469. doi: 10.1111/1467-9469.00115. URL http://dx.doi.org/10.1111/1467-9469.00115.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993. URL http://www.cs.toronto.edu/~R./ftp/review.pdf.

R. M. Neal. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC Press, 2010.

R.M. Neal. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

A. O'Hagan. Monte Carlo is fundamentally unsound. *The Statistician*, 36:247–249, 1987.

A. O'Hagan. Bayes-Hermite quadrature. *Journal of Statistical Planning and Inference*, 29:245–260, 1991.

Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications (Universitext)*. Springer, 6th edition, January 2014. ISBN 3540047581. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/3540047581.

M. A. Osborne. *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*. PhD thesis, University of Oxford, 2010. Available at www.robots.ox.ac.uk/~mosb/full_thesis.pdf.

M. A. Osborne, R. Garnett, and S. J. Roberts. Gaussian Processes for Global Optimisation. In *3rd International Conference on Learning and Intelligent Optimization (LION3)*, 2009. available at http://lion.disi.unitn.it/intelligent-optimization//LION3/online_proceedings/94.pdf.

M. A. Osborne, D. K. Duvenaud, R. Garnett, C. E. Rasmussen, S. J. Roberts, and Z. Ghahramani. Active learning of model evidence using Bayesian quadrature. In P. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, NIPS. MIT Press, Cambridge, MA, 2012a.

M. A. Osborne, S. J. Roberts, A. Rogers, and N. R. Jennings. Real-time information processing of environmental sensor network data. *ACM Transactions on Sensor Networks*, 9 (1), 2012b. doi: 10.1145/2379799.2379800.

M.A. Osborne, R. Garnett, S.J. Roberts, C. Hart, S. Aigrain, N.P. Gibson, and S. Aigrain. Bayesian quadrature for ratios. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*, 2012c.

C. J. Paciorek and M. J. Schervish. Nonstationary covariance functions for Gaussian process regression. In *Advances in Neural Information Processing Systems*, NIPS. MIT Press, 2004.

N. S. Pillai, Q. Wu, F. Liang, S. Mukherjee, and R. L. Wolpert. Characterizing the Function Space for Bayesian Kernel Models. *Journal of Machine Learning Research*, 8(8), 2007.

I. Psorakis, S. J. Roberts, I. Rezek, and B. C. Sheldon. Inferring social network structure in ecological systems from spatio-temporal data streams. *Journal of The Royal Society Interface*, 9(76):3055–3066, 2012. ISSN 1742-5689. doi: 10.1098/rsif.2012.0223.

J. Quiñonero Candela and C. E. Rasmussen. A Unifying View of Sparse Approximate Gaussian Process Regression. *Journal of Machine Learning Research*, 6:1939–1959, December 2005. ISSN 1532-4435.

V. Rao and Y. W. Teh. Gaussian Process Modulated Renewal Processes. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, Granada, Spain*, pages 2474–2482, 2011.

C. E. Rasmussen and Z. Ghahramani. Bayesian Monte Carlo. In *Advances in Neural Information Processing Systems 15*, pages 489–496, Cambridge, MA, USA, October 2003. Max-Planck-Gesellschaft, MIT Press.

C. E. Rasmussen and H. Nickisch. Gaussian processes for machine learning toolbox. *The Journal of Machine Learning Research*, 11, 2010.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2nd edition 2006 edition, 2006. ISBN 0-262-18253-X.

S. L. Rathbun and N. Cressie. Asymptotic properties of estimators for the parameters of spatial inhomogeneous Poisson point processes. *Advances in Applied Probability*, page 122–154, 1994.

W. Rudin. *Real and Complex Analysis, 3rd Ed.* McGraw-Hill, Inc., New York, NY, USA, 1987. ISBN 0070542341.

S. Sarkka. *Bayesian Filtering and Smoothing*. Cambridge University Press, New York, NY, USA, 2013. ISBN 1107619289, 9781107619289.

Y. Shoham, R. Powers, and T. Grenager. If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171(7):365–377, 2007.

A. Simma and M. I. Jordan. Modeling Events with Cascades of Poisson Processes. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, UAI, 2010.

J. Skilling. Nested sampling. *Bayesian inference and maximum entropy methods in science and engineering*, 735:395–405, 2004.

E. Snelson and Z. Ghahramani. Sparse Gaussian Processes using Pseudo-inputs. In *Advances in Neural Information Processing Systems*, NIPS, 2005.

P. Stone and M. Veloso. Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robots*, 8(3):345–383, 2000.

D Stoyan, W.S. Kendall, and J. Mecke. *Stochastic geometry and its applications*. Wiley series in probability and mathematical statisitics. Wiley, Chichester, W. Sussex, New York, 1987. ISBN 0-471-90519-4. URL http://opac.inria.fr/record=b1117459. Rev. translation of: Stochastische Geometrie.

B. Szabó, A. W. van der Vaart, and J. H. van Zanten. Frequentist coverage of adaptive nonparametric Bayesian credible sets. *Annals of Statistics*, 43(4):1391–1428, 08 2015. doi: 10.1214/14-AOS1270. URL http://dx.doi.org/10.1214/14-AOS1270.

Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric Latent Factor Models. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, volume 10 of *AISTATS*, 2005.

M. K. Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *Artificial Intelligence and Statistics*, AISTATS, 2009.

M. K. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. In *Artificial Intelligence and Statistics*, AISTATS, 2010.

G. Wei, P. Clifford, and J. Feng. Population death sequences and Cox processes driven by interacting Feller diffusions. *Journal of Physics A: Mathematical and General*, 35(44): 9309, 2002. URL http://stacks.iop.org/0305-4470/35/i=44/a=303.

A. Wilson, E. Gilboa, A. Nehorai, and J. Cunningham. Fast kernel learning for multidimensional pattern extrapolation. In *Advances in Neural Information Processing Systems*, NIPS. 2014.